ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS ADMINISTRATIVAS

UNIDAD DE TITULACIÓN

COMPARACIÓN ENTRE VARIOS MÉTODOS DE PRONÓSTICOS BASADOS EN SERIES DE TIEMPO PARA PREDECIR LA DEMANDA DE PLACAS DIGITALES EN EMPRESAS DEL SECTOR GRÁFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN GERENCIA EMPRESARIAL

JHIMY XAVIER PONCE JARRIN
jponce@globalgraphic.com.ec

Director: Ing. Alex Dávila Frías
alex.davila@epn.edu.ec

DECLARACIÓN

Yo, Jhimy Xavier Ponce Jarrín declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Jhimy Xavier Ponce Jarrín

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Jhimy Xavier Ponce Jarrín bajo mi supervisión.

Ing. Alex Vicente Dávila Frías

DIRECTOR

AGRADECIMIENTOS

Agradezco a mi familia por su compromiso, constante preocupación y comprensión durante el desarrollo de esta investigación.

Además un merecido agradecimiento a la Escuela Politécnica Nacional, por toda la formación impartida durante pre-grado y maestría, que han permitido la culminación exitosa del presente trabajo.

A todas las empresas del sector gráfico quiteño que facilitaron la información de sus consumos de placas digitales, un sincero agradecimiento.

Finalmente un especial agradecimiento al Ing. Alex Dávila Frías por su acertada dirección.

DEDICATORIA

Para mi esposa Ceci y nuestros hijos Santi y Dieguito, por su constante apoyo.

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS i
LISTA DE TABLAS ii
LISTA DE ANEXOS iii
RESUMEN iv
ABSTRACT v

1 INTRODUCCIÓN 1
1.1 PLANTEAMIENTO DEL PROBLEMA 2
1.2 FORMULACIÓN Y SITEMATIZACIÓN DEL PROBLEMA 3
1.2.1 Formulación 3
1.2.2 Sistematización 4
1.3 OBJETIVOS DE LA INVESTIGACION 4
1.3.1 Objetivo General.- 4
1.3.2 Objetivos Específicos.- 4
1.4 JUSTIFICACION DEL PROYECTO 5
1.4.1 Justificación Metodológica 5
1.4.2 Justificación Práctica.- 6
2 MARCO TEÓRICO 8
2.1 ANTECEDENTES 8
2.2 PRONÓSTICOS CON SUAVIZAMIENTO EXPONENCIAL.- 9
2.2.1 MÉTODO ESTACIONAL DE HOLT – WINTERS 11
2.2.1.1 Método de Holt-Winters con Estacionalidad Multiplicativa 11
2.2.1.2 Método de Holt-Winters con Estacionalidad Aditiva 11
2.3 METODOLOGÍA DE BOX-JENKINS 12
2.3.1 IDENTIFICACIÓN 12
2.3.2 ESTIMACIÓN 13
2.3.3 DIAGNÓSTICO 13
2.3.4 PRONÓSTICO 14
2.3.5 PARSIMONIA.- 15
2.3.6 ECUACIONES DEL MODELO ARIMA (p,d,q) 15
2.4 METODOLOGÍA DE REDES NEURONALES 16

2.4.1 BREVE INTRODUCCIÓN BIOLÓGICA 16
2.4.2 Aprendizaje 17
2.4.3 Cerebro y Computador 18
2.4.4 Estructura de un Sistema Neuronal Artificial 18
2.4.4.1 Modelo de Neurona Artificial 18
2.4.4.2 Modelo General de Neurona Artificial 19
2.4.5 Pronósticos con Redes Neuronales 20
2.4.6 Clasificación de los modelos neuronales 22
2.4.6.1 El Perceptrón Multicapa (MLP) 23
2.4.6.1.1 Aprendizaje por retro-propagación de errores (backpropagation) 25
2.4.6.1.2. Generalización de la Red 26
2.4.6.2 Redes Neuronales Dinámicas 26
2.5 ESTUDIOS ANTERIORES 27
2.6 MARCO CONCEPTUAL 28
3 METODOLOGÍA 31
3.1 ALCANCE DE LA INVETIGACIÓN 31
3.2 DISEÑO DE LA INVETIGACIÓN 31
3.3 SELECCIÓN DE LA MUESTRA 31
3.4 RECOLECCIÓN DE DATOS 32
3.5 ANÁLISIS DE DATOS 32
3.6 FUENTES DE INFORMACIÓN 32
4 METODOS DE SUAVIZAMIENTO EXPONENCIAL 33
4.1 INTRODUCCION 33
4.1.1 Estrategia para el pronóstico 34
4.2 FUNDAMENTO TEORICO DEL METODO DE HOLT - WINTERS 36
4.2.1 Suavizamiento Exponencial Simple 36
4.2.2 Metodo de Holt – Winters 37
4.2.2.1 Estacionalidad Multiplicativa 38
4.2.2.2 Estacionalidad Aditiva 40
4.2.2.3 Inicialización de Método de Holt - Winters 40
4.3 EJEMPLO PRÁCTICO DEL METODO DE HOLT – WINTERS 48

4.4 PRONOSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRAFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, CON EL METODO DE HOLT – WINTERS. 62
4.4.1 Gráfico de los datos 63
4.4.2 Examine Potenciales Comportamientos inusuales 64
4.4.3 Describir la forma de la variación exponencial 64
4.4.4 Calculo de Valores Iniciales 64
4.4.5 Chequeo de errores de pronóstico 65
4.4.6 Cálculo de Pronósticos a Futuro 75
5 METODOLOGIA DE BOX – JENKINS 77
5.1 INTRODUCCION 77
5.1.1 PRONOSTICO DE SERIES DE TIEMPO 78
5.1.2 Modelos matematicos dinamicos estocasticos y deterministicos 78
5.1.2.1 Modelos Estocásticos Estacionarios y No Estacionarios para Pronósticos 79
5.1.3 Modelos ARIMA comparados con otros modelos 82
5.2 FUNDAMENTO TEÓRICO DE LA METODOLOGÍA DE BOX-JENKINS .. 82
5.2.1 Conceptos fundamentales 82
5.2.1.1 Correlación 82
5.2.1.2 Diferencia 91
5.2.1.3 Desviación de la Media 92
5.2.1.4 Proceso, Realización y Modelo 92
5.2.1.5 Inferencia Estadística 95
5.2.1.6 Notación con el Operador de Retardo B 97
5.2.1.7 Prueba de Dickey – Fuller 99
5.2.1.8 Etapas de la Metodología de Box-Jenkins 100
5.3 EJEMPLOS PRÁCTICOS CON LA METODOLOGÍA DE BOX-JENKINS. 159
5.3.1 EJEMPLO #. 1 159
5.3.2 EJEMPLO #. 2 169
5.4 PRONÓSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRÁFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, CON LA METODOLOGÍA DE BOX - JENKINS 183
6 REDES NEURONALES 196
6.1 INTRODUCCION 196

6.1.1 Historia 197
6.1.2 Arquitectura 200
6.1.2.1 Redes Unidireccionales de una Sola Capa 200
6.1.2.2 Redes Unidireccionales Multicapa 201
6.1.2.3 Redes Recurrentes 202
6.1.3 Aplicaciones 204
6.2 FUNDAMENTO TEÓRICO DE LAS REDES NEURONALES 205
6.2.1 Modelo de neurona 205
6.2.1.1 Notación 205
6.2.1.2 Neurona con una sola entrada 205
6.2.1.3 Funciones de Transferencia 206
6.2.1.4 Neurona con Múltiples Entradas 209
6.2.2 Arquitectura de la red 210
6.2.2.1 Capa de Neuronas 210
6.2.2.2 Múltiples Capas de Neuronas 212
6.2.2.3 Redes Recurrentes 214
6.2.2.4 Reglas de Aprendizaje 215
6.2.3 Perceptrón Multicapa MLP 216
6.2.3.1 Clasificación de Patrones 217
6.2.3.2 Aproximador de Funciones 219
6.2.3.3 Algoritmo de Retropropagación 222
6.2.3.4 Utilización del Algoritmo de Retropropagación 228
6.2.3.5 Variaciones del Algoritmo de Retropropagación 234
6.2.4 Generalización 252
6.2.4.1 Planteamiento del Problema 254
6.2.4.2 Métodos para Mejorar la Generalización 256
6.2.5 Redes dinámicas y pronósticos 270
6.2.5.1 Redes Dinámicas Digitales Multicapa 271
6.2.5.2 Pronósticos 274
6.3 EJEMPLO PRÁCTICO CON EL METODO DE REDES NEURONALES 280
6.3.1 Descripción del sistema de levitación magnetico 280
6.3.2 Recolección de datos 281
6.3.3 Arquitectura de la red 283

6.3.4 Entrenamiento de la red 284
6.3.5 Validación de la red 286
6.3.6 Pronóstico de la red 289
6.4 PRONÓSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRÁFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, REDES NEURONALES 297
6.4.1 Datos para el pronóstico 297
6.4.2 Arquitectura de la red 298
6.4.3 Entrenamiento de la red 300
6.4.4 Validación de la red 303
6.4.5 Pronóstico de la red 306
7 RESULTADOS Y DISCUCIONES 313
7.1 RESULTADOS DE LOS TRES MÉTODOS DE PRONÓSTICOS 313
7.1.1 Método de Holt-Winters 313
7.1.2 Métodología de Box – Jenkins 314
7.1.3 Metodología de Redes Neuronales 315
7.2 PREDICCIÓN DEL RESTO DE FORMATOS DE PLACAS DIGITALES CON LA METODOLOGÍA DE BOX – JENKINS. 317
7.2.1 Predicción del Formato 525x450X0.15 (SM52) 317
7.2.2 Predicción del Formato 650X550X0.30 (kord-mo) 326
7.2.3 Predicción Formato 754X605X0.30 (pm74) 334
7.2.4 Predicción Formato 1030X790X0.30 (PM102) 343
8 CONCLUSIONES Y RECOMENDACIONES 353
8.1 CONCLUSIONES 353
8.2 RECOMENDACIONES 354
REFERENCIAS 356
ANEXOS 359
APÉNDICES 371

LISTA DE FIGURAS

Figura 2.1 Etapas en un enfoque iterativo en la construcción de modelos 14
Figura 2.2 Esquema de una neurona biológica típica 16
Figura 2.3 Modelo Genérico de Neurona Artificial 19
Figura 2.4 Esquema de una red neuronal con una capa oculta 21
Figura 2.5 Clasificación de los ANS por el tipo de Aprendizaje y Arquitectura 23
Figura 2.6 Arquitectura del Perceptrón Multicapa MLP 24
Figura 4.1 Patrones basados en la clasificación de Pegel (1969) 34
Figura 4.2 Estrategia para evaluar los métodos de suavizamiento para Pronósticos 35
Figura 4.3 Gráfico Ejemplo Método de Holt – Winters 49
Figura 4.4 Optimización MSE con Solver 51
Figura 4.5 Gráfico Descomposición Clásica del Ejemplo 55
Figura 4.6 Gráfico Pronóstico Método de Holt – Winters Ejemplo) 58
Figura 4.7 Gráfico de Autocorrelación del Error del Ejemplo. 60
Figura 4.8 Gráfico de Autocorrelación Parcial del Error del Ejemplo. 61
Figura 4.9 Distribución de Probabilidad del Error del Ejemplo 61
Figura 4.10 Consumo de placas formato 510x400x0.15 (GTO_52) 63
Figura 4.11 Pronóstico de placas formato 510x400x0.15 (GTO_52) 65
Figura 4.12 Histograma del Error de Pronóstico del Formato 510x400x0.15. 67
Figura 4.13 Prueba de Normalidad Error de Pronóstico del Formato 510x400x0.15 67
Figura 4.14 Función de Autocorrelación del Error de Pronóstico de Placas Digitales. 68
Figura 4.15 ACF del error Simulación # 1 70
Figura 4.16 ACF del error Simulación # 2 71
Figura 4.17 ACF del error Simulación # 3 72
Figura 4.18 ACF del error Simulación # 4 73
Figura 4.19 Pronóstico de placas formato 510x400x0.15 (GTO_52) Final. 74
Figura 4.20 Histograma del Error de Pronóstico del Formato 510x400x0.15 Final 75
Figura 5.1 Etapas en un enfoque iterativo en la construcción de modelos 81
Figura 5.2 Ejemplos de acfs y pacfs teóricas para dos procesos AR(1) 103

Figura 5.3 Ejemplos de acfs y pacfs teóricas para procesos AR(2) 105
Figura 5.4 Funciones típicas acfs y pacfs teóricas para procesos AR(2) 106
Figura 5.5 Ejemplos de acfs y pacfs teóricas para procesos MA(1) 109
Figura 5.6 Funciones típicas acfs y pacfs teóricas para procesos MA(2) 110
Figura 5.7 Ejemplos de acfs y pacfs teóricas para procesos MA(2) 111
Figura 5.8 Funciones acfs y pacfs teóricas para varios procesos ARMA(1,1) 112
Figura 5.9 Ejemplos de acfs y pacfs teóricas para procesos ARMA(1,1) 113
Figura 5.10 Serie con media No Estacionaria 116
Figura 5.11 Serie con Varianza No Estacionaria 118
Figura 5.12 Resultados Estimación Ejemplo ARMA(1,1) 123
Figura 5.13 Pronóstico de un Modelo AR(1) 139
Figura 5.14 Pronóstico de un Modelo MA(1) 140
Figura 5.15 Pronóstico de un Modelo ARIMA(0,1,1) 141
Figura 5.16 Pronóstico de un Modelo ARIMA(0,2,1) 141
Figura 5.17 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984 149
Figura 5.18 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984 d=1 149
Figura 5.19 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984 d=D=1 150
Figura 5.20 ACF y PACF teóricas para un proceso estacional $AR(1)_4$ con $\emptyset_4 = 0.8$ 151
Figura 5.21 ACF y PACF teóricas para un proceso estacional $MA(1)_4$ con $\theta_4 = 0.8$ 152
Figura 5.22 Función ACF teórica para un proceso estacional NO estacionario 153
Figura 5.23 Ejemplo # 1 Permisos de Construcción desde 1947 a 1967 160
Figura 5.24 Ejemplo # 1 Función de Autocorrelación Muestral SACF 161
Figura 5.25 Ejemplo # 1 Función de Autocorrelación Parcial Muestral SPACF 162
Figura 5.26 Ejemplo # 1 Función de Autocorrelación Primeras Diferencias 163
Figura 5.27 Ejemplo # 1 Función de Autocorrelación Modelo MA(2) 164
Figura 5.28 Ejemplo # 1 Función de Autocorrelación Parcial Modelo MA(2) 164
Figura 5.29 Ejemplo # 1 Función de Autocorrelación del Modelo AR(2) 165
Figura 5.30 Ejemplo # 1 Correlograma del Modelo ARMA(2,2) 167
Figura 5.31 Ejemplo # 1 Pronósticos con Modelo ARMA(2,2) 169
Figura 5.32 Ejemplo # 2 Ventas de un Producto desde 1997 hasta 2005 170
Figura 5.33 Ejemplo # 2 Función de Autocorrelación Muestral SACF 171
Figura 5.34 Ejemplo # 2 Función de Autocorrelación Parcial SPACF 172

Figura 5.36 Ejemplo # 2 SACFde la serie con una diferencia estacional 173
Figura 5.37 Ejemplo # 2 SPACF de la serie con una diferencia estacional 173
Figura 5.38 Correlograma Modelo 1 Ejemplo2 175
Figura 5.39 Correlograma Modelo 2 Ejemplo 2 176
Figura 5.40 Correlograma Modelo 3 Ejemplo 2 177
Figura 5.41 Correlograma Modelo 4 Ejemplo 2 178
Figura 5.42 Correlograma Modelo 5 Ejemplo 2 180
Figura 5.43 Inverso de las Raices Modelo 3 Ejemplo 2 181
Figura 5.44 Pronóstico, Intervalo de Confianza y Valores Reales Modelo 3 Ejemplo 2 182
Figura 5.45 Consumo de placas formato 510x400x0.15 (GTO_52) (2009 – 2015) 184
Figura 5.46 Autocorrelación Muestral Consumo de placas formato GTO_52 185
Figura 5.47 Autocorrelación Parcial Consumo de placas digitales formato GTO_52 186
Figura 5.48 Primera diferencia de la serie: consumo de placas formato GTO_52 187
Figura 5.49 SACF Primera diferencia de la serie: consumo de placas formato GTO_52 187
Figura 5.50 SPACF Primera diferencia de la serie: consumo de placas formato GTO_52188
Figura 5.51 SACF de los residuos del Modelo 1 para el consumo de placas GTO_52 189
Figura 5.52 SPACF de los residuos del Modelo 1 para el consumo de placas GTO_52 189
Figura 5.53 SACF de los residuos del Modelo 2 para el consumo de placas GTO_52 190
Figura 5.54 SPACF de los residuos del Modelo 2 para el consumo de placas GTO_52 191
Figura 5.55 SACF de los residuos del Modelo 3 para el consumo de placas GTO_52 192
Figura 5.56 SPACF de los residuos del Modelo 3 para el consumo de placas GTO_52 192
Figura 6.1 Red Unidireccional de una sola capa 201
Figura 6.2 Red Unidireccional Multicapa 202
Figura 6.3 Red Recurrente sin autorealimentación y sin neuronas ocultas 203
Figura 6.4 Red Recurrente con neuronas ocultas 203
Figura 6.5 Neurona con una sola entrada 205
Figura 6.6 Función de transferencia *hardlim* 207
Figura 6.7 Función de transferencia *lineal* 207
Figura 6.8 Función de transferencia *log-sigmoid* 207
Figura 6.9 Neurona con múltiples entradas 209
Figura 6.10 Notación abreviada de una neurona con R entradas 210
Figura 6.11 Capa con *S* neuronas 211

Figura 6.12 Notación Abreviada de una capa con S neuronas y R entradas 212
Figura 6.13 Red de tres capas 212
Figura 6.14 Red de tres capas, Notación Abreviada 213
Figura 6.15 Red Recurrente 214
Figura 6.16 Bloque de Retardo 215
Figura 6.17 Perceptrón de tres capas 217
Figura 6.18 Problema de O Exclusivo (XOR) 218
Figura 6.19 Secciones de Decisión Red XOR 218
Figura 6.20 Región de Decisión Red Neuronal 2-2-1 219
Figura 6.21 Red Neuronal 2-2-1 para resolver el problema del XOR 219
Figura 6.22 Red Neuronal 1-2-1 Aproximador de Funciones 220
Figura 6.23 Respuesta de la red de la figura 6.22 220
Figura 6.24 Efectos al cambiar los parámetros de la red 221
Figura 6.25 Notación Abreviada de una red de tres capas 222
Figura 6.26 Red 1-3-1 Aproximando a la función Seno 229
Figura 6.27 Efecto al incrementar el número de neuronas en la capa oculta 230
Figura 6.28 Convergencia hacia un Mínimo Global 231
Figura 6.29 Convergencia hacia un Mínimo Local 232
Figura 6.30 Aproximación a la función $g(p)$ con una Red 1-2-1. 233
Figura 6.31 Aproximación a la función $g(p)$ con una Red 1-9-1 233
Figura 6.32 Dos Trayectorias del SDBP (Versión por lotes) 237
Figura 6.33 Error Cuadrático de las trayectorias “a” y “b” 238
Figura 6.34 Trayectoria con un tasa de aprendizaje grande 239
Figura 6.35 Efecto de Suavizamiento por Inercia 240
Figura 6.36 Trayectoria con Inercia 241
Figura 6.37 Trayectoria con Tasa de Aprendizaje Variable 245
Figura 6.38 Posibles Direcciones Algoritmo de Levenberg-Marquardt 251
Figura 6.39 Trayectoria LMBP 251
Figura 6.40 Ejemplo de Sobreajuste y Pobre Generalización 255
Figura 6.41 Ejemplo de Buena Interpolación y Pobre Extrapolación 256
Figura 6.42 Para Temprana 258
Figura 6.43 Ejemplos de Parada Temprada 259

Figura 6.44 Efecto de los Pesos en la Respuesta de la Red 261
Figura 6.45 Efecto de la Relación de Regularización $\frac{\alpha}{\beta}$ 262
Figura 6.46 Ajuste con Regularización Bayesiana 269
Figura 6.47 Ejemplo Red Neuronal Dinámica 272
Figura 6.48 Red Neuronal FTDNN 274
Figura 6.49 Red Neuronal NARX 275
Figura 6.50 Red Neuronal NARX: Arquitectura Paralela y Serie – Paralelo 276
Figura 6.51 $R_e(\tau)$ para una red entrenada inadecuadamente 277
Figura 6.52 $R_e(\tau)$ para una red entrenada correctamente 278
Figura 6.53 $R_{pe}(\tau)$ para una red entrenada inadecuadamente 279
Figura 6.54 $R_{pe}(\tau)$ para una red entrenada correctamente 279
Figura 6.55 Sistema de Levitación Magnético 281
Figura 6.56 Corriente de Entrada 282
Figura 6.57 Posición del Magneto 282
Figura 6.58 Arquitectura de la Red NARX 283
Figura 6.59 MSE Regularización Bayesiana 285
Figura 6.60 Pantalla de Entrenamiento Reg. Bayesiana 286
Figura 6.61 Regresión Reg. Bayesiana 287
Figura 6.62 Autocorrelación del error Reg. Bayesiana 288
Figura 6.63 Correlación Cruzada Reg. Bayesiana 289
Figura 6.64 Serie de Tiempo con Reg. Bayesiana y sus Errores 290
Figura 6.65 Esquema configuración Serie –Paralelo para Pronósticos 290
Figura 6.66 Esquema configuración Paralelo para Pronósticos 291
Figura 6.67 Pronóstico 200 pasos adelante configuración Paralelo 292
Figura 6.68 Pronóstico 200 pasos adelante configuración Paralelo Ampliado 292
Figura 6.69 MSE Algoritmo Levenberg-Marquardt 293
Figura 6.70 Autocorrelación del error con Levenberg-Marquardt 294
Figura 6.71 Correlación Cruzada con Levenberg-Marquardt 295
Figura 6.72 Pronóstico 200 pasos adelante configuración Paralelo con LM 295
Figura 6.73 Pronóstico 200 pasos adelante configuración Paralelo con LM Ampliado 296
Figura 6.74 Consumo de placas formato 510x400x0.15 (GTO_52) (2009 – 2015) 298
Figura 6.75 Arquitectura de una Red Tipo NAR 299

Figura 6.76 MSE Regularización Bayesiana 301
Figura 6.77 Pantalla de Entrenamiento Reg. Bayesiana 303
Figura 6.78 Regresión Reg. Bayesiana 304
Figura 6.79 Autocorrelación del error Reg. Bayesiana 305
Figura 6.80 Serie de Tiempo con Reg. Bayesiana y sus Errores 306
Figura 6.81 Esquema configuración Serie –Paralelo para Pronósticos 307
Figura 6.82 Esquema configuración Paralelo para Pronósticos 307
Figura 6.83 Pronóstico 6 pasos adelante configuración Paralelo 308
Figura 6.84 Pronóstico 6 pasos adelante configuración Paralelo Ampliado 308
Figura 6.85 Resultado Numérico del Pronóstico 6 pasos adelante 309
Figura 6.86 MSE Algoritmo Levenberg-Marquardt 310
Figura 6.87 Autocorrelación del error con Levenberg-Marquardt 310
Figura 6.88 Pronóstico 6 pasos adelante configuración Paralelo con LM 311
Figura 6.89 Pronóstico 6 pasos adelante configuración Paralelo con LM Ampliado 311
Figura 7.1 Pronóstico con el método de Holt – Winters vs Real 313
Figura 7.2 Pronóstico con la metodología ARIMA vs Real 314
Figura 7.3 Pronóstico con Redes Neuronales vs Real 315
Figura 7.4 Pronóstico con los tres métodos vs Real 316
Figura 7.5 Consumo de placas digitales formato 525x459x0.15 (SM_52) (2009 – 2015) 318
Figura 7.6 Autocorrelación Muestral del Consumo de placas digitales formato SM_52 320
Figura 7.7 Autocorrelación Parcial del Consumo de placas digitales formato SM_52 321
Figura 7.8 Autocorrelación residuos modelo ARMA(1,1), formato SM_52 322
Figura 7.9 Autocorrelación parcial residuos modelo ARMA(1,1), formato SM_52 322
Figura 7.10 Autocorrelación residuos modelo ARIMA(0,1,1), formato SM_52 323
Figura 7.11 Autocorrelación parcial residuos modelo ARIMA(0,1,1), formato SM_52 324
Figura 7.12 Pronósticos con modelo ARIMA(0,1,1), formato SM_52 325
Figura 7.13 Consumo de placas formato 650x550x0.30 (KORD_MO) (2009 – 2015) 327
Figura 7.14 Correlograma del Consumo de placas digitales formato KORD_MO 329
Figura 7.15 Correlograma diferencia estacional del Consumo del formato KORD_MO 330
Figura 7.16 Correlograma Modelo 1 331
Figura 7.17 Correlograma Modelo 2 333
Figura 7.18 Raíces inversas del modelo 2 333

Figura 7.19 Consumo de placas formato 745x605x0.30 (PM_74) (2009 – 2015) 336
Figura 7.20 Correlograma del Consumo del formato PM_74 del 2009 al 2015 336
Figura 7.21 Diferencia regular consumo formato PM_74 del 2009 al 2015 338
Figura 7.22 Logaritmo Natural consumo formato PM_74 del 2009 al 2015 338
Figura 7.23 Correlograma del logaritmo del consumo del formato PM_74 339
Figura 7.24 Correlograma residuos modelo Log. ARIMA(0,1,1), formato PM_74 340
Figura 7.25 Correlograma residuos modelo ARIMA(0,1,1), formato PM_74 342
Figura 7.26 Consumo de placas formato 1030x790x0.30 (SM_102) (2009 – 2015) 345
Figura 7.27 Autocorrelación del Consumo de placas digitales formato SM_102 347
Figura 7.28 Autocorrelación Parcial del Consumo de placas digitales formato SM_102 347
Figura 7.29 Gráfico de la serie después de una diferencia estacional formato SM_102 348
Figura 7.30 Autocorrelación de la serie con diferencia estacional formato SM_102 349
Figura 7.31 Autocorrelación parcial de la serie con diferencia estacional SM_102 349
Figura 7.32 Autocorrelación residuos $ARIMA(1,0,0)(1,1,0)_{12}$ Formato SM_102 350
Figura 7.33 Autocorrelación parcial residuos $ARIMA(1,0,0)(1,1,0)_{12}$ SM_102 351
Figura 7.34 Gráfico Pronósticos Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ Formato SM_102 352
Figura A.1 Función de densidad de una variable aleatoria continua A.1
Figura A.2 a) Asimetría y b) curtosis de una distribución de probabilidad A.11
Figura A.3 Intervalo de confianza al 95% para μ A.14
Figura A.4 Distribución del estadístico Z A.15
Figura D.1 Optimización con el método del gradiente descendente D.4
Figura D.2 Optimización con el método de Newton D.6
Figura D.3 Optimización con el método de Gradiente Conjugado D.9
Figura D.4 Red ADALINE D.10

LISTA DE TABLAS

Tabla 4.1 Datos Ejemplo Método de Holt – Winters 48
Tabla 4.2 Ejemplo Método de Holt – Winters Propuesto por Makridakis 50
Tabla 4.3 Pronóstico hacia atrás (Backcasting) del Ejemplo 52
Tabla 4.4 Pronóstico utilizando inicialización con Backcasting 53
Tabla 4.5 Descomposición Clásica del Ejemplo 54
Tabla 4.6 Regresión Lineal Ejemplo con Estacionalidad Ajustada 55
Tabla 4.7 Pronóstico Inicializando con Descomposición Clásica 56
Tabla 4.8 Pronóstico con Método de Holt – Winters Amortiguado 57
Tabla 4.9 Resumen de errores producidos por los diferentes métodos de inicialización 58
Tabla 4.10 Pronóstico con Programa Minitab 17 Ejemplo 59
Tabla 4.11 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015 62
Tabla 4.12 Pronóstico de Placas Formato 510x400x0.15 desde el 2009 hasta 2015. 66
Tabla 4.13 Método de Holt – Winters Multiplicativo (Original) 69
Tabla 4.14 Método de Holt – Winters Multiplicativo (Simulación # 1) 70
Tabla 4.15 Método de Holt – Winters Multiplicativo (Simulación # 2) 71
Tabla 4.16 Método de Holt – Winters Multiplicativo (Simulación # 3) 72
Tabla 4.17 Método de Holt – Winters Multiplicativo (Simulación # 4) 73
Tabla 4.18 Pronóstico a Futuro Método de Holt – Winters Multiplicativo Final 75
Tabla 4.19 Datos Pronóstico a Futuro Método de Holt – Winters Multiplicativo Final 76
Tabla 5.1 EACF Teórica para un modelo ARIMA(1,1,1), o ARIMA(2,0,1), o ARIMA(0,2,1) 89
Tabla 5.2 Características de un buen modelo ARIMA 94
Tabla 5.3 Ejemplo 1 Permisos de Construcción de Viviendas desde 1947 a 1967 USA 159
Tabla 5.4 Prueba de Dickey Fuller Aumentada Ejemplo # 1 160
Tabla 5.5 Estimación Modelo ARMA(2,2) Ejemplo # 1 166
Tabla 5.6 Inverso de las raíces modelo ARMA(2,2) Ejemplo # 1 167
Tabla 5.7 Pronóstico Modelo ARMA(2,2) Ejemplo # 1 año 1968 168
Tabla 5.8 Datos Ejemplo # 2 Ventas de un producto desde 1997 hasta 2006 169
Tabla 5.9 Resultados Estimación Modelo 1 174

Tabla 5.10 Resultados Estimación Modelo 2 175
Tabla 5.11 Resultados Estimación Modelo 3 177
Tabla 5.12 Resultados Estimación Modelo 4 178
Tabla 5.13 Resultados Estimación Modelo 5 179
Tabla 5.14 Comparación de Modelos Ejemplo 2 180
Tabla 5.15 Pronóstico, Valores Reales, Intervalo de Confianza, obtenido con el Modelo 3 182
Tabla 5.16 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015 183
Tabla 5.17 Prueba DFA del Consumo de Placas Digitales Formato 510x400x0.15 185
Tabla 5.18 Pronósticos y errores Modelo 1 193
Tabla 5.19 Pronósticos y errores Modelo 2 193
Tabla 5.20 Pronósticos y errores Modelo 3 193
Tabla 5.21 Coeficientes del Modelo 3 194
Tabla 5.22 Estadístico de Ljung – Box Q^* para el Modelo 3 194
Tabla 5.23 Matriz de Correlación de los parámetros estimados para el Modelo 3 194
Tabla 5.24 Pronósticos y errores Modelo 3 195
Tabla 6.1 Funciones de transferencia en Mat Lab 208
Tabla 6.2 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015 297
Tabla 6.3 Pronósticos y Errores Regularización Bayesiana 312
Tabla 6.4 Pronósticos y Errores Levenberg-Marquardt 312
Tabla 7.1 Mejor Predicción con el Método de Holt-Winters 313
Tabla 7.2 Mejor Predicción con el Método de Box-Jenkins 314
Tabla 7.3 Mejor Predicción con el Método de Redes Neuronales 315
Tabla 7.4 Consumo de Placas Formato 525x459x0.15 desde el 2009 hasta 2015 317
Tabla 7.5 Prueba DFA del Consumo de Placas Formato 525x459x0.15 319
Tabla 7.6 Estimación modelo ARMA (1,1) Formato 525x459x0.15 (SM_52) 321
Tabla 7.7 Estimación modelo ARIMA (0,1,1) Formato 525x459x0.15 (SM_52) 323
Tabla 7.8 Estadístico de Ljung – Box Q^* para el Modelo ARIMA(0,1,1) SM_52 324
Tabla 7.9 Pronósticos y errores Modelo ARIMA (0,1,1) SM_52 325
Tabla 7.10 Consumo de Placas Formato 650x550x0.30 desde el 2009 hasta 2015 326
Tabla 7.11 Prueba DFA del Consumo de Placas Digitales Formato 650x550x0.30 328
Tabla 7.12 Estimación del modelo 1 331

Tabla 7.13 Estimación modelo 2 332
Tabla 7.14 Pronósticos y errores modelo 2 334
Tabla 7.15 Consumo de Placas Formato 745x605x0.30 desde el 2009 hasta 2015 335
Tabla 7.16 Prueba DFA del Consumo de Placas Digitales Formato 745x605x0.30 337
Tabla 7.17 Estimación modelo ARIMA (0,1,1) Formato PM_74 340
Tabla 7.18 Pronósticos y errores Modelo ARIMA (0,1,1) para formato PM_74 341
Tabla 7.19 Estimación Modelo ARIMA (0,1,1) PM_74 342
Tabla 7.20 Pronósticos y errores Modelo ARIMA (0,1,1) PM_74 343
Tabla 7.21 Consumo de Placas Formato 1030x790x0.30 desde el 2009 hasta 2015 344
Tabla 7.22 Prueba DFA del Consumo de Placas Digitales Formato 1030x790x0.30 346
Tabla 7.23 Estimación modelo $ARIMA(1,0,0)(1,1,0)_{12}$Formato SM_102 350
Tabla 7.24 Estadístico de Ljung – Box Q^* $ARIMA(1,0,0)(1,1,0)_{12}$Formato SM_102 350
Tabla 7.25 Pronósticos y errores del Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ SM_102 352

LISTA DE ANEXOS

ANEXO A - Modelo de la orden de encuadernación 360
ANEXO B - Información del consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2105 361
ANEXO C - Tablas de Distribución de Probabilidad 364
ANEXO D - Programa: Test_Mag_Data_NARX.m (Redes Neuronales) 369
ANEXO E - Programa: Prog_Narnet_Tesis_OK.m (Redes Neuronales) 370

LISTA DE APÉNDICES

APÉNDICE A – Repaso de Conceptos Estadísticos A.1
APÉNDICE B – Nociones Básicas de Algebra Matricial. B.1
APÉNDICE C – Ecuaciones de Diferencias y Operador de Retardo C.1
APÉNDICE D – Métodos de Estimación y Optimización D.1

RESUMEN

Todos los proveedores quiteños sin excepción se han quedado sin inventario de algún insumo o materia prima esencial para el proceso de producción de impresos. Esto ha causado un malestar generalizado en sus clientes y los proveedores han sufrido pérdidas económicas cuantiosas. El problema radica en la mala predicción de la demanda, he aquí la importancia de utilizar métodos de pronósticos acordes con esta demanda.

Una buena predicción de la demanda de insumos gráficos permite a las empresas distribuidoras tener inventarios óptimos, liberando recursos para invertir en otras líneas de negocios, que a fin de cuentas mejorarán la rentabilidad de la misma. Por otra parte mantiene a sus clientes operando de manera estable, sin permitir el ingreso de la competencia.

La presente tesis realiza una comparación entre tres métodos de pronósticos avanzados los mismos que servirán como herramientas para la predicción de placas digitales en el mercado gráfico quiteño, la información se ha recolectado desde el año 2009 hasta el año 2015 de una muestra significativa de empresas de este sector.

Se estudian métodos de pronósticos avanzados basados en series de tiempo ya que el tipo de demanda de este insumo constituye un proceso estocástico no estacionario, que además contiene tendencia y estacionalidad, lo que complica su predicción con métodos de pronósticos tradicionales.

Se comparán tres métodos de pronósticos avanzados como son: método de Holt – Winters, Box – Jenkins y Redes Neuronales.

Las placas digitales tienen varios formatos acorde con la máquina de impresión en la que se va a imprimir, se realizó el pronóstico de un solo formato (GTO_52) para la comparación entre los métodos de pronósticos avanzados, los errores que se consiguieron fueron: método de Holt-Winters con un MAPE de 4.46%, Box-Jenkins con MAPE de 3.67% y Redes Neuronales con un MAPE de 4.12%, por lo tanto el mejor método fue el de Box – Jenkins, ya que produjo el menor error porcentual

absoluto medio (MAPE). Por esta razón se utilizará este método para la predicción del resto de formatos.

Palabras clave: Procesos No Estacionarios, Pronósticos, Holt – Winters, Box – Jenkins, Redes Neuronales Artificiales.

ABSTRACT

All providers of Quito without exception have run out of inventory of some supplies or material essential to the printing process. This has caused an overall bad feeling in its customers and suppliers have suffered big economic losses. The problem is poor demand forecasting, here is the importance of using forecasting methods consistent with this demand.

A good prediction of demand for graphics supplies allows distribution companies to have optimal inventories, freeing up resources to invest in other business lines that ultimately improve profitability. Moreover keeps your customers operate stably without allowing that competition access to them.

This thesis makes a comparison between three methods of advanced forecasting methods, they will serve as tools for the prediction of digital plates in the printing market of Quito and information has been collected since 2009 until 2015 of a significant sample of sector's companies.

Advanced forecasting methods based on time series are studied, because the demand for this type of supply is a non-stationary stochastic process, which also contains trend and seasonality, complicating its forecast with traditional methods of forecasting.

Three advanced methods: Holt - Winters, Box - Jenkins and Neural Networks: are compared.

Digital plates have various formats according to the printing machine which is to be printed, perform the forecast of a single format (GTO_52) for the comparison between the methods of advanced forecasting, errors that were achieved was: method Holt-Winters with MAPE of 4.46%, Box-Jenkins with MAPE of 3.67% and Neural Networks with MAPE of 4.12%, therefore the best method was the Box - Jenkins, as it produced the lowest mean absolute percentage error (MAPE). For this reason this method for predicting other formats are used.

Keywords: non-stationary processes, Forecasts, Holt - Winters, Box - Jenkins, Artificial Neural Networks

1 INTRODUCCIÓN

Pronosticar valores futuros en base a series de tiempo es un problema importante en muchas áreas como, la economía, planificación de la producción, pronósticos de ventas y control de la demanda e inventarios.

Existen una gran cantidad de métodos de pronósticos hoy en día disponibles, es muy importante entender que un solo método no es universalmente aplicable, la elección del método dependerá de un conjunto de condiciones dadas. Así los pronósticos no son sagrados o la última palabra se debe estar preparado para modificarlos a la luz de cualquier información externa. (Chatfield, 2012 Sixth Edition)

No cabe ninguna duda que pronosticar es una tarea difícil pero quién obtenga el menor error es quién obtendrá una ventaja competitiva. (Hyndman R., 2014)

El pronóstico es un insumo clave en la planificación y el control de los negocios así: el marketing recurre a los pronósticos para la política de precios, planificación de productos y la promoción, las finanzas lo utilizan en la planificación financiera, las operaciones lo utilizan para tomar decisiones de: diseño de procesos, planificación de la capacidad y control de inventarios. (Schroeder R., 2011)

Básicamente existen dos métodos de pronósticos **los cualitativos** y **los cuantitativos.** Los primeros se basan en criterios administrativos y no utilizan modelos específicos, se los utiliza cuando existe una falta de datos o los datos disponibles no son confiables a futuro. (Schroeder R., 2011)

Los métodos cuantitativos utilizan modelos matemáticos fundamentales para llegar a un pronóstico. El supuesto básico aquí es que se puede predecir el futuro en base a datos del pasado. (Schroeder R., 2011)

En el presente trabajo se analizarán los pronósticos de la demanda de uno de los insumos para la industria gráfica en el mercado quiteño, las placas digitales. Se ha recolectado información de una muestra significativa de empresas consumidoras de este insumo, para así formar una serie de tiempo a partir del año 2009 hasta el 2015 con esta información realizaremos las predicciones, que servirán para el manejo del inventario de las empresas distribuidoras de estos insumos.

Estos insumos (las placas digitales) no se producen en nuestro país razón por la cual tienen que ser importados y estas importaciones tardan de tres a cuatro meses

en llegar al país, por esta razón es importante tener una alta precisión en los pronósticos y estos son a corto plazo (menor a seis meses).

Los expertos recomiendan utilizar series de tiempo (método cuantitativo) cuando se necesita una alta precisión en los pronósticos y el horizonte de tiempo es corto. (Schroeder R., 2011)

Los pronósticos han sido dominados durante varias décadas por los métodos lineales por su facilidad en el desarrollo, implementación e interpretación. Pero estos modelos tienen serias limitaciones cuando las relaciones entre sus datos son no lineales, ya que la aproximación de modelos lineales a complicadas relaciones no lineales de datos no siempre es satisfactoria. (Zhang, 2004).

Es por esto que en la presente tesis se estudiarán tanto los métodos lineales (Holt-Winters y Box – Jennkins) como los no lineales (Redes Neuronales Artificiales).

Los métodos lineales de pronósticos se estudiarán en los capítulos 4 y 5, para en el capítulo 6 estudiar los modelos no lineales como son las redes neuronales artificiales finalmente en el capítulo 8 se sugerirán recomendaciones y conclusiones.

Se termina esta introducción con una cita famosa de uno de los genios de la física Niels Bohr: “La predicción es muy difícil, especialmente cuando es acerca del futuro”.

1.1 PLANTEAMIENTO DEL PROBLEMA

La industria gráfica ecuatoriana utiliza los siguientes insumos básicos: papel, tinta, placas y químicos, los mismos que en su totalidad son importados. De aquí se deriva la importancia de los proveedores de insumos gráficos, ya que la falta de uno de estos insumos podría parar completamente en proceso de impresión.

Después de una observación directa y varias entrevistas con proveedores gráficos en la ciudad de Quito se determinó que existen alrededor de 80 empresas con equipos de última tecnología para la producción de placas digitales, algunos de ellos son imprentas que utilizan los equipos para su propio consumo y otras son empresas que brindan el servicio de elaboración de las placas digitales a las imprentas medianas y pequeñas.

Los proveedores de insumos gráficos, han tenido un problema latente, la falta de inventario de productos en ciertas épocas del año.

Todos los proveedores sin excepción se han quedado sin inventario de algún insumo o materia prima esencial para el proceso de producción de impresos. Esto ha causado un malestar generalizado en sus clientes, los proveedores han sufrido pérdidas económicas cuantiosas y lo peor: la pérdida de la lealtad de sus clientes.

El problema radica en el tipo de demanda, son procesos No Estacionarios (sus distribuciones de probabilidad no se mantienen estables con el paso del tiempo), es decir muy difíciles de predecir sin métodos avanzados de pronósticos. Además, tienen tendencia y estacionalidad, que complican aún más su predicción.

El propósito de la presente tesis es describir y comparar varios métodos de pronósticos basados en series de tiempo para predecir la demanda de placas digitales en empresas consumidoras del sector gráfico quiteño.

Se contrastarán tres métodos de pronósticos avanzados como son: método de Holt – Winters, Box – Jenkins y Redes Neuronales Artificiales.

Se han considerado estos métodos ya que son capaces de predecir procesos estocásticos no estacionarios.

El estudio se llevará a cabo en una muestra heterogénea de 52 empresas de la ciudad de Quito (de un total aproximado de 80 empresas que producen placas en el mercado quiteño), las mismas que nos proporcionarán información confiable del consumo de sus placas digitales desde el año 2009 hasta el año 2015.

Con estos datos podremos comparar los tres métodos de pronósticos y finalmente probar cuál de ellos es el más preciso para el tipo de demanda de estas empresas.

1.2 FORMULACIÓN Y SITEMATIZACIÓN DEL PROBLEMA

1.2.1 FORMULACIÓN

¿Cuál de los siguientes métodos de pronóstico basado en series de tiempo como son: Holt–Winters, Box-Jenkins y Perceptrón Multicapa (MLP) (Red Neuronal), producen la mejor predicción en empresas que utilizan placas digitales en el mercado gráfico quiteño desde el año 2009 hasta el año 2015?

1.2.2 SISTEMATIZACIÓN

- ✓ ¿La metodología de Holt–Winters es aplicable en la predicción del consumo de placas digitales del sector gráfico quiteño?
- ✓ ¿Cómo la metodología de Box–Jenkins resuelve el problema de la predicción de placas digitales del sector gráfico quiteño?
- ✓ ¿La metodología de Redes Neuronales (Perceptrón Multicapa MLP) es aplicable en la predicción del consumo de placas digitales del sector gráfico quiteño?
- ✓ ¿Cómo se medirá el desempeño de los diferentes modelos de pronósticos con datos reales del consumo de placas digitales en el mercado gráfico quiteño?

1.3 OBJETIVOS DE LA INVESTIGACION

1.3.1 OBJETIVO GENERAL.-

Determinar cuál de los siguientes métodos de pronóstico basado en series de tiempo como son: Holt –Winters, Box-Jenkins y Perceptrón Multicapa (MLP) (Red Neuronal), producen la mejor predicción en empresas que utilizan placas digitales en el mercado gráfico quiteño desde el año 2009 hasta el año 2015.

1.3.2 OBJETIVOS ESPECÍFICOS.-

- ✓ Examinar la metodología de Holt – Winters en la predicción del consumo de placas digitales del sector gráfico quiteño.
- ✓ Analizar la metodología de Box – Jenkins para el pronóstico del consumo de placas digitales en el sector gráfico quiteño.
- ✓ Estudiar la metodología de Redes Neuronales (Perceptrón Multicapa MLP) en la predicción del consumo de placas digitales del sector gráfico quiteño

- ✓ Medir el desempeño de los diferentes modelos de pronósticos con datos reales del consumo de placas digitales en el mercado gráfico quiteño
- ✓ Realizar los pronósticos con datos reales del consumo de placas digitales en el mercado gráfico quiteño, con el método que mejor desempeño mostró en la presente investigación.

1.4 JUSTIFICACION DEL PROYECTO

1.4.1 JUSTIFICACIÓN METODOLÓGICA

El alcance del presente trabajo es descriptivo, ya que pretendemos medir el error de pronóstico para determinar cuál metodología es más exacta en la predicción de placas digitales en el mercado gráfico quiteño desde el 2009 hasta el 2015.

Se tratará de responder a las preguntas de investigación mediante un diseño de investigación no experimental ya que no existirá manipulación alguna de las variables y solo se observará la demanda de placas digitales en un ambiente natural para calcular los diferentes errores de pronósticos.

Además seguiremos esta demanda y el error de los pronósticos con el paso del tiempo con datos desde el 2009 hasta el 2015 por lo que se enmarca en el tipo de diseño de investigación longitudinal de panel.

Resumiendo tendríamos un diseño de investigación: No experimental longitudinal de panel.

La muestra está claramente limitada de la población de insumos gráficos (tinta, placas, papel, químicos y otros) hemos escogido para este estudio solamente las placas digitales para el mercado de la ciudad de Quito desde el año 2009 hasta el 2015. Además en la ciudad de Quito se cuenta con alrededor de 80 empresas con equipos de última tecnología para la producción de placas digitales, hemos escogido una muestra NO probabilística de 52 empresas ya que ellas han podido suministrar información válida y confiable para el presente trabajo.

La recolección de datos se realizó mediante visitas personales (observación) a los clientes que utilizan las placas digitales, para garantizar la validez, confiablidad y objetividad de la información.

El análisis de datos se realizará mediante varios programas estadísticos como son: Minitab, EViews y Matlab, los que nos servirán para estimar los diferentes parámetros de los modelos de las diferentes metodologías, para así calcular el error de pronóstico MAPE y finalmente presentar los resultados.

Las fuentes de información son básicamente primarias, se ha recogido información de libros especializados y artículos de revistas especializadas.

Después de realizar la predicción correcta de la demanda, se necesitará determinar cuál de los métodos produce el pronóstico más exacto, para determinar la exactitud del pronóstico se pueden calcular varios tipos de errores, los comúnmente utilizados son: MSE (Error Cuadrático Medio) y MAPE (Error Porcentual Absoluto Medio).

Ya que el MSE es muy sensible a la distorsión por la presencia de Outliers (comportamientos inusuales), preferiremos el Error Porcentual Absoluto Medio (MAPE). (Yaffee, 2000). El error porcentual absoluto medio MAPE lo calcularemos mediante la siguiente fórmula:

$$\text{MAPE}=\frac{\sum_{i=1}^{n}\frac{100|Real_i-Pronóstico_i|}{Real_i}}{n}$$

El método que produzca el menor MAPE será el considerado más exacto, para de esta manera resolver el problema planteado.

El producto de esta investigación serán varios modelos de pronósticos avanzados capaces de predecir demandas de placas digitales en el mercado gráfico quiteño.

1.4.2 JUSTIFICACIÓN PRÁCTICA.-

Esta investigación ayudará a resolver un problema real en las empresas distribuidoras de insumos gráficos, la predicción de su demanda, para que de esta manera puedan mantener un inventario permanente de dichos insumos y beneficiar a todo el sector gráfico quiteño ya que ellos podrán mantener una producción

continua. Más no lo que ocurre en la actualidad, la producción se ve interrumpida por la falta de algún insumo.

Por otro lado contribuirá en la formación de magísteres en Gerencial Empresarial, ya que ellos contarán con una referencia actualizada de modelos de pronósticos avanzados, con suficientes ejemplos académicos y reales, para poder aplicar en cada una de sus empresas.

Esta investigación es perfectamente viable, ya que se tiene acceso total a las empresas que utilizan este tipo de insumo y además se ha recolectado de ellos la información necesaria para el presente trabajo.

Además se cuenta con la disponibilidad de tiempo, los recursos financieros y materiales necesarios para culminar la investigación con éxito.

Finalmente cabe recalcar que no habrá ninguna consecuencia ni ética, moral o legal al terminar esta investigación.

2 MARCO TEÓRICO

Después de una revisión selectiva de la literatura, se ha podido encontrar la existencia de una teoría ampliamente desarrollada para dar respuesta a nuestras preguntas de investigación, la teoría de **Pronósticos basados en series de tiempo** dentro de la cual se analizarán tres metodologías: ***Métodos de Pronósticos con Suavizamiento Exponencial*** (Hyndman R., 2014), (Makridakis S., 1998), (Chatfield, 2012 Sixth Edition), (Schroeder R., 2011), (Winston, 2005), entre otros, ***Metodología de Box-Jenkins o técnicamente conocida como metodología ARIMA*** (Box G., 2008 4th Edition), (Hyndman R., 2014), (Makridakis S., 1998), (Chatfield, 2012 Sixth Edition), (Harvey, 1991), (Harvey, Forecasting, Structural Time Series Models and the Kalman Filter, 1990), (Pankratz, 1991), (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983), (Brockwell, 2009), (Mills, 1992), (Hamilton, 1994), (Yaffee, 200), (Capa, 2008). (Capa, Modelación de Series Temporales, 2007), (Gujarati Damodar N, 2010 Quinta Edición), (Wooldridge, 2010), (Cryer J., 2009), (Peña D., 2001), (Shumway R., 2011), (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983), entre otros, y ***La Metodología de Redes Neuronales Artificiales***, (Makridakis S., 1998), (Hyndman R., 2014), (Hertz, Krogh, & Palmer, 1991), (Lisboa, Edisbury, & Vellido, 2000), (Fausett, 1994), (Hecht Nielsen, 1991), (McNelis, 2005), (Haykin, 1999), (Rogers & Vemuri, 1994), (Del Brío, 2007), (Principe, 2000), (Ramón y Cajal, 1990), (Ramón y Cajal, Histology of the Nervous System Vol. I,), (Du K., 2014) (Shukla, 2010) ,(Zhang, 2004) , (Hagan M., 2015), entre otros.

2.1 ANTECEDENTES

Según (Schroeder R., 2011) La preparación de pronósticos es el arte y la ciencia de predecir eventos futuros. Hasta la última década era, en gran medida, un arte, pero hoy en día se ha convertido en una ciencia. Aunque en esta área siempre se requerirá del criterio del administrador, en la actualidad los administradores se

apoyan en herramientas y métodos matemáticos muy sofisticados. La predicción ha avanzado mucho respecto del arte oscuro de adivinar la suerte por medio de las estrellas, las hojas de té o las bolas de cristal.

Cabe indicar que un método de pronóstico debe seleccionarse cuidadosamente para el uso particular que se le pretende dar. No existe un método universal de pronóstico para todas las situaciones y existen métodos que funcionan muy bien en ciertas situaciones y no tan bien en otras, el administrador deberá estar preparado para modificarlo de acuerdo a las condiciones externas. (Schroeder R., 2011)

Los modelos de pronósticos basados en series de tiempo toman un gran auge, a partir de la crítica de Lucas, quién criticó a los modelos de regresión de ecuaciones simultáneas debido a la crisis de los precios del petróleo de 1973 y de 1979. El argumento de esta crítica fue que los parámetros estimados de un modelo de regresión de ecuaciones simultáneas dependen de la política prevaleciente en el momento en que se estima el modelo y cambian conforme lo hace la política. Esta crítica hizo posible que Robert E. Lucas obtuviera el Premio Nobel de Economía. (Gujarati Damodar N, 2010 Quinta Edición)

Además de la crítica de Lucas, George E.P. Box y Gwilym M. Jenkins publican el libro “Time series Analysis: Forecasting and Control”, en el año de 1976, el mismo que marcó el inicio de una nueva generación de herramientas de pronósticos basados en series de tiempo. (Gujarati Damodar N, 2010 Quinta Edición)

Las tres metodologías de pronósticos que se estudiarán en el presente trabajo son: Métodos de Pronósticos con Suavizamiento Exponencial (Holt - Winters), Metodología de Box-Jenkins o técnicamente conocida como metodología ARIMA y la Metodología de Redes Neuronales (Perceptrón Multicapa MLP y otras) serán sometidas a comprobación empírica con datos reales y confiables del consumo de placas metálicas del sector gráfico Quiteño, desde el año 2009 hasta el año 2015.

2.2 PRONÓSTICOS CON SUAVIZAMIENTO EXPONENCIAL.-

El suavizamiento exponencial fue propuesto a finales de 1950s (Brown 1959, Holt 1957 y Winters 1960 con sus trabajos pioneros) y han motivado a muchos métodos de pronósticos más modernos.

Un pronóstico producido utilizando suavizamiento exponencial son promedios ponderados de observaciones pasadas, donde el peso decae exponencialmente cuando la observación es más antigua. En otras palabras si una observación es más reciente es más alto el peso asociado. Estos métodos generan pronósticos rápidos y confiables para un amplio rango de series de tiempo que son aplicadas a la industria. (Hyndman R., 2014).

Dentro de esta metodología tenemos básicamente tres métodos:

- Suavizamiento Exponencial Simple,
- Método de Holt con tendencia lineal (Suavizamiento Exponencial con tendencia) y
- Método Estacional de Holt – Winters (Suavizamiento Exponencial con tendencia y estacionalidad).

En el presente trabajo se someterá a prueba al Método Estacional de Holt – Winters, ya que el tipo de demanda que se trata forma una serie de tiempo con tendencia y estacionalidad.

2.2.1 MÉTODO ESTACIONAL DE HOLT – WINTERS

El método de Holt – Winters se utiliza para pronosticar series de tiempo en las cuales están presentes tendencia y estacionalidad. (Winston, 2005), ya que los métodos de Suavizamiento Exponencial Simple y de Holt, son no apropiados por sí mismos cuando existe tendencia o estacionalidad. (Makridakis S., 1998).

El método de Holt fue extendido por Winters (1960) para capturar la estacionalidad directamente. El método de Holt – Winters se basa en tres ecuaciones de suavizamiento, una para el nivel L_t, otra para la tendencia T_t y otra para la estacionalidad S_t.

Existen dos métodos diferentes de Holt – Winters dependiendo si la estacionalidad es modelada de forma aditiva o multiplicativa. (Makridakis S., 1998)

2.2.1.1 Método de Holt-Winters con Estacionalidad Multiplicativa

Las ecuaciones básicas del Método de Holt-Winters con estacionalidad Multiplicativa son:

Nivel: $L_t = \alpha \frac{Y_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + b_{t-1})$

Tendencia: $b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}$

Estacionalidad: $S_t = \gamma \frac{Y_t}{L_t} + (1-\gamma)S_{t-s}$

Pronóstico: $F_{t+m} = (L_t + b_t * m)S_{t-s+m}$

Donde: s es la longitud de la estacionalidad así: si es mensual s=12, trimestral s=3…, L_t representa el nivel de la serie, b_t denota la tendencia, S_t es el componente estacionario y F_{t+m} es el pronóstico m períodos adelante. (Makridakis S., 1998).

2.2.1.2 Método de Holt-Winters con Estacionalidad Aditiva

Las ecuaciones básicas del Método de Holt-Winters con estacionalidad Aditiva, aunque es menos común, son:

Nivel: $L_t = \alpha(Y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1})$

Tendencia: $b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}$

Estacionalidad: $S_t = \gamma(Y_t - L_t) + (1-\gamma)S_{t-s}$

Pronóstico: $F_{t+m} = (L_t + b_t * m) + S_{t-s+m}$

La única diferencia con las ecuaciones anteriores es que los índices de estacionalidad son sumados y restados en lugar de ser multiplicados y divididos. (Makridakis S., 1998)

Para aplicar este algoritmo es importante tomar en cuenta los siguientes pasos:

1.- Examinar el gráfico de los datos para ver si el efecto de estacionalidad multiplicativa o aditiva es más apropiado,

2.- Proveer los valores iniciales para el algoritmo, aquí se analizarán varias opciones.

3.- Determinar los valores de $\alpha, \beta \; y \; \gamma$, se analizarán también algunas opciones.

6.- Obtener un intervalo de Predicción. (Chatfield, 2012 Sixth Edition), (Hyndman R., 2014).

Con estas guías se analizarán algunos ejemplos tipo (académicos) para entender este método, finalmente se procederá a realizar el pronóstico de las placas metálicas digitales en el mercado quiteño desde el 2009 hasta el 2015.

2.3 METODOLOGÍA DE BOX-JENKINS

La estrategia de este modelo tiene cuatro partes: (1) Identificación del Modelo, (2) Estimación del modelo, (3) Diagnóstico del Modelo y (4) Pronóstico

2.3.1 IDENTIFICACIÓN

En esta etapa se intentará obtener una idea de la naturaleza tanto de la relación entre la entrada y la salida como un patrón estructurado de tiempo en la señal de error, permitiendo "*que los datos hablen por sí mismos".* Esto podría ocasionar la estimación de una gran cantidad de parámetros, entonces trataremos de escoger entre una familia de modelos uno tentativo que resuma estos parámetros con

parsimonia (forma compacta). Es decir se intentará identificar un modelo (parámetros p,d,q) que sea *consistente con los datos y que tenga el menor número de coeficientes*, necesarios para explicar adecuadamente el comportamiento de los datos. Se verá como la ACF (función de auto-correlación) y la PACF (función de auto-correlación parcial) nos ayudarán en el escogitamiento de los modelos en esta etapa. (Gujarati Damodar N, 2010 Quinta Edición), (Pankratz, Forecasting with Dynamic Regression Models, 1991)

2.3.2 ESTIMACIÓN

En esta etapa se obtendrá un estimado de los parámetros del modelo tentativo. Tras identificar los valores de (p,d,q), la siguiente etapa es estimar los parámetros de los términos autoregresivos y medias móviles incluidos en el modelo. Para este cálculo se utilizarán métodos como son: mínimos cuadrados, máximo de verosimilitud o recurrir a métodos de estimación no lineal (en parámetros). Normalmente esta labor la desarrollará el software estadístico a utilizar (Minitab ver. 17). Pero como se considera que los fundamentos matemáticos son muy importantes, incluiremos estos métodos en los apéndices respectivos.
(Gujarati Damodar N, 2010 Quinta Edición), (Pankratz, Forecasting with Dynamic Regression Models, 1991), (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

2.3.3 DIAGNÓSTICO

Después de seleccionar un modelo ARIMA particular y de estimar sus parámetros, se tratará de ver si el modelo seleccionado se ajusta a los datos en forma razonablemente buena, pues es posible que exista otro modelo ARIMA que también lo haga. Es por esto que el diseño de modelos ARIMA de Box-Jenkins es un arte más que una ciencia; se requiere de gran habilidad para seleccionar el modelo ARIMA correcto. Una simple prueba del modelo seleccionado es ver si los residuales estimados a partir de este modelo son ruido blanco; si lo son, aceptamos

el ajuste particular, si no lo son se debe reformular el modelo. Por lo tanto la metodología de Box – Jenkins es un proceso iterativo. Ver figura 2.1.
Además se debe detectar y tratar los residuales por Outliers (valores inusuales). Un estudio de los outliers nos conducen a un mejor entendimiento de los datos, una mejor identificación, mejor estimado de los coeficientes del modelo y un mejor pronóstico. (Pankratz, Forecasting with Dynamic Regression Models, 1991)

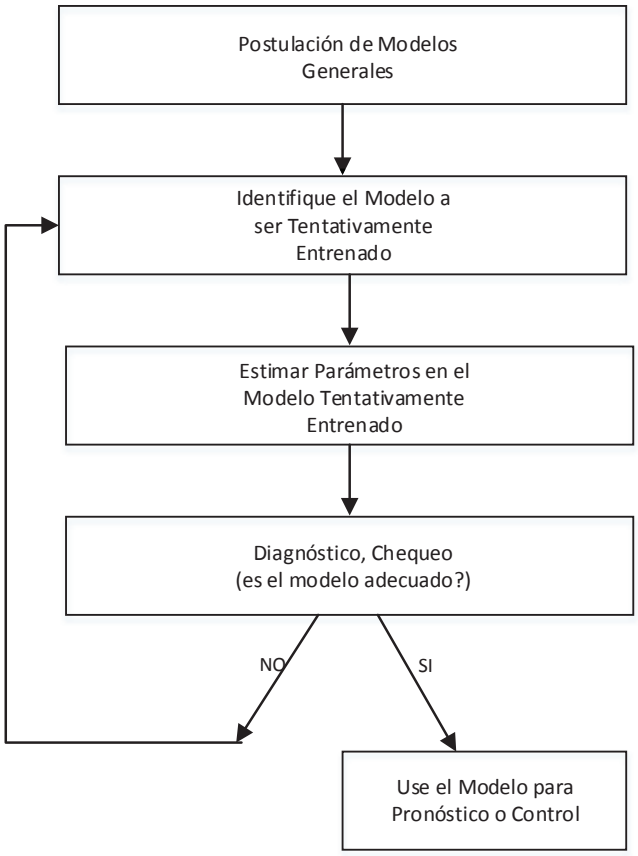


Figura 2.1 Etapas en un enfoque iterativo en la construcción de modelos

Adaptado de (Box G., 2008 4th Edition Pág.18)

2.3.4 PRONÓSTICO

Una razón de la popularidad del proceso de construcción de modelos ARIMA es su éxito en el pronóstico. En muchos casos, los pronósticos obtenidos por este método son más confiables que los obtenidos por modelos tradicionales, en particular en el caso de pronósticos a corto plazo. (Gujarati Damodar N, 2010 Quinta Edición)

2.3.5 PARSIMONIA.-

Box y Jenkins enfatizan el importante principio de parsimonia. Acorde con este principio, se necesita encontrar un modelo adecuado pero el más simple –uno que contenga pocos coeficientes y explique adecuadamente el comportamiento de los datos observados. Modelos con este principio hacen buen uso de una muestra limitada y tienden a dar pronósticos más precisos. (Pankratz, Forecasting with Dynamic Regression Models, 1991), (Box G., 2008 4th Edition), (Gujarati Damodar N, 2010 Quinta Edición).

Para predecir la demanda objeto de esta investigación, es decir una serie de tiempo no estacionaria con estacionalidades y tendencia, es importante estudiar los modelos ARIMA con estacionalidades como así se hará. Luego se analizarán algunos ejemplos tipo para la familiarización con esta metodología y finalmente proceder a realizar el pronóstico de las placas metálicas digitales en el mercado quiteño desde el 2009 hasta el 2015, con esta metodología.

2.3.6 ECUACIONES DEL MODELO ARIMA (P,D,Q)

El modelo ARIMA (p,d,q), puede ser escrito de una manera compacta utilizando las siguientes definiciones:

$\nabla^d = (1-B)^d$ *Operador de diferencias de orden d*

$\emptyset(B) = (1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p$ *(Operador AR (auto-regresivo) de orden p)*

$\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$ $(Operador\ MA\ (medias\ móviles) de\ orden\ q)$

Utilizando estas definiciones el modelo general ARIMA (p,d,q) es:

$$\emptyset(B)\nabla^d z_t = C + \theta(B)a_t$$

En base a este modelo ARIMA básico se estudiará el modelo ARIMA estacional, que nos permitirá predecir el tipo de demanda del presente trabajo.

2.4 METODOLOGÍA DE REDES NEURONALES

El estudio de las redes neuronales artificiales o ANS (*Artificial Neural Systems*), puede orientarse en dos direcciones, bien como modelos del sistema nervioso y los fenómenos cognitivos, bien como herramientas para la solución de problemas prácticos; este último será el punto de vista que se tratará en el presente trabajo.
Se considerará que las redes neuronales artificiales son sistemas, hardware y software, de procesamiento, que copian esquemáticamente la estructura neuronal del cerebro para tratar de reproducir sus capacidades. Los ANS son capaces así de aprender de la experiencia a partir de las señales o datos provenientes del exterior, dentro de un marco de computación paralela y distribuida, fácilmente implementable en dispositivos de hardware específicos. (Del Brío, 2007)

2.4.1 BREVE INTRODUCCIÓN BIOLÓGICA

Con el fin de establecer el paralelismo entre los sistemas neuronales biológicos y los ANS es conveniente exponer algunos conceptos básicos de los primeros.

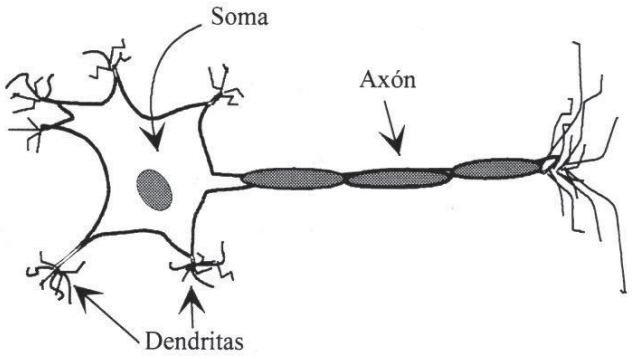


Figura 2.2 Esquema de una neurona biológica típica

((Del Brío, 2007) Pág. 4)

Se estima que el sistema nervioso contiene alrededor de 100.000 millones (10^{11}) de neuronas, vistas al microscopio muchas de ellas tienen un aspecto como el de la Figura 2.2.

Desde el punto de vista funcional, las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, poseen un canal de entrada de información, las dendritas; un órgano de cómputo, el soma y una canal de salida, el axón. Se calcula que una neurona del córtex cerebral recibe información, por término medio, de unas 10.000 neuronas (convergencia) y envía impulsos a varios cientos de ellas (divergencia).

Un punto importante es la existencia de una organización horizontal en capas (hasta seis capas) en el córtex cerebral, coexistiendo una organización vertical en forma de columnas de neuronas.

La unión entre dos neuronas se llama sinapsis. En el tipo más común de sinapsis no existe un contacto físico entre neuronas existe un vacío de unas 0.2 micras. En relación a la sinapsis se habla de neuronas pre-sinápticas (las que envían las señales) y post-sinápticas (las que reciben). Las sinapsis son unidireccionales ya que la información fluye en un único sentido.

Las señales nerviosas pueden trasmitirse eléctrica o químicamente. La transmisión química prevalece fuera de ella, mientras que la eléctrica se da en su interior. La trasmisión química se basa en el intercambio de neurotransmisores (glutamato, adrenalina), mientras que la eléctrica hace uso de descargas que se producen en el cuerpo celular y se trasmiten a través del axón. (Del Brío, 2007).

2.4.2 APRENDIZAJE

La intensidad de una sinapsis no viene representada por una cantidad fija, sino que puede ser modulada en una escala temporal mucho más amplia que la del disparo de las neuronas (horas, días o meses). Esta plasticidad sináptica se supone constituye, al menos en buena medida, el aprendizaje. Este tipo de acciones (en especial la modificación de las intensidades sinápticas) serán las que utilicen los sistemas neuronales artificiales para llevar a cabo el aprendizaje. (Del Brío, 2007)

La bibliografía en este campo es abundante, si el lector desea profundizar recomendamos las siguientes referencias: (Haykin, 1999), (Del Brío, 2007), (Hecht Nielsen, 1991) y (Hertz, Krogh, & Palmer, 1991).

2.4.3 CEREBRO Y COMPUTADOR

Los ANS imitan la estructura de Hardware del sistema nervioso, con la intención de construir sistemas de información paralelos, distribuidos y adaptivos que puedan presentar un cierto comportamiento inteligente.

La idea que subyace en los sistemas neuronales artificiales es que para abordar el tipo de problemas que el cerebro resuelve con eficiencia, puede resultar conveniente construir sistemas que copien en cierto modo la estructura de las redes neuronales biológicas con el fin de alcanzar una funcionalidad similar.

Conceptos clave de los sistemas nerviosos que pretenden emular los artificiales son:

- Paralelismo de cálculo
- Memoria distribuida
- Adaptabilidad al entorno (Del Brío, 2007)

2.4.4 ESTRUCTURA DE UN SISTEMA NEURONAL ARTIFICIAL

Formalmente, y desde el punto de vista del grupo PDP (*Parallel Distributed Processing Research Group*) de la Universidad de California en San Diego, un **sistema neuronal o conexionista**, está compuesto por los siguientes elementos:

- Un conjunto de procesadores elementales o neuronas artificiales.
- Un patrón de conectividad o arquitectura.
- Una dinámica de activaciones.
- Una regla o dinámica de aprendizaje
- El entorno donde opera.

2.4.4.1 Modelo de Neurona Artificial

Aunque el comportamiento de algunos sistemas neuronales biológicos sea lineal, en general la respuesta de la neurona biológica es de tipo NO lineal, característica que es emulada en los ANS ya desde la neurona original.

La formulación de la neurona artificial como dispositivo no lineal constituye una de sus características más destacables y una de las que proporciona un mayor interés a los ANS, pues el tratamiento de problemas altamente NO lineales no suele ser fácil de abordar mediante técnicas convencionales. (Del Brío, 2007)

2.4.4.2 Modelo General de Neurona Artificial

La estructura genérica de neurona artificial en el marco establecido por el grupo PDP (Rumelhart, McClelland, & Group, 1987), se muestra en la Figura 2.3.

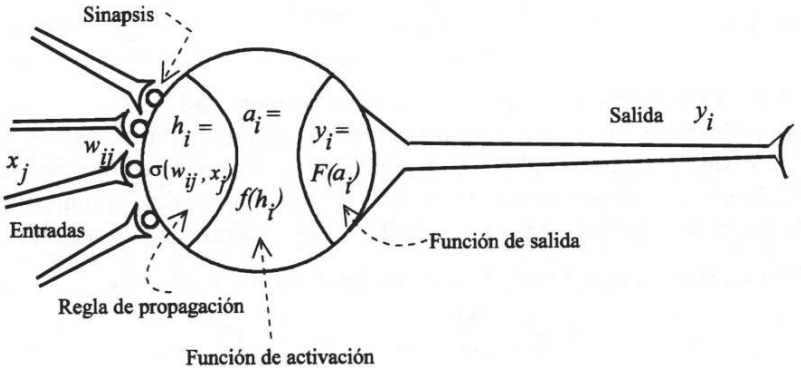


Figura 2.3 Modelo Genérico de Neurona Artificial

(Del Brío, 2007 Pág. 15)

Los elementos que constituyen la neurona etiquetada como i son los siguientes:

- Conjunto de **Entradas**, $x_j(t)$
- **Pesos sinápticos** de la neurona i, w_{ij} que representa la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i.
- **Regla de Propagación** $\sigma(\mathrm{w_{ij}}|\mathrm{x_j})$, que proporciona el valor del potencial postsináptico $h_i(t) = \sigma(\mathrm{w_{ij}}|\mathrm{x_j})$ de la neurona i en función de sus pesos y entradas.
- **Función de Activación** $f_i(a_i(t-1)|h_i(t))$, que proporciona es estado de activación actual $a_i(t) = f_i(a_i(t-1)|h_i(t))$ de la neurona i, en función de su estado anterior $a_i(t-1)$ y de su potencial postsináptico actual.
- **Función de Salida** $\mathrm{F_i(a_i(t))}$, que proporciona la salida actual $y_i(t) = \mathrm{F_i(a_i(t))}$ de la neurona i en función de su estado de activación.

De este modo, la operación de la neurona i puede expresarse como

$$y_i(t) = \mathrm{F_i}\left(f_i\left(a_i(t-1)\middle|\sigma\left(\mathrm{w_{ij}}\middle|\mathrm{x_j(t)}\right)\right)\right)$$

Este modelo formal se inspira en la operación biológica, en el sentido de integrar una serie de entradas y proporcionar cierta respuesta que se propaga por el axón. (Del Brío, 2007).

En el desarrollo de la tesis se profundizará cada uno de estos elementos.

2.4.5 PRONÓSTICOS CON REDES NEURONALES

Como se ha visto las redes neuronales están basadas en modelos matemáticos que emulan la forma de pensar del cerebro para trabajar. Cuando se aplican a series de tiempo, nos proveen un método de pronóstico no Lineal. Los pronósticos con redes neuronales requieren más cantidad de observaciones que los otros métodos discutidos en esta investigación, pero en general son más flexibles.

Los métodos de redes neuronales han adoptado terminología muy diferente a los otros métodos, que podría conducir a alguna confusión. Así en lugar de “Modelo”, tenemos “Red”; en lugar de “parámetros” tenemos “pesos”; en lugar de hablar de “estimar los parámetros” se habla de “entrenamiento de la red”.

Una red neuronal puede ser pensada como una red de neuronas o unidades organizadas en capas. La capa superior consiste en un conjunto de unidades de entrada y la capa inferior consiste en un set de unidades de salida. Las unidades en cada capa son entrelazadas con las unidades de las capas superiores.

En una red neuronal se deben especificar los siguientes elementos:

- **Arquitectura**.- se debe especificar el número de capas o unidades en la red y la forma cómo están conectadas.
- **Función de Activación**.- describe como cada unidad combina las entradas para obtener una salida.
- **Función Costo**.- es una medida de la precisión del pronóstico como MSE.
- **Algoritmo de entrenamiento**.- el mismo que encuentra los parámetros que minimizan la función costo.

El poder de las redes neuronales viene con la inclusión de capas adicionales intermedias consistente en unidades no lineales escondidas entre las entradas y

las salidas. Un ejemplo se muestra en la Figura 2.4 que muestra una red neuronal que contiene una sola capa oculta o escondida.

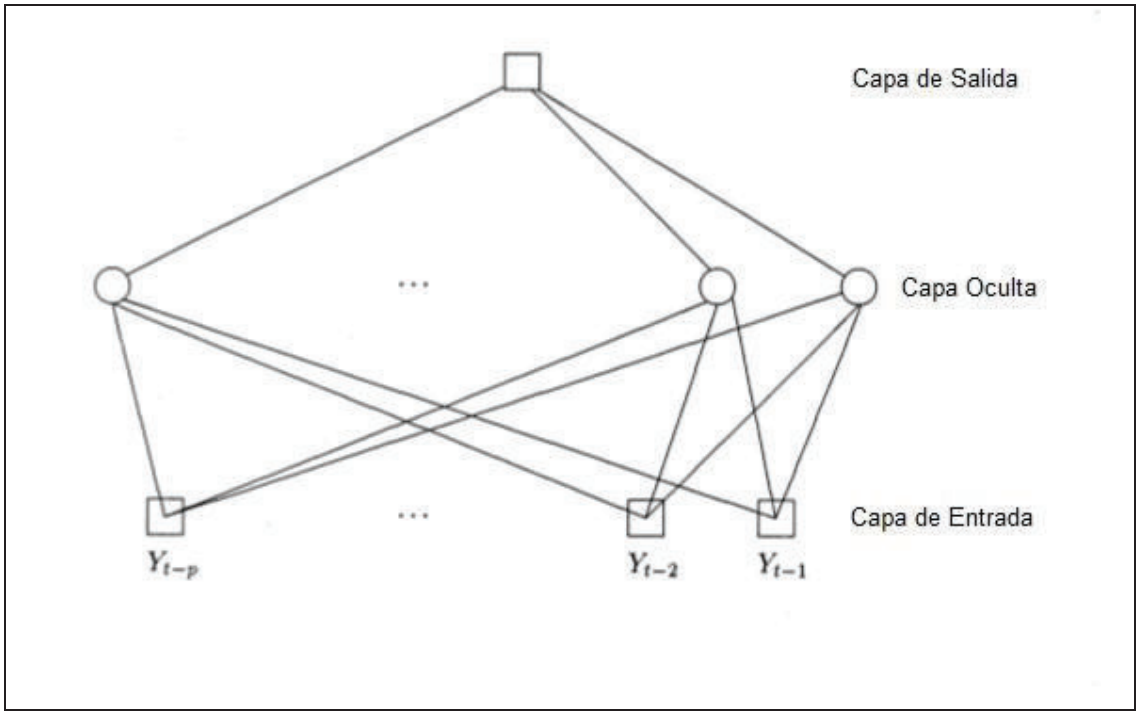


Figura 2.4 Esquema de una red neuronal con una capa oculta

(Modificado de Makridakis S., 1998 Pág. 438)

Aquí las entradas son conectadas a una capa escondida con unidades no lineales las cuales están conectadas a una unidad de salida lineal. Es posible también definir una red con entradas y salidas no lineales.

La respuesta de una unidad se llama su “valor de activación”. Una elección común para una función de activación no lineal es una combinación de una función lineal de entrada seguida por una función de “atenuación” conocida como Sigmoidea o logística. Por ejemplo, las entradas a la unidad escondida en la Figura 2.4 pueden ser linealmente combinadas para dar:

$$Z = b + \sum_{i=1}^{n} w_i \, Y_{t-i}$$

La cual sucesivamente es una entrada a la función no lineal

$$S(Z) = \frac{1}{1 + e^{-aZ}}$$

Uno de los beneficios de la función Sigmoidea es que reduce el efecto de los valores de entrada extremos, proveyendo así cierto grado de robustez a la red. En este ejemplo los parámetros a, b y $w_{1,}\ldots w_p$.

Los valores resultantes de $S(Z)$ para cada una de las unidades escondidas en la Figura 2.4 son entonces combinadas usando una función lineal para dar la salida (o Pronóstico).

Esta red neuronal es equivalente a un modelo de auto-regresión no lineal

$Y_t = f\left(Y_{t-1,\ldots}\, Y_{t-p}\right) + e_t$

La capa de entrada de la red consiste en un conjunto de variables explicativas como sea posible y además valores retardados de la serie de tiempo. Para datos estacionales, la práctica general es tener tantas entradas retardadas como hay períodos en la estación. El número de unidades en la capa de salida corresponde al número de variables a ser predichas.

Una desventaja de los métodos de redes neuronales es que no nos permiten entender bien los datos ya que no hay un modelo explícito. Ellas proveen un enfoque de "caja negra" para predecir. Por otro lado, ellas podrían trabajar en situaciones donde el enfoque basado en un modelo explícito falla. Lo prometedor de esta metodología es que se puede adaptar a irregularidades o comportamientos inusuales (Outliers) en la serie de tiempo de interés. (Makridakis S., 1998)

2.4.6 CLASIFICACIÓN DE LOS MODELOS NEURONALES

Dependiendo del modelo de neuronas concreto, de la arquitectura o topología de conexión y del algoritmo de aprendizaje, surgirán distintos modelos de redes neuronales:

En la Figura 2.5 se clasifican a los ANS por el tipo de aprendizaje y arquitectura.

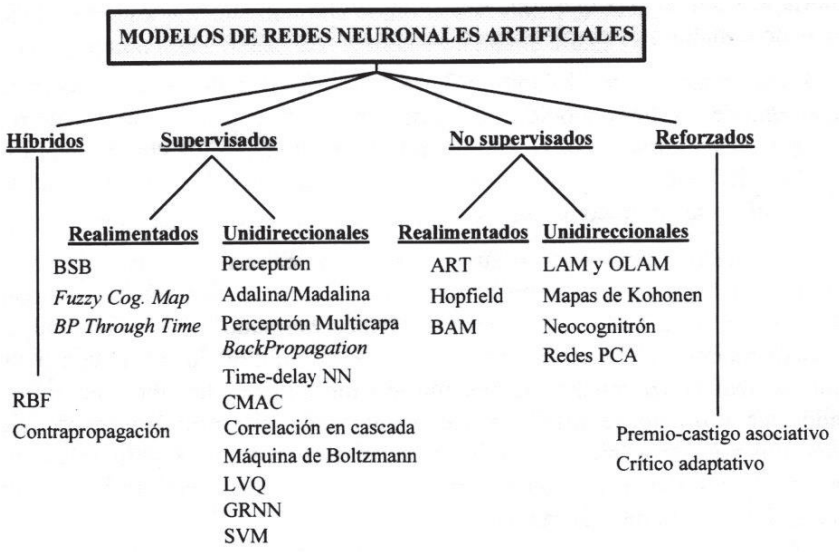


Figura 2.5 Clasificación de los ANS por el tipo de Aprendizaje y Arquitectura

(Del Brío, 2007 Pág. 31

Así de acuerdo al tipo de aprendizaje se puede clasificar a les redes neuronales en: Redes Supervisadas, No supervisadas, Hibridas y Reforzadas.

De acuerdo a la topología de la red se clasifican en Unidireccionales y Realimentadas.

Varios autores recomiendan el Perceptrón Multicapa (MLP) con aprendizaje BP (Back Propagation) para la aplicación de pronósticos. (Del Brío, 2007), (Haykin, 1999), (Makridakis S., 1998), (Hertz, Krogh, & Palmer, 1991), (McNelis, 2005), (Rogers & Vemuri, 1994).

Se seguirán esas recomendaciones y en el presente trabajo utilizaremos el MLP como *metodología básica* de pronóstico.

2.4.6.1 El Perceptrón Multicapa (MLP)

Una red como la de la figura 2.6 con al menos una capa oculta se denomina *Perceptrón multicapa (MLP)*.

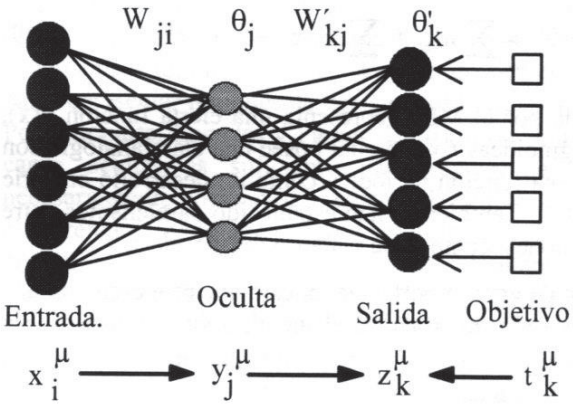


Figura 2.6 Arquitectura del Perceptrón Multicapa MLP

(Del Brío, 2007 Pág. 66

EL grupo PDP en 1986 lo popularizó, Esta arquitectura suele entrenarse mediante el algoritmo denominado de retro-propagación de errores o BP (Back-propagation), o alguna de sus variantes o derivados, razón por la cual al conjunto *arquitectura MLP + aprendizaje BP* suele denominarse **red de retropropagación** o simplemente **BP**. (Del Brío, 2007)

En la Figura 2.6 se denominan x_i a las entradas de la red, y_i a las salidas de la capa oculta y z_k a las de la capa final (y globales de la red), t_k serán las salidas objetivo (target). Además, w_{ji} son los pesos de la capa oculta y θ_j sus umbrales o bías, $w_{kj}^{'}$ los pesos de la capa de salida y $\theta_k^{'}$ sus umbrales o bías. La operación de un MLP con una capa oculta y neuronas de salida lineal se expresa matemáticamente de la siguiente manera:

$$z_k = \sum_j w_{kj}^{'} - \theta_i^{'} = \sum_j w_{kj}^{'} f(\sum w_{ji}\, x_i - \theta_j) - \theta_i^{'}$$

Siendo f(.) del tipo sigmoideo, como por ejemplo las siguientes:

$$\boldsymbol{f(x)} = \frac{\mathbf{1}}{\mathbf{1}+e^{-}} \qquad \text{o} \qquad \boldsymbol{f(x)} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \mathbf{tan}\, \boldsymbol{h(x)}$$

la primera proporciona una salida en el intervalo $[0, +1]$ y la segunda en $[-1, +1]$. Esta es la arquitectura más común de MLP, aunque existen numerosas variantes, como incluir neuronas no lineales en la capa de salida, introducir más capas ocultas, emplear otras funciones de activación, limitar el número de conexiones

entre una neurona y la de la capa siguiente, introducir dependencias temporales o ***arquitecturas recurrentes***, etc.

2.4.6.1.1 Aprendizaje por retro-propagación de errores (backpropagation)

La solución al problema de entrenar los nodos de las capas ocultas de la arquitectura multicapa la proporciona el algoritmo de retro-propagación de errores o BP.

El método consiste en proponer una función costo o error que mida el rendimiento de la red, función que dependerá de los pesos sinápticos. Dada esta función costo, se introduce un procedimiento general de optimización que sea capaz de proporcionar una configuración de pesos que correspondan a un mínimo de la función propuesta. El método de optimización aplicado a la función costo proporcionará una regla de actualización de pesos, que en función de los patrones de aprendizaje modifique iterativamente los pesos hasta alcanzar el punto óptimo de la red neuronal. El método de optimización (minimización) más habitualmente empleado es el denominado **descenso por el gradiente**. (Del Brío, 2007)

En estudios anteriores como los de (Del Brío, 2007), (Haykin, 1999) y (Hertz, Krogh, & Palmer, 1991), (Hagan M., 2015), nos previenen de un inconveniente del BP, su lentitud de convergencia. Ellos sugieren la utilización de un segundo grupo de algoritmos de aprendizaje denominados genéricamente **métodos de segundo orden**, se basan en realizar el descenso del gradiente utilizando también la información proporcionada por el ritmo de cambio de la pendiente. (Del Brío, 2007).

Por esta razón en el presente trabajo utilizaremos uno de ellos que es el algoritmo de **Levenberg-Marquardt.** Este algoritmo es mucho más robusto que el BP, puede acelerar en uno o dos órdenes de magnitud la convergencia, aunque en contrapartida es mucho más complejo de implementar y precisa más recursos de cálculo.

Cabe destacar que ninguno de los algoritmos descritos puede considerarse superior, en general: un buen método en un caso puede proporcionar un rendimiento pobre en otro. (Del Brío, 2007).

En el manual de *Neural Networks Toolbox* del programa **Matlab**, se proporcionan algunas pistas que nos han hecho inclinar por el algoritmo de Levenberg – Marquardt.

2.4.6.1.2.- Generalización de la Red

Uno de los aspectos más importantes de los ANS es la capacidad de generalizar a partir de ejemplos, lo que constituye el **problema de la memorización frente a generalización**. Por **generalización** se entiende la capacidad de la red de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento. Una red neuronal correctamente entrenada generalizará, lo que significa que ha aprendido adecuadamente la correspondencia o *mapping* no sólo de los ejemplos concretos presentados, sino también responderá satisfactoriamente ante patrones nunca antes vistos.

2.4.6.2 Redes Neuronales Dinámicas

Como se ha analizado antes, nuestro cerebro tiene una estructura fuertemente recurrente, esta recurrencia se demuestra en la dinámica de auto-alimentación en los procesos de percepción, actuación y aprendizaje además en el mantenimiento del organismo vivo. En las redes neuronales recurrentes dinámicas existe al menos un lazo de realimentación. (Du K., 2014).

El MLP puramente estático es incapaz de procesar información relacionada con el tiempo. En aplicaciones de sistemas dinámicos, se necesita el pronóstico al tiempo (t+1) a partir de un estado de la red neuronal al tiempo (t). (Du K., 2014).

En el presente caso se necesita predecir al menos hasta cuatro pasos adelante (t+4) ya que el proceso de importación toma de tres a cuatro meses.

Para generar una red neuronal dinámica, la memoria debe ser introducida. El más simple elemento de memoria es una unidad de retardo en el tiempo, cuya función de transferencia es $H(z) = Z^{-1}$. La más simple arquitectura de memoria es una línea de retardo dedicada que consiste en una serie de retardos en la unidad de tiempo. Un MLP puede convertirse en una red neuronal dinámica introduciendo

lazos de retardo en el tiempo a la entrada, salida o en la capa escondida. Una arquitectura de red que incorpora retardos en el tiempo es conocida como: red neuronal con retardo en el tiempo (TDNN) del inglés (time-delay neural network). (Du K., 2014).

Dentro de la TDNN se puede destacar dos: Nonlinear autoregressive with external input (NARX) y Nonlinear autoregressive neural network (Narnet), que son las que se utilizarán en el presente trabajo para la predicción de las placas digitales en el mercado gráfico quiteño, recomendadas por Hagan, Demuth y Beale (autores del módulo de redes neuronales del programa MATLAB), (Beale M., 2015), para pronósticos en su libro “Neural Network Design”. (Hagan M., 2015).

2.5 ESTUDIOS ANTERIORES

En el trabajo de investigación de (Arteaga E., 2010) realizado para la facultad de ciencias de la EPN, se realiza el pronóstico de ventas en base a series de tiempo de cuatro marcas de bebidas gaseosas ecuatorianas: Cocal Cola, Fanta, Sprite y Fioravanti. Después de realizar la prueba de Dickey Fuller aumentada el autor comprueba que las series de tiempo de las cuatro marcas de bebidas gaseosas son estacionarias.

Demuestra que la metodología ARIMA obtiene mejores resultados que la metodología de suavizamiento exponencial que utilizaban en ese entonces y la metodología VAR de vectores autoregresivos (maneja series de tiempo multivariadas) muy poderosa para la predicción, siempre y cuando se escojan las variables apropiadas.

En las series de tiempo de este trabajo se tiene presencia de estacionalidad y se comprueba que la metodología ARIMA la procesa muy bien.

En la investigación de (Zavaleta E., 2010) compara varias metodologías de pronósticos basadas en series de tiempo para predecir la demanda de productos farmaceúticos. Así menciona que las metodologías de suavizamiento exponencial y ARIMA producen errores de pronóstico en el orden del 15% mientras la metodología de Redes Neuronales Artificiales produce errores menores al 4%.

En ese trabajo demuestra que una arquitectura perceptrón multicapa con backpropagation da excelentes pronósticos.
Además utiliza el programa Matlab para los cálculos y simulación de la red neuronal.

En la tesis de (Valverde A., 2012) aplica dos métodos de series de tiempo ARIMA y VARX para la predicción de la inflación en el Ecuador desde el año 2000 al 2010. Aplica la prueba de Dickey Fuller Aumentada y comprueba que la serie es estacionaria. En base a los correlogramas (gráficos de la función de autocorrelación y autocorrealción parcial) identifica los valores de p y q para el modelo ARIMA, ya que la serie fue estacionaria. Además observando los rezagos 12 y 24 identifica una estacionalidad presente en los datos, razón por la cual aplica un modelo estacional SARIMA. Se prueba para una estacionalidad anual pero los correlogramas de los residuos muestran una pequeña correlación, razón por la cual se sube el grado del término auto regresivo estacional a dos ($AR(2)_{12}$), se observa el correlograma de los residuos y ya se puede ver que tiene un comportamiento de ruido blanco, el modelo final fue: SARIMA(1,0,2)(2,0,0)$_{12}$.
La predicción con este modelo resulta ser satisfactoria. Luego se realiza el mismo ejercicio con un modelo multivariado (VECX) y finalmente se concluye que el modelo multivariado VECX es el que menor error cuadrático medio produce.

2.6 MARCO CONCEPTUAL

Los términos que utilizaremos en el desarrollo de la presente tesis son los siguientes:
Variable Aleatoria.- variable cuyo valor está determinado por el resultado de un experimento al azar. (Wooldridge, 2010).
Desviación Estándar.-es el promedio de desviación de las puntuaciones con respecto a la media. Cuanto mayor sea la dispersión de los datos alrededor de la media, mayor será la desviación estándar. (Levin, 2010).
Varianza.- es la desviación estándar elevada al cuadrado. (Levin, 2010).
Covarianza.- valor que indica el grado de variación conjunta entre dos variables aleatorias. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Ruido Blanco.- Proceso aleatorio independiente, serialmente no correlacionado e idénticamente distribuido como una distribución normal con media cero y varianza constante. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Series de Tiempo o Proceso Estocástico.- secuencia de variables aleatorias indexadas en el tiempo. (Gujarati Damodar N, 2010 Quinta Edición).

Proceso Estocástico NO Estacionario.- proceso en el que su función de distribución de probabilidad varía con el tiempo. Es decir su media, varianza y covarianza varían en el tiempo. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Autocorrelación.- nos indica como los valores actuales de una serie de tiempo están relacionados con sus propios valores futuros o sus propios valores pasados. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Error de Pronóstico.- La exactitud general de cualquier modelo de pronóstico puede determinarse al comparar los valores pronosticados con los valores reales u observados. (Gujarati Damodar N, 2010 Quinta Edición).

Error de Pronóstico = Demanda Real – Valor pronosticado

Las tres medidas más populares del error de pronóstico son: Desviación Absoluta Media (MAD), Error Cuadrático Medio **(**MSE) y error porcentual absoluto medio (MAPE).

Desviación Absoluta Media MAD del inglés **(*mean absolute deviation*).-** La desviación absoluta media se calcula sumando los valores absolutos de los errores individuales del pronóstico y dividiendo el resultado entre el número de períodos con datos (n). (HEIZER & RENDER, 2004)

$$\text{MAD}= \frac{\sum |Real - Pronóstico|}{n}$$

Error Cuadrático Medio MSE del inglés **(*mean squared error*).-** El error cuadrático medio es el promedio de los cuadrados de las diferencias encontradas entre los valores pronosticados y los observados. (HEIZER & RENDER, 2004).

$$\text{MSE}= \frac{\sum (Errores\ de\ Pronóstico)^2}{n}$$

Error porcentual absoluto medio MAPE del inglés **(*mean absolute percent error*).-** El error porcentual absoluto medio se calcula como el promedio de las diferencias absolutas encontradas entre los valores pronosticados y los reales, expresado como un porcentaje de los valores reales. (HEIZER & RENDER, 2004).

$$MAPE=\frac{\sum_{i=1}^{n}\frac{100|Real_i-Pronóstico_i|}{Real_i}}{n}$$

Principio de Parsimonia.- un modelo cumple con el principio de parsimonia si contiene el menor número de coeficientes posible y explique adecuadamente el comportamiento de los datos observados. (Box G., 2008 4th Edition).

Neurona.-Procesador de información sencillo. (Del Brío, 2007).

Red Neuronal.- red de neuronas o unidades organizadas en capas. (Del Brío, 2007).

Arquitectura de una red neuronal.- Especifica el número de capas o unidades en la red y la forma cómo están conectadas. (Del Brío, 2007).

Pesos Sinápticos.-representa la intensidad de interacción entre neuronas. (Del Brío, 2007).

Función de Activación.- describe como cada unidad combina las entradas para obtener una salida. (Makridakis S., 1998).

Función Costo.- es una medida de la precisión del pronóstico podría ser por ejemplo el MSE. (Del Brío, 2007).

Algoritmo de entrenamiento.- el mismo que encuentra los parámetros que minimizan la función costo. (Del Brío, 2007).

Aprendizaje Supervisado.- en este tipo de aprendizaje se le presenta a la red un conjunto de patrones, junto con la salida deseada u objetivo, e iterativamente la red ajusta sus pesos hasta que su salida tienda a ser la deseada. (Hagan M., 2015).

Aprendizaje No Supervisado o auto-organizado.-se puede describir genéricamente como la estimación de la densidad de probabilidad p(**x**) que describe la distribución de patrones x pertenecientes al espacio de entrada a partir de muestras (ejemplos). (Del Brío, 2007).

Mapping.- Encontrar una función que a partir de un conjunto de entradas proporcione la salida deseada. (Hagan M., 2015).

Algoritmo de retro-propagación de errores (Backpropagation BP).- este algoritmo ajusta los pesos de la red en proporción a la diferencia existente entre la salida actual de la red y la salida deseada, con el objetivo de minimizar el error actual de la red. (Del Brío, 2007).

3 METODOLOGÍA

3.1 ALCANCE DE LA INVETIGACIÓN

El alcance del presente trabajo es descriptivo, ya que se pretende medir el error porcentual absoluto medio MAPE para determinar cuál metodología es la más exacta en la predicción de placas digitales en el mercado gráfico quiteño desde el 2009 hasta el 2015

3.2 DISEÑO DE LA INVETIGACIÓN

Se tratará de responder a las preguntas de investigación mediante un diseño de investigación no experimental ya que no existirá manipulación alguna de las variables y solo se observará la demanda de placas digitales en una ambiente natural para calcular las diferentes estadísticas de pronósticos.
Además seguiremos esta demanda y el error de los pronósticos con el paso del tiempo con datos desde el 2009 hasta el 2015 por lo que se enmarca en el tipo de diseño de investigación longitudinal de panel.
Resumiendo tendríamos un diseño de investigación: No experimental longitudinal de panel.

3.3 SELECCIÓN DE LA MUESTRA

La muestra está claramente limitada de la población de insumos gráficos (tinta, placas, papel, químicos y otros) se ha escogido para este estudio solamente las placas digitales del mercado de la ciudad de Quito desde el año 2009 hasta el 2015. Además en la ciudad de Quito se cuenta con alrededor de 80 empresas con equipos de última tecnología para la producción de placas digitales, se ha escogido una muestra no probabilística de 52 empresas ya que ellas han podido suministrar información válida y confiable para el presente trabajo.

3.4 RECOLECCIÓN DE DATOS

La recolección de datos se realizará mediante visitas personales (observación) a los clientes que utilizan las placas digitales, para garantizar la validez, confiablidad y objetividad de la información.

3.5 ANÁLISIS DE DATOS

El análisis de datos se realizará mediante los programas estadísticos como son: Minitab, EViews y MatLab, los que servirán para estimar los diferentes parámetros de los modelos de las distintas metodologías, para así calcular el error porcentual absoluto medio MAPE y finalmente presentar los resultados.

3.6 FUENTES DE INFORMACIÓN

Las fuentes de información son básicamente primarias, se ha recogido información de libros, artículos especializados y material de Internet, actualizados en el tema de pronósticos en base a series de tiempo.

4 METODOS DE SUAVIZAMIENTO EXPONENCIAL

4.1 INTRODUCCION

Como se conoce de la estadística un estimador que minimiza el error cuadrático medio (MSE) es la media. Si la media es utilizada como herramienta de pronóstico, entonces, como cualquier método de pronóstico, su uso óptimo requiere del conocimiento de las condiciones que se deben cumplir.

Así para la media, la condición es que los datos sean *estacionarios*, esto significa que los datos oscilen alrededor de un valor constante (la media) y que la varianza alrededor de la media sea constante en el tiempo. (Makridakis S., 1998)

En otras palabras, si una serie de tiempo es estacionaria sujeta a algún error aleatorio (o ruido), la media es un buen estadístico y se puede usar para predecir los valores futuros. Si la serie de tiempo contiene tendencia (sea creciente o decreciente), o efectos estacionales (Altas ventas de combustible en época de invierno) o ambos a la vez, la media ya no es un buen predictor.

En este capítulo se analizarán varios métodos de pronósticos que superan a la media como predictor de valores futuros, conocidos como métodos de suavizamiento exponencial, su nombre debido a que aplican pesos diferentes a los valores pasados, y estos decrecen exponencialmente desde la observación más reciente hasta la más lejana en el tiempo.

Los métodos de suavizamiento exponencial se clasifican en: Suavizamiento Exponencial Simple, Método Lineal de Holt (útil cuando hay tendencia) y Método de Holt – Winters (útil cuando hay tendencia y estacionalidad).

Todos estos métodos requieren de ciertos parámetros para su operatividad, estos parámetros están en el intervalo de $[0,1]$. Estos parámetros son los que dan diferentes pesos a los valores pasados. Así el método de suavizamiento exponencial simple (SES por sus siglas en inglés Single Exponential Smoothing) requiere que un solo parámetro sea estimado, EL método de Holt requiere de dos parámetros diferentes y es útil para pronosticar series de tiempo con tendencia y el

método de Holt – Winters que requiere de tres parámetros de suavizamiento y sirve para pronosticar series de tiempo con tendencia y estacionalidad.

Pegel (1969) clasificó a los métodos de suavizamiento exponencial por su tendencia y estacionalidad y dependiendo si estas son aditivas o multiplicativas. Las formas de las series de tiempo basadas en esta clasificación se muestran en la Figura 4.1. (Makridakis S., 1998).

Cabe destacar que el uso apropiado de un modelo es de vital importancia. Claramente un método inapropiado de pronóstico, así este optimizado, será inferior a un modelo de pronóstico apropiado. (Makridakis S., 1998)

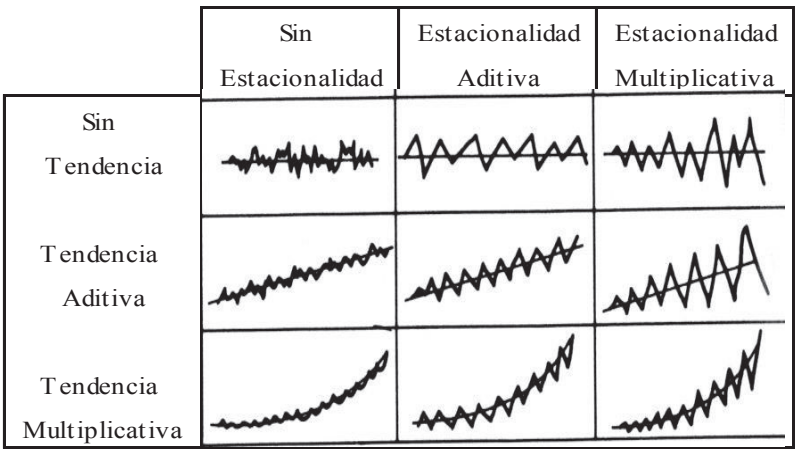


Figura 4.1 Patrones basados en la clasificación de Pegel (1969)
(Makridakis S., 1998 Pág. 138)

4.1.1 ESTRATEGIA PARA EL PRONÓSTICO

Una vez elegido el modelo de pronóstico, se ajusta el modelo a los datos conocidos (Eligiendo parámetros e inicializando el modelo) y se obtienen los ***valores ajustados***. Para los datos conocidos esto nos permite calcular el ***error de ajuste***, una medida de la bondad del ajuste del modelo, una vez que nuevas observaciones se vayan obteniendo se podrá calcular el ***error de pronóstico***. (Makridakis S., 1998)

En la figura 4.2 se describe la estrategia para evaluar los métodos de pronósticos.

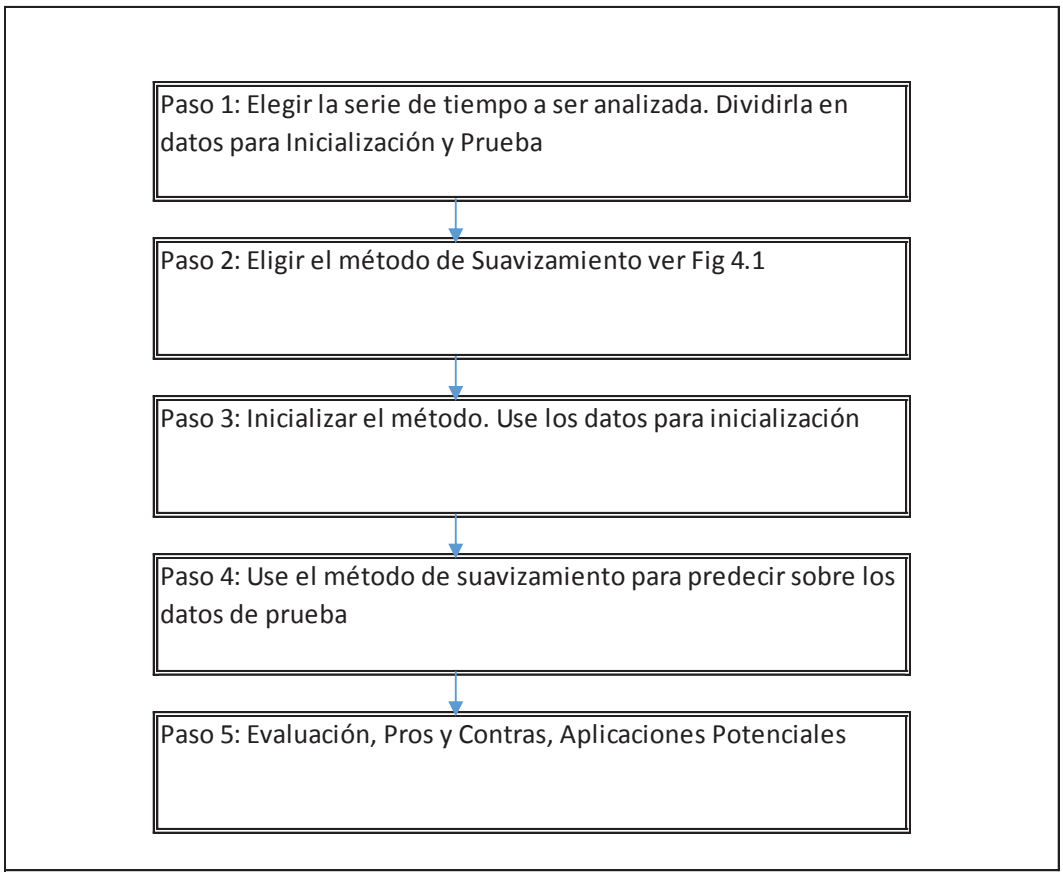


Figura 4.2 Estrategia para evaluar los métodos de suavizamiento para Pronósticos

(Makridakis S., 1998 Pág. 140)

Paso 1 La serie de tiempo de interés es dividida en dos partes (Un conjunto de datos para "inicialización" y un conjunto de datos para "prueba"), Así la evaluación del método de pronóstico puede ser efectuada.

Paso 2 Un método de Pronóstico es elegido de una lista de posibles métodos.

Paso 3 El conjunto de datos para inicialización es utilizado en el arranque del método de pronóstico. En este paso se estiman componentes de tendencia, estacionalidad y valores de parámetros.

Paso 4 El método se aplica a los datos de prueba para ver que tan bien se desempeña con datos que no fueron utilizados para estimar los componentes del modelo. Se calcula el error de pronóstico para tener una idea de la precisión del modelo, en esta etapa se puede modificar el proceso de inicialización y búsqueda de parámetros óptimos del modelo, para alcanzar una precisión adecuada.

Paso 5 Finalmente el método de pronóstico es evaluado para ver su seguridad de acuerdo al patrón de datos (Ver figura 4.1) y poder determinar su aplicación potencial.

4.2 FUNDAMENTO TEORICO DEL METODO DE HOLT - WINTERS

La base de los métodos de Holt y Holt – Winters es el método de suavizamiento exponencial simple, se tratará de una manera rápida para luego adentrarnos al método de Holt – Winters.

4.2.1 SUAVIZAMIENTO EXPONENCIAL SIMPLE

Si se necesita predecir el valor siguiente de una serie de tiempo Y_t que aún no ha sido observada. Su pronóstico se denota por F_t. Cuando la observación Y_t se vuelve disponible, el error de pronóstico se calcula como (Y_t - F_t). El método de suavizamiento exponencial simple toma el pronóstico del período anterior y lo ajusta utilizando el error de pronóstico. Así el pronóstico para el siguiente período será:

$$F_{t+1} = F_t + \alpha(Y_t - F_t). \quad (4.1)$$

Donde α toma un valor entre 0 y 1.

Se puede observar que el nuevo pronóstico F_{t+1} es igual al anterior pronóstico F_t más un ajuste producido por el error cometido en el último pronóstico (Y_t - F_t).

La ecuación (4.1) representa un principio básico de realimentación negativa, igual a los que se utilizan en los dispositivos de control automático como son: termostatos, pilotos automáticos, etc. El error del pronóstico pasado es utilizado para corregir el siguiente pronóstico en la dirección opuesta a este error. Habrá un ajuste hasta que el error sea corregido. (Makridakis S., 1998).

Otra forma de escribir la Ec. (4.1) es

$$F_{t+1} = \alpha Y_t + (1 - \alpha)\, F_t. \quad (4.2)$$

Donde el pronóstico F_{t+1} está basado en la ponderación de la observación más reciente Y_t con un peso α y el más reciente pronóstico F_t con un peso (1 - α).

La ecuación (4.2) es la forma general del suavizamiento exponencial. Esta forma reduce cualquier problema de almacenamiento, ya que no es necesario almacenar un conjunto de datos históricos (como era el caso de las medias móviles). En este caso solamente es necesario almacenar la observación más reciente, el pronóstico más reciente y el parámetro α .

Si se expande la ecuación (4.2) se puede obtener:

$$F_{t+1} = \alpha Y_t + \alpha(1-\alpha)\, Y_{t-1}. + \alpha(1-\alpha)^2\, Y_{t-2} + \alpha(1-\alpha)^3\, Y_{t-3} + \cdots + \alpha(1-\alpha)^{t-1}\, Y_1 + (1-\alpha)^t F_1 \quad (4.3)$$

Así F_{t+1} representa la media móvil ponderada de todas las observaciones pasadas. (Makridakis S., 1998).

Para un valor de α entre 0 y 1 los pesos de cada observación decrecen exponencialmente para las observaciones más antiguas, de allí su nombre de "*suavizamiento exponencial"*. (Hyndman R., 2014)

La inicialización del método y la optimización del parámetro α se verán más adelante en el método de Holt – Winters.

Lo que sí se puede observar es que para valores de α altos (Por ejemplo 0.9) nos dará un suavizamiento muy pequeño en el pronóstico ya que dará más peso a las observaciones más recientes mientras que para valores bajos (Por ejemplo 0.1) nos dará un suavizamiento considerable ya que dará más peso a las observaciones más distantes. (Makridakis S., 1998).

4.2.2 MÉTODO DE HOLT – WINTERS

Holt (1957) extendió el suavizamiento exponencial simple al suavizamiento exponencial lineal para permitir predecir datos con tendencia.

El pronóstico mediante el método de suavizamiento exponencial lineal de Holt utiliza dos constantes de suavizamiento, $\alpha\, y\, \beta$ con valores entre 0 y 1 y tres ecuaciones:

Nivel: $$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + b_{t-1}) \quad (4.4)$$

Tendencia: $$b_t = \beta(L_t - L_{t-1}) + (1-\beta) b_{t-1} \quad (4.5)$$

Pronóstico: $$F_{t+m} = (L_t + b_t * m) \quad (4.6)$$

Donde L_t denota un estimado del nivel de la serie al tiempo t y b_t denota un estimado de la pendiente de la serie al tiempo t.

La ecuación (4.4) ajusta directamente la tendencia del período anterior, b_{t-1} sumando el último valor suavizado L_{t-1} . Esto ayuda a eliminar el retraso y trae a L_t al nivel aproximado del valor actual.

La ecuación (4.5) entonces actualiza la tendencia, la cual es expresada como la diferencia entre los dos últimos valores suavizados. Esto es apropiado ya que si existe alguna tendencia en los datos, nuevos valores podrían ser mayores o menores que los valores anteriores. Si existe alguna aleatoriedad remanente, la tendencia es modificada suavizando con β la tendencia en el último período $(L_t - L_{t-1})$ y sumando la tendencia del período previo multiplicada por $(1-\beta)$. Así la ecuación (4.5) es similar a la forma básica del suavizamiento exponencial simple dado en la ecuación (4.2) pero aplicada a actualizar la tendencia. Finalmente la ecuación (4.6) para el pronóstico futuro. La tendencia b_t es multiplicada por el número de períodos a futuro m y sumada al valor base L_t (Makridakis S., 1998).

Si los datos no presentan patrones de tendencia o estacionalidad, medias móviles o suavizamiento exponencial simple son métodos apropiados. Si los datos presentan tendencia lineal, el método lineal de Holt es apropiado. Pero si los datos son estacionales, estos métodos no pueden manejar adecuadamente el problema. El método de Holt fue extendido por Winters (1960) para capturar la estacionalidad directamente. El método de Holt – Winters está basado en tres ecuaciones de suavizamiento, una para el nivel, otra para la tendencia y otra para la estacionalidad. Existen dos métodos de Holt – Winters diferentes, aditivo o multiplicativo, dependiendo de la forma como es modelada la estacionalidad (Makridakis S., 1998).

4.2.2.1 Estacionalidad Multiplicativa

Las ecuaciones básicas del método de Holt – Winters multiplicativo son las siguientes:

Nivel: $$L_t = \alpha \frac{Y_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + b_{t-1}) \quad (4.7)$$

Tendencia: $$b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \quad (4.8)$$

Estacionalidad: $S_t = \gamma \frac{Y_t}{L_t} + (1-\gamma) S_{t-s}$ (4.9)

Pronóstico: $F_{t+m} = (L_t + b_t * m) S_{t-s+m}$ (4.10)

Dónde: s es la longitud de la estacionalidad así: si es mensual s =12, trimestral s=3…, L_t representa el nivel de la serie, b_t denota la tendencia, S_t es el componente estacionario y F_{t+m} es el pronóstico m períodos adelante. (Makridakis S., 1998).

Los valores α , β y γ varían entre 0 y 1.

Mediante la ecuación (4.7) se actualiza la estimación de la base de la serie, tomando un promedio ponderado de las dos cantidades siguientes:

- $(L_{t-1} + b_{t-1})$, que es la estimación del nivel base antes de observar Y_t.
- La observación desestacionalizada $\frac{Y_t}{S_{t-s}}$, la cual es una estimación de la base obtenida a partir del período actual.

La ecuación (4.8) es idéntica a la ecuación (4.5) del método de Holt.

Por otro lado, la ecuación (4.9) actualiza y suaviza la estacionalidad del mes t, aplicando un promedio ponderado de las dos cantidades siguientes:

- La estimación más reciente de la estacionalidad S_{t-s},
- $\frac{Y_t}{L_t}$, la cual es una estimación de la estacionalidad del mes t, obtenida a partir del mes actual.

Al final del período t, la predicción F_{t+m} del período $(t+m)$ se obtiene con la ecuación (4.10), donde, para pronosticar el valor de la serie durante el período $(t+m)$ se multiplica la estimación base $(L_t + b_t * m)$ por la estimación más reciente del factor de estacionalidad S_{t-s+m} del período $(t+m)$. (Winston, 2005).

Es importante entender que en este método el nivel L_t es un valor suavizado que no incluye estacionalidad (estacionalmente ajustado) ya que al dividir Y_t para S_{t-s} se está eliminando la estacionalidad.

Se debe recordar que Y_t contiene estacionalidad y aleatoriedad, para suavizar esta aleatoriedad, la ecuación (4.9) realiza un promedio ponderado entre el factor estacional nuevo con γ y al valor estacional más reciente con $(1-\gamma)$. (Makridakis S., 1998).

4.2.2.2 Estacionalidad Aditiva

La componente de estacionalidad del método de Holt – Winters puede ser tratada aditivamente, aunque es menos común.

Las ecuaciones básicas son:

Nivel: $L_t = \alpha(Y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1})$ (4.11)

Tendencia: $b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1}$ (4.12)

Estacionalidad: $S_t = \gamma(Y_t - L_t) + (1-\gamma)S_{t-s}$ (4.13)

Pronóstico: $F_{t+m} = (L_t + b_t * m) + S_{t-s+m}$ (4.14)

La única diferencia con las ecuaciones del método de Holt – Winters multiplicativo es que los índices de estacionalidad son ahora sumados y restados en lugar de dividir y multiplicar. (Makridakis S., 1998)

4.2.2.3 Inicialización del Método de Holt - Winters

Para iniciar el método de Holt – Winters, se necesitan valores iniciales del nivel L_t, de la tendencia b_t, y de los índices estacionales S_t.Para determinar los valores iniciales de los índices estacionales necesitamos datos de al menos una estación completa es decir s períodos. Por lo tanto se inicializa el nivel y la tendencia al período s.

$$L_s = \frac{1}{s}(Y_1 + Y_2 + \cdots + Y_s) \quad (4.15)$$

Se puede notar que es una media móvil de orden s y así se elimina la estacionalidad en los datos. Para inicializar la tendencia es conveniente utilizar dos estaciones completas (2 s) períodos de la siguiente manera:

$$b_s = \frac{1}{s}\left[\frac{Y_{s+1}-Y_1}{s} + \frac{Y_{s+2}-Y_2}{s} + \cdots + \frac{Y_{s+s}-Y_s}{s}\right] \quad (4.16)$$

Cada uno de estos términos es un estimado de la tendencia en una estación completa y el valor inicial b_s es un promedio de s de tales términos.

Finalmente, los índices estacionales se inicializan utilizando la relación de los primeros datos y la media del primer año así:

$$S_1 = \frac{Y_1}{L_s}, \quad S_2 = \frac{Y_2}{L_s}, \ldots, \quad S_s = \frac{Y_s}{L_s}. \tag{4.17}$$

Existen otros valores de inicialización como los propuestos por Hyndman (Hyndman R., 2014) donde dichos valores inician desde el período t_0 en lugar de t_s, así los valores iniciales de nivel y tendencia son:

$$L_0 = \frac{1}{s}(Y_1 + Y_2 + \cdots + Y_s) \tag{4.18}$$

$$b_0 = \frac{1}{s}\left[\frac{Y_{s+1}-Y_1}{s} + \frac{Y_{s+2}-Y_2}{s} + \cdots + \frac{Y_{s+s}-Y_s}{s}\right] \tag{4.19}$$

$$S_0 = \frac{Y_s}{L_0}, \quad S_{-1} = \frac{Y_{s-1}}{L_0}, \ldots, \quad S_{-s+1} = \frac{Y_1}{L_0}. \tag{4.20}$$

Para estimar los parámetros $\alpha, \beta\ y\ \gamma$ se puede elegir aquellos que minimicen el error cuadrático medio (MSE) o el error porcentual absoluto medio (MAPE). Se puede utilizar un algoritmo de optimización no lineal para encontrar los valores óptimos de los parámetros.

La herramienta Solver de Excel tiene ya en sus nuevas versiones, algoritmos de optimización no lineal como se verá más adelante en un ejemplo.

Si bien los dos métodos arriba indicados son efectivos y simples, se analizarán dos métodos más elaborados para la inicialización como son: Retropredicción (Backcasting) y Descomposición, para luego aplicarlos en un ejemplo.

4.2.2.3.1 Retropredicción (Backcasting)

Es un método utilizado en la metodología de Box – Jenkins (Box G., 2008 4th Edition) que se puede aplicar a los métodos de suavizamiento exponencial y su fundamento teórico se lo puede profundizar en (Box G., 2008 4th Edition) pp 222-226.

Alan Pankratz hace un buen resumen de esta metodología en la referencia (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)pp: 220-223, el cual mostramos a continuación:

“Si se tiene una secuencia de observaciones $z_1, z_2, z_3, \ldots z_n$. Se considera algún valor subsecuente z_{n+l+1} con una cierta relación de probabilidad con los valores previos $z_1, z_2, z_3, \ldots z_n$. Se puede demostrar que un valor previo a la secuencia z_1,

$z_2,\ z_3, \ldots z_n$, (designado como z_{-l}) tiene la misma relación de probabilidad con la secuencia $z_n\ z_{n-1}, z_{n-2}, \ldots z_1$ como z_{n+l+1} tiene con la secuencia $z_1, z_2,\ z_3, \ldots z_n$."

En otras palabras si se reversan los datos de una serie e iniciamos el proceso de estimación desde el final (valor más reciente) y terminamos con el primero (valor más antiguo). Haciendo esto se proveerá el pronóstico o estimación de parámetros para el inicio de la serie, los cuales pueden ser utilizados como valores iniciales cuando los datos son pronosticados en la secuencia usual.

El procedimiento propuesto es el siguiente:

- Inicie con cualquier realización (serie de tiempo),
- Reverse la serie en el tiempo donde z_1 se vuelve la última observación y z_n se transforma en la primera observación,
- Pronostique la serie reversada (es decir pronostique hacia atrás la serie original) utilizando los más recientes valores de los coeficientes estimados. Por ejemplo si tenemos la siguiente ecuación de pronóstico z_t= 0.5 z_{t-1}. pero para la serie reversada la ecuación sería: z_t= 0.5 z_{t+1}. Así $z_0 = 0.5\, z_1$, $z_{-1} = 0.5\, z_0$ y así sucesivamente.
- Finalmente reverse la serie reversada con su pronóstico para obtener la serie original z_t.con sus valores previos obtenidos del pronóstico hacia atrás (backasting). (Pankratz, Forecasting with Dynamic Regression Models, 1991)

Así ya se pueden obtener los valores iniciales para el método de Holt – Winters. Más adelante se aclarará con un ejemplo.

4.2.2.3.2 Descomposición

Rob Hyndman en la referencia (Hyndman R., 2014) pp: 159-161, propone el siguiente procedimiento:

1.- Calcule una media móvil 2xS de los primeros dos años. Con esto hacemos un suavizamiento inicial.

2.- Entonces reste (en el caso de estacionalidad aditiva) o divida (en el caso de estacionalidad multiplicativa) la señal original para la señal suavizada en el numeral anterior, para obtener los datos sin tendencia. Los valores estacionales

iniciales se obtienen promediando los datos de los valores sin tendencia. Por ejemplo el valor inicial para Enero es el promedio de los valores de Enero de la señal sin tendencia.

3.- Luego reste (en el caso de estacionalidad aditiva) o divida (en el caso de estacionalidad multiplicativa) los valores originales para valores estacionales obtenidos en el numeral anterior, para obtener los datos desestacionalizados o estacionalmente ajustados.

6.- Calcule una regresión lineal de los datos estacionalmente ajustados, se obtienen: el valor inicial L_0 (la intersección) y el valor inicial de la tendencia b_0 (la pendiente).

Este procedimiento es generalmente muy bueno y rápido de implementar para producir pronósticos desde el período 1, sin embargo se requieren datos de dos o tres años anteriores. (Hyndman R., 2014).

De igual forma se aclarará este procedimiento con un ejemplo más adelante.

Cabe indicar que el método descrito corresponde a una descomposición clásica de series de tiempo, existen otros métodos más avanzados para descomposición de series de tiempo como son: ARIMA X-12 o X-13 y descomposición STL. Si al lector le interesa profundizar en el tema puede utilizar dos referencias: (Makridakis S., 1998) pp: 112 – 125 y (Hyndman R., 2014) pp: 161-166. El paquete estadístico EViews ver.9 tiene implementada la opción para descomposición de series de tiempo mediante el método ARIMA X-12 y X-13, también se lo puede utilizar.

4.2.2.3.3 Optimización

Todos los métodos de suavizamiento exponencial requieren de ciertas especificaciones de los parámetros de suavizamiento. Ya que estos controlan que tan rápido el pronóstico reaccionará ante cambios en los datos.

Con el advenimiento de computadores mucho más rápidos se ha convertido en una tarea mucho más fácil seleccionar parámetros óptimos utilizando algoritmos de optimización no lineal. Es decir software especializado puede minimizar el error cuadrático medio (MSE) y obtener los valores de los parámetros automáticamente.

Es posible minimizar otras medidas del error como (MAE o MAPE) pero el MSE es el más fácil de utilizar. (Makridakis S., 1998).

Una alternativa para no preocuparse en encontrar valores óptimos es encontrar valores iniciales confiables de las componentes $L_t, b_t \; y \; S_t$ luego especificar valores pequeños de los parámetros de suavizamiento (alrededor de 0.1 a 0.2), se obtendrá una respuesta lenta pero estable en el tiempo. (Makridakis S., 1998).

Chris Chatfield y Mohammad Yar en su artículo "Holt – Winters Forecasting: some practical isuues" (Chatfield C., 1988), se recomienda utilizar valores pequeños de los parámetros así ($\alpha \; y \; \gamma$) no excedan valores de 0.3 y el valor de β menor a 0.1 pudiendo tomar el valor de 0. El pronóstico reaccionará lentamente pero de una manera continua a cambios en los datos. La desventaja de esta estrategia es que brinda una respuesta lenta al pronóstico, pero vale la pena ya que a largo plazo dará estabilidad y es un método de bajo costo.

Se sugiere además en el artículo que si una serie tiene un crecimiento exponencial los valores de α y β podrían ser mayores a 1 y esto indica que este método se torna inseguro, como podría ocurrir también con valores fuera de la región de estabilidad. En estos casos se sugiere utilizar la tecnología de Box – Jenkins, que se tratará en el capítulo 5.

Además se debe tomar en cuenta las siguientes sugerencias respecto a los parámetros de suavizamiento exponencial:

- Comportamientos Inusuales (Outliers), altas variaciones aleatorias y ausencia de estabilidad en la tendencia y variación estacional mantienen bajos los parámetros óptimos,
- Un cambio de estructura en la serie generalmente mantiene valores altos,
- No se logra la optimización de los parámetros de suavizamiento solamente por las propiedades de la serie, sino también por la elección de los valores iniciales.
- No necesariamente los parámetros que dan una mejor optimización dan el mejor pronóstico.

Se recomienda lo siguiente en el citado artículo:

- Los valores iniciales se calculen utilizando datos de uno o dos años o utilizando Retro predicción (Backcasting).

- Los parámetros de suavizamiento sean optimizados automáticamente en una longitud segura de datos históricos.
- Si lo último no es posible utilizar los siguientes valores iniciales: $\alpha = \beta = 0.4$ y $\gamma = 0.1$, para arrancar el método.

4.2.2.3.4 Intervalos de Predicción

Los pronósticos considerados hasta ahora han sido ***pronósticos puntuales***, es decir simples números que se piensa la serie tendrá en el futuro. Esto es a veces lo requerido, pero a menudo es deseable tener una medida con cierta incertidumbre asociada al pronóstico.

Los valores que miden la precisión del pronóstico como son MSE o MAPE nos dan una idea de la incertidumbre en el pronóstico, pero un método más intuitivo es brindar un ***intervalo de predicción*** que es el rango en el cual el pronosticador puede estar bastante seguro de que los verdaderos valores tomarán. (Makridakis S., 1998).

Desafortunadamente los métodos de suavizamiento exponencial no permiten un cálculo fácil de intervalos de predicción. Un enfoque bastante utilizado es encontrar un modelo estadístico para el cual un método de suavizamiento exponencial es óptimo. Entonces los intervalos de predicción pueden ser obtenidos del modelo estadístico. La equivalencia entre métodos de suavizamiento exponencial y modelos estadísticos se discutirá en el capítulo 5 sección (5.2.1.8.5.5).

Además si el lector desea profundizar Rob Hyndman en su referencia (Hyndman R., 2014) pp: 197 - 202, da una equivalencia entre métodos de suavizamiento exponencial y modelos de espacio de estado. En esta equivalencia se notan dos ventajas: primero se puede calcular intervalos de predicción y segundo se puede escoger cualquier modelo de una manera objetiva utilizando los criterios de selección de modelos que se verán en el capítulo 5. (Hyndman R., 2014).

Sin embargo existen algunas dificultades con estos enfoques:

- Los métodos de suavizamiento exponencial pueden ser utilizados aunque los datos no satisfagan un modelo estadístico para el cual el método es óptimo. En efecto los métodos de suavizamiento exponencial fueron

desarrollados como un enfoque ampliamente aplicable a una variedad de series de tiempo. Ellos nunca intentaron ser óptimos en el sentido de un modelo estadístico subyacente.

- En los modelos estadísticos se asume que los errores de pronóstico no son correlacionados y esto es crucial para calcular los intervalos de predicción, particularmente intervalos de predicción varios pasos a futuro. A menudo un método de suavizamiento exponencial será aplicado y dará errores de pronóstico que están correlacionados. Entonces los intervalos de predicción no serán válidos.
- Para algunos algoritmos de suavizamiento exponencial el modelo estadístico es desconocido. (Makridakis S., 1998).

Por estas razones, se debe tener mucha precaución al usar intervalos de predicción y se deben chequear los errores de pronóstico que satisfagan las asunciones citadas. (Makridakis S., 1998).

Chris Chatfield y Mohammad Yar en su artículo “Holt – Winters Forecasting: some practical isuues” (Chatfield C., 1988), recomiendan seguir el siguiente procedimiento para pronosticar con el método de Holt – Winters:

1.- Graficar las series y examinar su estructura, para esto es muy útil la figura 4.1 mostrada al inicio de este capítulo. El método de Holt – Winters es seguro para series de tiempo dominados por la tendencia y la estacionalidad. Para series con crecimiento exponencial y con discontinuidades la metodología de Box – Jenkins es recomendable.

2.- Examinar potenciales comportamientos inusuales (Outliers) y considerar ajustar observaciones sospechosas preferiblemente después tomando en cuenta información externa. Ajustes por variaciones de calendario podrían necesitar ser consideradas.

3.- Describir la forma de la variación estacional, si existe, escoger el método de suavizamiento exponencial más apropiado. (Ver figura 4.1). Considere la posibilidad de transformar datos, pero tener en mente que el modelo con los datos originales es normalmente de mayor ayuda para el pronóstico.

4.- Calcular valores de arranque para en nivel, tendencia y estacionalidad y estime los parámetros de suavizamiento con cualquiera de los métodos descritos en este capítulo.
5- Chequee el error de pronóstico, particularmente su autocorrelación. Si los errores están autocorrelacionados entonces Holt-Winters no es óptimo, se deberá utilizar otro método de pronóstico o una variación del método de Holt – Winters.
6.- Calcule el pronóstico tan adelante como requiera. Decida si el pronóstico necesita ser ajustado subjetivamente por alguna razón.
Al final de este mismo artículo los autores muestran como desarrollo futuro al método de Holt – Winters amortiguado, mismo que se tratará a continuación.

4.2.2.3.5 Método de Holt-Winters Amortiguado

Evidencia empírica indica que los métodos de suavizamiento exponencial tienden a sobre-pronosticar, especialmente cuando los horizontes de predicción son largos. Motivados por esta observación Gardner y McKenzie (1985) introducen un parámetro adicional que amortigua la tendencia a una línea plana en algún momento en el futuro. Estos métodos que incluyen una tendencia amortiguada han sido probados con muy buenos resultados, por esta razón se han vuelto muy populares para el pronóstico de muchas series de tiempo. (Hyndman R., 2014)
Las ecuaciones del método de Holt - Winters Amortiguado son muy similares a las del método de Holt – Winters sin amortiguamiento visto anteriormente, las mismas se muestran a continuación:

Nivel: $$L_t = \alpha \frac{Y_t}{S_{t-s}} + (1-\alpha)(L_{t-1} + \emptyset b_{t-1}) \quad (4.21)$$

Tendencia: $$b_t = \beta(L_t - L_{t-1}) + (1-\beta)\emptyset b_{t-1} \quad (4.22)$$

Estacionalidad: $$S_t = \gamma \frac{Y_t}{(L_{t-1} + \emptyset b_{t-1})} + (1-\gamma) S_{t-s} \quad (4.23)$$

Pronóstico: $$F_{t+m} = (L_t + b_t * (\emptyset + \emptyset^2 + \cdots + \emptyset^m) S_{t-s+m} \quad (4.24)$$

Valores iniciales se pueden calcular con las ecuaciones (4.18), (4.19) y (4.20), los parámetros de amortiguamiento se los puede calcular optimizando el MSE o MAPE, tal como se ha visto anteriormente.

4.3 EJEMPLO PRÁCTICO DEL METODO DE HOLT – WINTERS

Se han tomado los datos de la referencia (Makridakis S., 1998) pp: 161-168, para implementar de una manera práctica el método de Holt - Winters. En la Tabla 4.1 se muestran los datos para el ejemplo.

Tabla 4.1 Datos Ejemplo Método de Holt - Winters

Año	Cuarto de Año	Período t	Ventas Yt	Año	Cuarto de Año	Período	Ventas
1	1	1	362	4	1	13	544
	2	2	385		2	14	582
	3	3	432		3	15	681
	4	4	341		4	16	557
2	1	5	382	5	1	17	628
	2	6	409		2	18	707
	3	7	498		3	19	773
	4	8	387		4	20	592
3	1	9	473	6	1	21	627
	2	10	513		2	22	725
	3	11	582		3	23	854
	4	12	474		4	24	661

(Makridakis S., 1998 Pág. 162)

Los datos corresponden a las exportaciones por cuartos de año de una compañía francesa en un período de seis años.

Acorde con la estrategia de pronóstico planteada al inicio de este capítulo, dividiremos los datos en dos conjuntos: conjunto de datos para inicialización (dos años: períodos del 1 al 9) y conjunto de datos para prueba (períodos del 10 al 24) donde se realizará el análisis de errores.

Se seguirá el procedimiento propuesto para realizar el pronóstico en este ejemplo.

1.- Gráfico de la Serie de Tiempo.

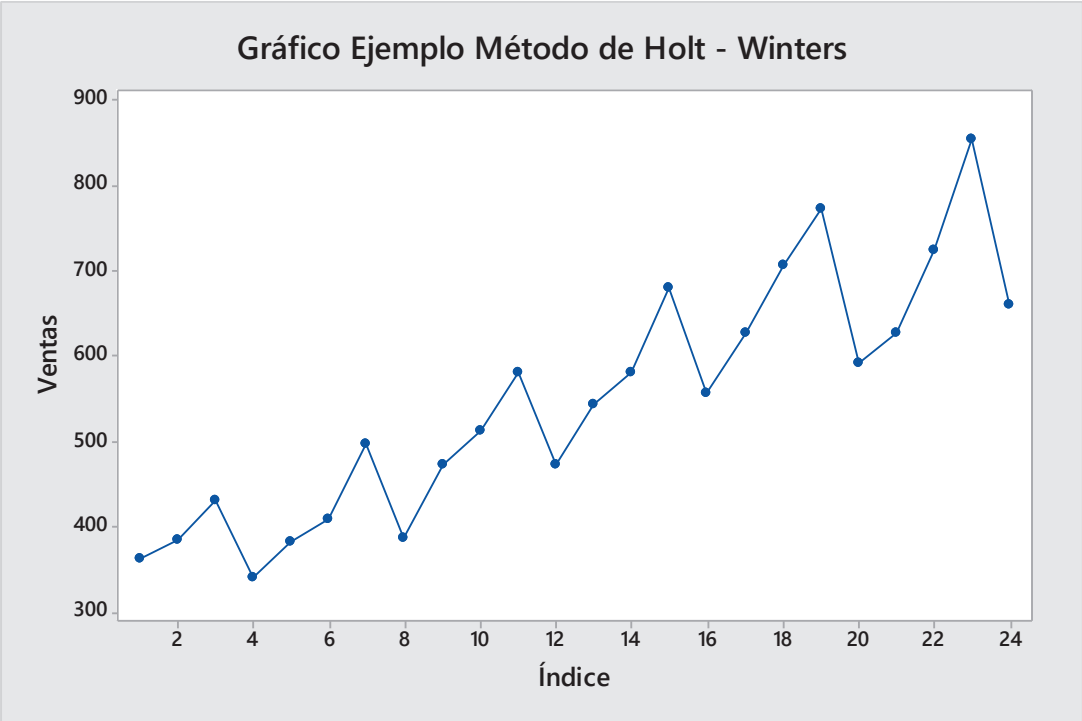


Figura 4.3 Gráfico Ejemplo Método de Holt – Winters

(Gráfico realizado por el autor de esta tesis)

La forma de la serie, sugiere la presencia de tendencia y estacionalidad, entonces el método de Holt – Winters es aplicable en este caso.

2.- Después de examinar los datos no se observa la presencia de comportamientos inusuales (Outliers) o discontinuidades,

3.- Si se observa la Figura 4.1 y comparamos con la figura 4.3, se puede concluir que existe una tendencia aditiva y una estacionalidad multiplicativa, razón por la cual se utilizá el método de Holt – Winters con estacionalidad multiplicativa.

4.- Cálculo de valores iniciales para Nivel, tendencia e índices estacionales y estimación de parámetros de suavizamiento exponencial.

En este punto se calcularán los valores iniciales de tres formas: Método propuesto por Makridakis, Retro predicción (Backcasting) y Descomposición.

Los pasos 4 5 y 6 se muestran a continuación en las tablas 4.2, 4.4, y 4.7.

En el método propuesto por Makridakis (Makridakis S., 1998) mostrado en la tabla 4.2, se utilizan las ecuaciones (4.15), (4.16) y (4.17) para calcular los valores

iniciales L_4 y b_4. Para el resto de cálculos iterativos se utilizan las ecuaciones: (4.12), (4.13) y (4.14) correspondientes al método de Holt – Winters Multiplicativo.

Tabla 4.2 Ejemplo Método de Holt – Winters Propuesto por Makridakis

Alfa	Beta	Gamma	MSE
0.822	0.055	0	582.92

Período t	Actual Yt	Nivel Lt	Tendencia bt	Estacionalidad St	Pronóstico Ft	Error Cuadrático Medio MSE	
1	362			0.953			
2	385			1.013			
3	432			1.137			
4	341	380	9.75	0.897			
5	382	398.99	10.26	0.953	371.29	114.74	
6	409	404.68	10.01	1.013	414.64	31.77	
7	498	433.90	11.06	1.137	471.43	705.87	
8	387	433.70	10.44	0.897	399.29	151.10	
9	473	487.20	12.81	0.953	423.11	2489.51	
10	513	505.21	13.10	1.013	506.59	41.12	
11	582	513.08	12.81	1.137	589.24	52.36	
12	474	527.80	12.92	0.897	471.92	4.35	
13	544	565.65	14.29	0.953	515.10	835.19	
14	582	575.42	14.04	1.013	587.57	31.00	
15	681	597.32	14.47	1.137	670.12	118.35	
16	557	619.12	14.87	0.897	549.01	63.90	
17	628	654.73	16.01	0.953	603.96	577.88	
18	707	693.00	17.24	1.013	679.58	752.12	
19	773	685.34	15.87	1.137	807.43	1185.36	
20	592	667.09	13.99	0.897	629.25	1387.35	
21	627	662.25	12.96	0.953	648.83	476.37	
22	725	708.40	14.78	1.013	684.10	1673.11	
23	854	746.22	16.05	1.137	822.14	1014.94	
24	661	741.17	14.89	0.897	684.03	530.49	
25				0.953	720.24	m=	1
26				1.013	781.09	m=	2
27				1.137	893.37	m=	3
28				0.897	718.54	m=	4
29				0.953	776.98	m=	5
30				1.013	841.43	m=	6

Inicialización del Método

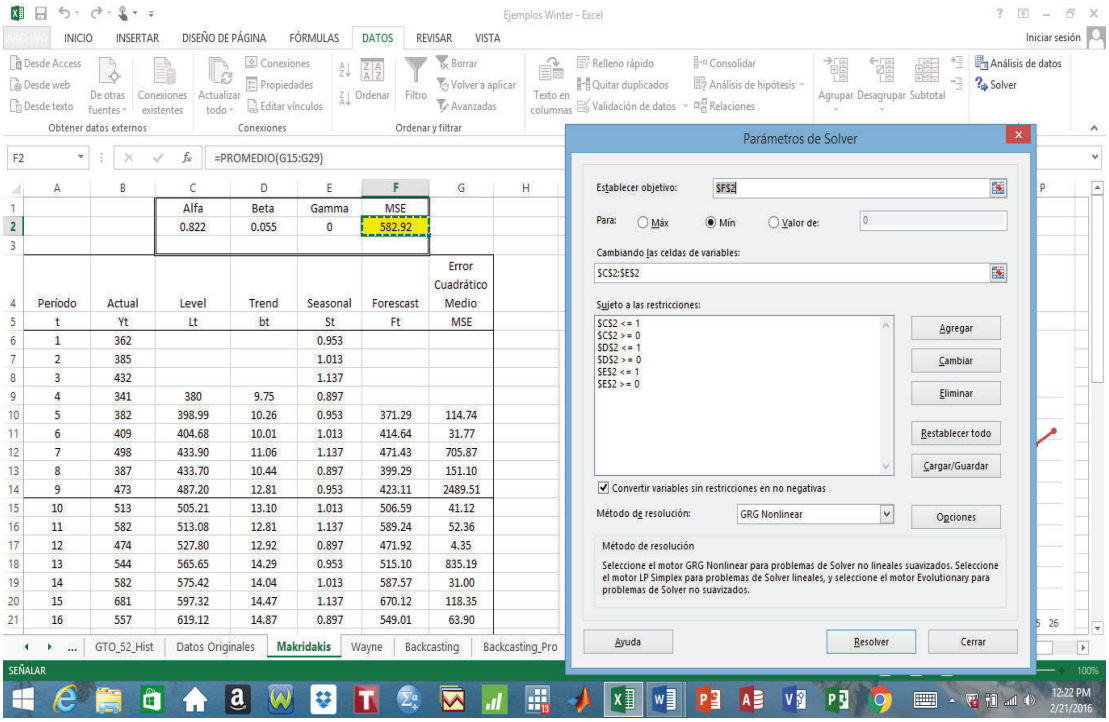
s= 4

L4	bs	St	
380	5	0.953	1
	6	1.013	2
	16.5	1.137	3
	11.5	0.897	4

b4= 9.75

(Makridakis S., 1998 Pág. 167)

Los parámetros $\alpha, \beta\ y\ \gamma$ se calculan utilizando la herramienta Solver de Excel optimizando el MSE como se muestra en la figura 4.4



	A	B	C	D	E	F	G
1			Alfa	Beta	Gamma	MSE	
2			0.822	0.055	0	582.92	
3							
4	Período	Actual	Level	Trend	Seasonal	Forecast	Error Cuadrático Medio
5	t	Yt	Lt	bt	St	Ft	MSE
6	1	362			0.953		
7	2	385			1.013		
8	3	432			1.137		
9	4	341	380	9.75	0.897		
10	5	382	398.99	10.26	0.953	371.29	114.74
11	6	409	404.68	10.01	1.013	414.64	31.77
12	7	498	433.90	11.06	1.137	471.43	705.87
13	8	387	433.70	10.44	0.897	399.29	151.10
14	9	473	487.20	12.81	0.953	423.11	2489.51
15	10	513	505.21	13.10	1.013	506.59	41.12
16	11	582	513.08	12.81	1.137	589.24	52.36
17	12	474	527.80	12.92	0.897	471.92	4.35
18	13	544	565.65	14.29	0.953	515.10	835.19
19	14	582	575.42	14.04	1.013	587.57	31.00
20	15	681	597.32	14.47	1.137	670.12	118.35
21	16	557	619.12	14.87	0.897	549.01	63.90

Figura 4.4 Optimización MSE con Solver

Se puede notar en la parte inferior de la ventana de los parámetros de Solver que el método de resolución es: GRG Non Linear.

Además se puede notar que el MSE es 582.92 para este método.

En la tabla 4.3 se inicializa el método de Holt – Winters mediante Backcasting.

Así los valores iniciales para el método de Holt – Winters son: $l_0 = 360.79$ y $b_0 = 16.50$.

En la tabla 4.4 se muestran los resultados obtenidos con estos valores iniciales aplicados al método de Holt – Winters. Para calcular los índices estacionales se utiliza esta vez las ecuaciones: (4.18), (4.19) y (4.20).ya que se inició el pronóstico desde el período t_1. La optimización de parámetros de suavizamiento exponencial se la realiza igual con Solver, indicado anteriormente.

Se puede notar que para este método el MSE fue de 528.17.

Tabla 4.3 Retropredicción (Backcasting) del Ejemplo

Período t	Actual Yt	Nivel Lt	Tendencia bt	Indices Estacionales St	Pronóstico Ft	Error Cuadrático Medio MSE	
24	661			0.9222			
23	854			1.1915			
22	725			1.0115			
21	627	716.75	-10.44	0.8748			
20	592	689.31	-12.65	0.8702	651.37	3525.31	
19	773	669.29	-13.61	1.1615	806.23	1104.41	
18	707	667.11	-12.12	1.0511	663.23	1915.65	
17	628	671.60	-9.96	0.9243	572.97	3027.88	
16	557	655.95	-10.70	0.8529	575.77	352.18	
15	681	629.68	-12.73	1.0959	749.46	4686.58	
14	582	600.25	-14.90	0.9842	648.50	4421.81	
13	544	586.20	-14.79	0.9273	541.01	8.94	
12	474	567.27	-15.33	0.8387	487.38	178.90	
11	582	546.43	-16.05	1.0706	604.85	522.16	
10	513	527.97	-16.36	0.9739	522.03	81.49	
9	473	511.20	-16.41	0.9256	474.43	2.05	
8	387	485.98	-17.56	0.8039	414.97	782.58	
7	498	467.56	-17.67	1.0661	501.49	12.20	
6	409	441.98	-18.70	0.9341	438.15	849.50	
5	382	420.48	-19.06	0.9116	391.81	96.18	
4	341	407.42	-18.28	0.8310	322.71	334.36	
3	432	393.39	-17.73	1.0924	414.86	293.66	
2	385	385.30	-16.48	0.9875	350.90	1163.03	
1	362	376.30	-15.50	0.9530	336.20	665.42	
0	299.83	360.79	-15.50	**0.8310**	299.83	m=	1
-1	377.20	345.29	-15.50	1.0924	377.20	m=	2
-2	325.68	329.79	-15.50	0.9875	325.68	m=	3
-3	299.50	314.28	-15.50	0.9530	299.50	m=	4
-4	263.84	303.72	-14.86	0.8619	248.30	m=	5
-5	315.55	288.86	-14.86	1.0924	315.55	m=	6

Inicialización del Método (Makridakis)

s = 4

L4	bs	St	
716.75	-17.25	0.922	1
	-20.25	1.191	2
	-4.5	1.012	3
	0.25	0.875	4

b4= -10.4375

Tabla 4.4 Pronóstico utilizando inicialización con Retropredicción

Alfa	Beta	Gamma	MSE
1.00	0.00	0.25	528.17

Período t	Actual Yt	Nivel Lt	Tendencia bt	Estacionalidad St	Pronóstico Ft	Error Cuadrático Medio MSE	
0	299.83	360.79	15.50				
1	362	379.85	15.50	0.953	-		
2	385	380.06	15.50	1.013	-		
3	432	379.95	15.50	1.137	-		
4	341	380.16	15.50	0.897	-		
5	382	400.84	15.50	0.953	377.06	24.40	
6	409	403.75	15.50	1.013	421.75	162.61	
7	498	437.99	15.50	1.137	476.69	454.17	
8	387	431.44	15.50	0.897	406.78	391.44	
9	473	496.33	15.50	0.953	425.93	2215.39	
10	513	506.42	15.50	1.013	518.48	30.04	
11	582	511.87	15.50	1.137	593.42	130.40	
12	474	528.43	15.50	0.897	473.05	0.90	
13	544	570.83	15.50	0.953	518.36	657.23	
14	582	574.53	15.50	1.013	593.95	142.83	
15	681	598.94	15.50	1.137	670.87	102.71	
16	557	620.96	15.50	0.897	551.16	34.14	
17	628	658.97	15.50	0.953	606.55	460.31	
18	707	697.93	15.50	1.013	683.24	564.55	
19	773	679.86	15.50	1.137	811.17	1456.68	
20	592	659.98	15.50	0.897	623.74	1007.25	
21	627	657.92	15.50	0.953	643.73	279.90	
22	725	715.70	15.50	1.013	682.18	1833.82	
23	854	751.10	15.50	1.137	831.37	512.13	
24	661	736.90	15.50	0.897	687.64	709.67	528.17
25				0.953	717.04	m= 1	
26				1.013	777.88	m= 2	
27				1.137	890.73	m= 3	
28				0.897	716.61	m= 4	
29				0.953	775.82	m= 5	
30				1.013	840.82	m= 6	

Inicialización del Método (Hyndman)

s = 4

bo= 9.750

Lo= 380.000

Indices Estacionales	Si
S -3	0.953
S -2	1.013
S -1	1.137
S 0	0.897

En la tabla 4.7 se muestra el pronóstico del ejemplo, inicializando el método de Holt – Winters mediante descomposición clásica de series de tiempo.

Se realiza la descomposición tal como se describió anteriormente en 4.2.2.3.2 correspondiente a descomposición de series de tiempo. En la tabla 4.5 se muestra la descomposición clásica efectuada a los datos del ejemplo.

Tabla 4.5 Descomposición Clásica del Ejemplo

Período t	Actual Yt	4MA	2x4MA	Yt/Tt Rt Sin Tendencia	St	Estacionalidad Ajustada	Et
1	362				92.598	390.94	
2	385	380	382.5	100.654	99.196	388.12	3.881
3	432	385	388	111.340	110.388	391.35	3.913
4	341	391	399.25	85.410	84.448	403.80	4.038
5	382	407.5	413.25	92.438	92.598	412.54	4.125
6	409	419	430.375	95.033	99.196	412.32	4.123
7	498	441.75	454.75	109.511	110.388	451.14	4.511
8	387	467.75	478.25	80.920	84.448	458.27	4.583
9	473	488.75	499.625	94.671	92.598	510.81	5.108
10	513	510.5	519.375	98.773	99.196	517.16	5.172
11	582	528.25	536.875	108.405	110.388	527.23	5.272
12	474	545.5	557.875	84.965	84.448	561.29	5.613
13	544	570.25	580.625	93.692	92.598	587.49	5.875
14	582	591	601.5	96.758	99.196	586.72	5.867
15	681	612	627.625	108.504	110.388	616.91	6.169
16	557	643.25	654.75	85.071	84.448	659.58	6.596
17	628	666.25	670.625	93.644	92.598	678.20	6.782
18	707	675	674.875	104.760	99.196	712.73	
19	773	674.75	677	114.180	110.388	700.26	
20	592	679.25	689.375	85.875	84.448	701.02	
21	627	699.5	708.125	88.544	92.598	677.12	
22	725	716.75			99.196	730.88	
23	854				110.388	773.63	
24	661				84.448	782.73	

Con la ayuda de un gráfico se puede observar paso a paso la descomposición, ver figura 4.5, luego se realiza la regresión lineal de los datos con estacionalidad ajustada (desestacionalizados), para así obtener los valores iniciales l_0 y b_0, que se utilizan para arrancar el método de Holt – Winters, Ver tabla 4.6.

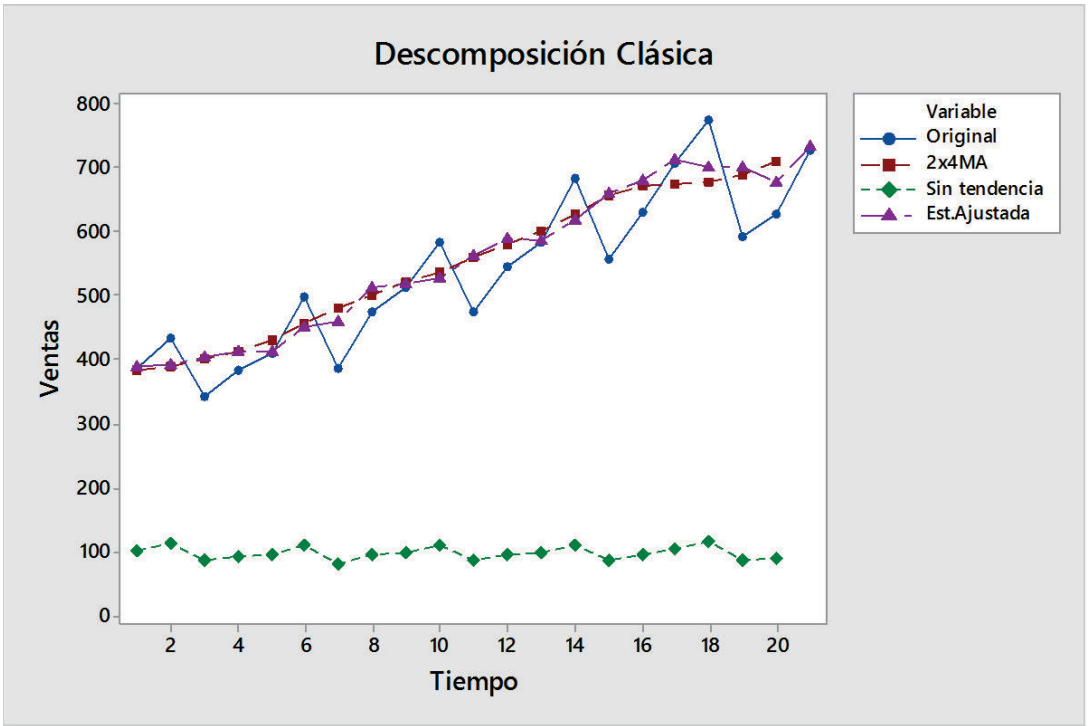


Figura 4.5 Gráfico Descomposición Clásica del Ejemplo

Tabla 4.6 Regresión Lineal Ejemplo con Estacionalidad Ajustada

Resumen		
Estadísticas de la regresión		
Coeficiente de correlación múltiple	0.9871119	
Coeficiente de determinación R^2	0.974389903	
R^2 ajustado	0.973225808	
Error típico	21.85101741	
Observaciones	24	
ANÁLISIS DE VARIANZA		
	Grados de libertad	*Suma de cuadrados*
Regresión	1	399657.1282
Residuos	22	10504.27316
Total	23	410161.4014
	Coeficientes	*Error típico*
Intercepción	334.9833483	9.206936799
Variable X 1	18.64210121	0.64435135

Tabla 4.7 Pronóstico Inicializando con Descomposición Clásica

	Alfa	Beta	Gamma	MSE	
	0.03	0.30	0.00	473.46	

Período t	Actual Yt	Nivel Lt	Tendencia bt	Indices Estacionales St	Pronóstico Ft	Error Cuadrático Medio MSE	
0		334.98	18.64				
1	362	354.30	18.85	0.953	-		
2	385	373.33	18.90	1.013	-		
3	432	391.91	18.80	1.137	-		
4	341	409.92	18.56	0.897	-		
5	382	427.76	18.35	0.953	408.34	693.92	
6	409	445.01	18.02	1.013	451.91	1841.30	
7	498	462.38	17.82	1.137	526.46	810.08	
8	387	478.93	17.44	0.897	430.74	1912.93	
9	473	496.37	17.44	0.953	473.04	0.00	
10	513	513.61	17.38	1.013	520.49	56.04	
11	582	530.50	17.23	1.137	603.74	472.68	
12	474	547.23	17.08	0.897	491.31	299.74	
13	544	564.48	17.13	0.953	537.78	38.63	
14	582	581.42	17.08	1.013	589.17	51.38	
15	681	598.51	17.08	1.137	680.49	0.26	
16	557	615.73	17.12	0.897	552.18	23.20	
17	628	633.53	17.33	0.953	603.10	619.78	
18	707	652.07	17.69	1.013	659.31	2274.04	
19	773	670.03	17.77	1.137	761.53	131.65	
20	592	687.08	17.55	0.897	616.96	622.95	
21	627	703.42	17.19	0.953	671.52	1981.78	
22	725	720.48	17.15	1.013	729.98	24.79	
23	854	737.98	17.26	1.137	838.69	234.39	
24	661	754.76	17.11	0.897	677.45	270.61	
25				0.953	735.60	m=	1
26				1.013	799.25	m=	2
27				1.137	916.54	m=	3
28				0.897	738.42	m=	4
29				0.953	800.52	m=	5
30				1.013	868.72	m=	6

Inicialización del Método (Hyndman)		Indices Estacionales	St
s =	4	S -3	0.953
bo=	9.750	S -2	1.013
Lo=	380.000	S -1	1.137
		S 0	0.897

El error mediante este método de inicialización es de: 473.46.

Ahora a este ejemplo se aplicará el método de Holt – Winters Amortiguado, En la tabla 4.8 se presentan los resultados.

Tabla 4.8 Pronóstico con Método de Holt – Winters Amortiguado

	Alfa	Beta	Gamma	Phi	MSE	
	1.00	0.00	0.06	0.99	529.76	
Período t	Actual Yt	Level Lt	Trend bt	Seasonal St	Forescast Ft	Error Cuadrático Medio MSE	
0		334.98	18.64				
1	362	380.00	18.45	0.953	336.69	640.49	
2	385	380.00	18.27	1.013	403.51	342.60	
3	432	380.00	18.09	1.137	452.56	422.77	
4	341	380.00	17.91	0.897	357.07	258.18	
5	382	400.99	17.73	0.953	378.89	9.69	
6	409	403.69	17.55	1.013	424.23	231.97	
7	498	438.06	17.37	1.137	478.88	365.55	
8	387	431.26	17.20	0.897	408.69	470.36	
9	473	496.52	17.03	0.953	427.22	2095.99	
10	513	506.34	16.86	1.013	520.30	53.36	
11	582	511.94	16.69	1.137	594.79	163.60	
12	474	528.21	16.52	0.897	474.38	0.14	
13	544	571.05	16.36	0.953	518.93	628.49	
14	582	574.44	16.19	1.013	595.14	172.55	
15	681	599.03	16.03	1.137	671.46	91.04	
16	557	620.70	15.87	0.897	551.93	25.66	
17	628	659.23	15.71	0.953	606.42	465.63	
18	707	697.82	15.56	1.013	683.82	537.32	
19	773	679.95	15.40	1.137	810.99	1443.47	
20	592	659.71	15.25	0.897	623.99	1023.25	
21	627	658.18	15.09	0.953	642.98	255.40	
22	725	715.58	14.94	1.013	682.13	1837.93	
23	854	751.20	14.79	1.137	830.49	552.55	
24	661	736.60	14.65	0.897	687.38	695.97	529.76
25					715.52	m= 1	
26					775.52	m= 2	
27					886.35	m= 3	
28					712.27	m= 4	
29					0.00	m= 5	
30					0.00	m= 6	

Inicialización del Método (Hyndman)		Indices Estacionales	
s = 4		S -3	0.953
Lo= 380.000		S -2	1.013
bo= 9.750		S -1	1.137
		S 0	0.897

A continuación se presenta la tabla 4.9 que resume el error producido por los diferentes métodos de inicialización aplicados a los datos del ejemplo.

Tabla 4.9 Resumen de errores producidos por los diferentes métodos de inicialización

	Alfa	Beta	Gamma	MSE	
Makridakis	0.822	0.055	0.000	582.92	
Backcasting	1.000	0.000	0.252	528.17	
Descomposición	0.026	0.302	0.000	473.46	MSE
Amortiguado	1.000	0.000	0.063	0.98	538.81

Finalmente se utilizará el paquete estadístico Minitab ver. 17 con este ejemplo para ver gráficamente el comportamiento del pronóstico y analizar ciertas propiedades del error comentadas anteriormente.

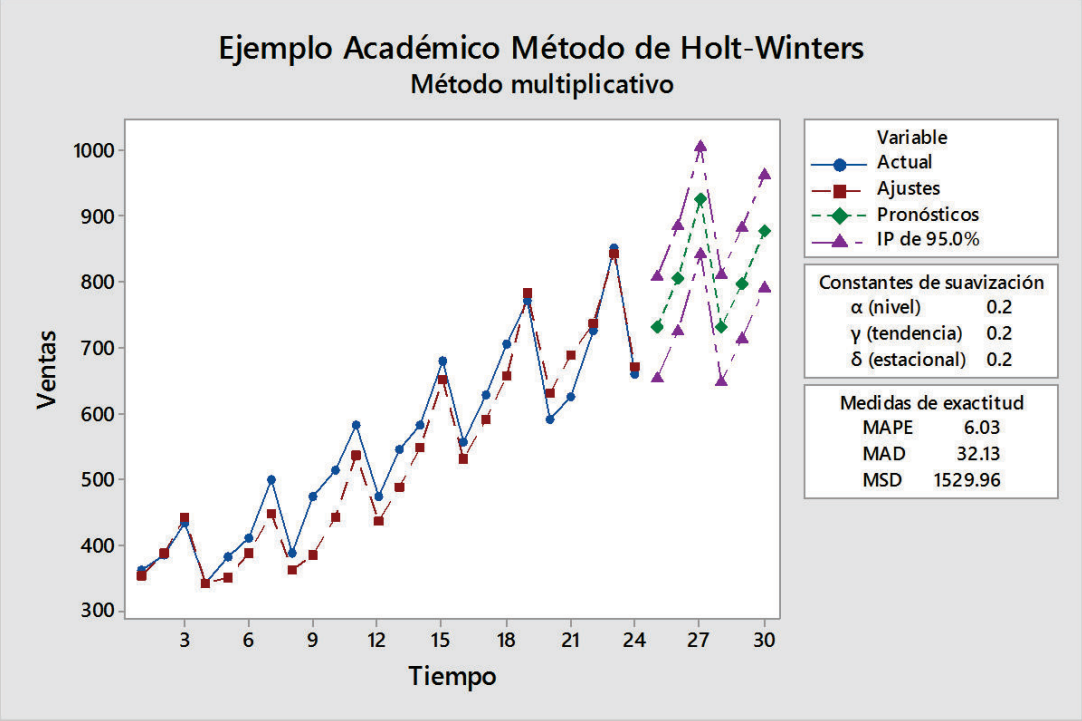


Figura 4.6 Gráfico Pronóstico Método de Holt – Winters Ejemplo)

(Realizado con el Programa Minitab ver 17.)

En la tabla 4.10 se muestra los cálculos realizados por el programa Minitab ver 17.y al final el pronóstico varios pasos a futuro. (Períodos 25 al 30).

Tabla 4.10 Pronóstico con Programa Minitab 17 Ejemplo

Período	Actual	SMOO1	LEVE1	TREN1	SEAS1
1	362.000	355.629	389.370	-1.217	0.926
2	385.000	393.933	381.896	-1.561	1.012
3	432.000	441.537	374.837	-1.863	1.156
4	341.000	339.600	375.776	-1.709	0.906
5	382.000	348.013	405.638	0.027	0.926
6	409.000	410.392	404.512	-0.036	1.012
7	498.000	467.686	426.058	1.151	1.156
8	387.000	386.005	427.165	1.149	0.906
9	473.000	395.605	496.064	4.875	0.926
10	513.000	501.877	505.969	5.151	1.012
11	582.000	584.987	504.762	4.802	1.156
12	474.000	457.310	520.759	5.417	0.906
13	544.000	482.284	576.501	8.185	0.926
14	582.000	583.257	576.937	7.759	1.012
15	681.000	667.038	588.244	7.954	1.156
16	557.000	532.945	611.485	8.795	0.906
17	628.000	566.307	667.808	11.409	0.926
18	707.000	675.634	695.323	12.295	1.012
19	773.000	803.913	675.533	10.530	1.156
20	592.000	612.028	659.237	9.055	0.906
21	627.000	610.530	675.467	9.449	0.926
22	725.000	683.383	710.962	10.882	1.012
23	854.000	821.995	735.654	11.641	1.156
24	661.000	666.496	732.739	10.841	0.906
25	688.642	636.649	740.635		
26	763.262	696.722	829.801		
27	884.774	800.533	969.016		
28	703.143	599.650	806.635		
29	728.801	605.231	852.372		
30	807.133	663.004	951.263		
		L_Inferior	L_Superior		

Bien ahora se analizan las gráficas del error, las figuras 4.7 y 4.8 nos indican que el error no está correlacionado es decir el método de Holt – Winters es seguro para el pronóstico en este ejemplo. Debido a que las funciones de autocorrelación y

autocorrelación parcial están dentro de los límites permitidos. Estas funciones se analizarán en el capítulo 5 con detenimiento.

Además se puede observar que la media del error es cero, es decir el pronóstico no está sesgado.

En la figura 4.9 se puede observar que la distribución de probabilidad del error se asemeja a una distribución normal, razón por la cual se puede aceptar los intervalos de predicción mostrados en la figura 4.6 y tabla 4.10.

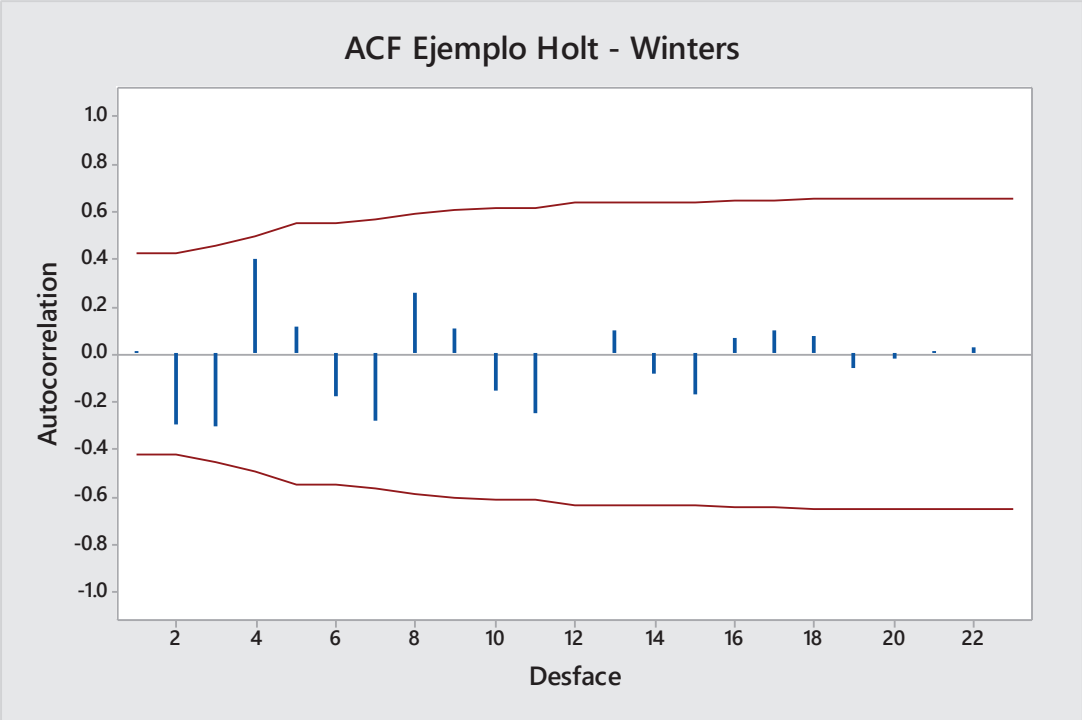


Figura 4.7 Gráfico de Autocorrelación del Error del Ejemplo.

(Programa Minitab ver 17)

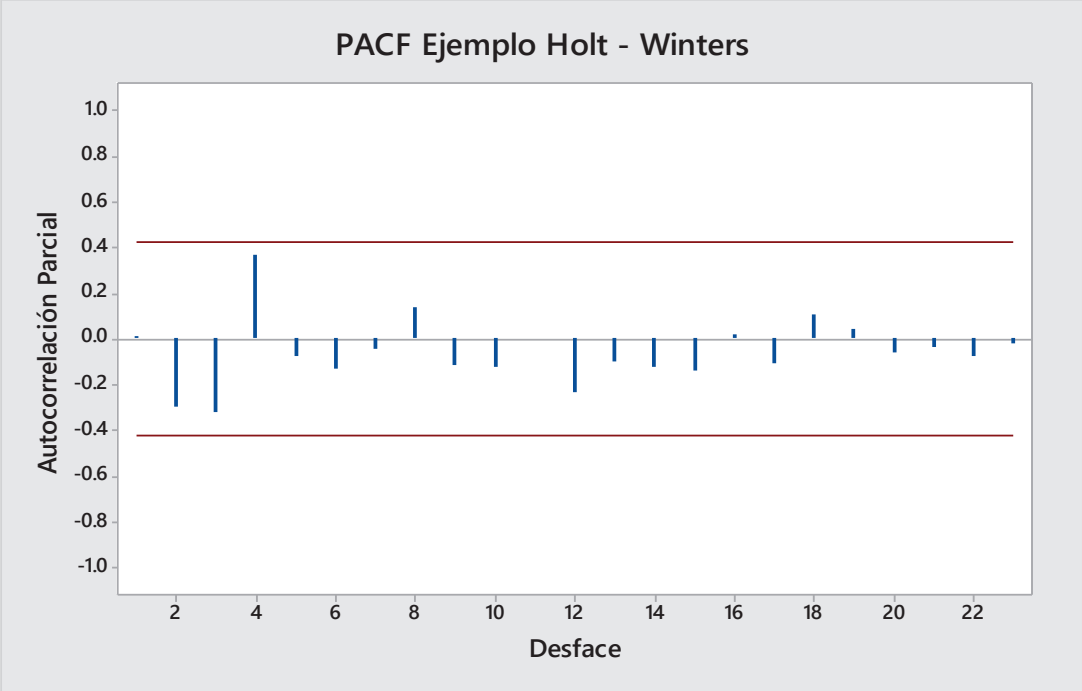


Figura 4.8 Gráfico de Autocorrelación Parcial del Error del Ejemplo.

(Programa Minitab ver 17)

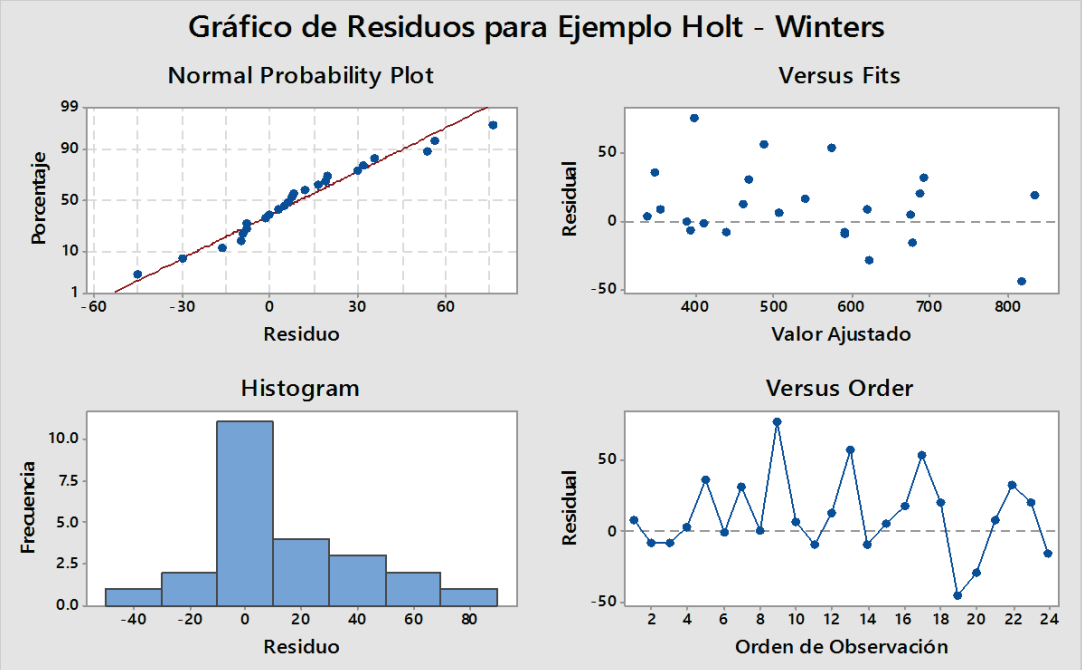


Figura 4.9 Distribución de Probabilidad del Error del Ejemplo

(Programa Minitab ver 17)

4.4 PRONOSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRAFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, CON EL METODO DE HOLT – WINTERS.

En la tabla 4.11 se muestra la información del consumo de placas digitales formato 510x400x0.15 (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo GTO_52, en la ciudad de Quito desde el año 2009 hasta el año 2015.

La recolección de datos se realizó mediante visitas personales (observación) a los clientes que utilizan las placas digitales, para garantizar la confiablidad de la información.

En esta sección se utiliza el paquete estadístico Minitab ver 17, que ayuda a realizar el pronóstico mediante el método de Holt – Winters para luego calcular el error de pronóstico MAPE.

En el Anexo B se presenta la información recolectada de cada cliente en el mercado gráfico quiteño. La Tabla 4.11 muestra los datos totalizados para el formato GTO_52.

Tabla 4.11 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	16600	16750	15300	17400	32390	45300	33700
2	16600	18000	18750	20350	30400	39672	30660
3	26850	26500	17750	24950	36300	34000	37900
4	16250	22800	14400	23800	37200	36862	35885
5	18150	23000	16300	37600	37956	42900	36400
6	18550	22150	19100	25323	32044	39233	36190
7	22200	20650	16100	29050	37700	38025	33900
8	21250	22265	16400	35600	34526	36300	32400
9	19300	25230	17700	30350	38880	37680	38300
10	21400	23100	18200	37100	41982	41911	36000
11	19950	18700	15150	42260	47700	47457	39300
12	28850	22800	19000	40900	48464	47563	39300

Los datos desde el período 1 hasta el período 14 se utilizarán para la inicialización del método de Holt - Winters, a partir del período 15 hasta el período 78 se utilizará como datos de prueba y finalmente desde el período 79 hasta el 84 no se utilizarán con el método para poder comparar con el pronóstico obtenido. Se someterá a prueba el método de Holt – Winters prediciendo los valores a partir del período 79 hasta el período 84, para luego compararlos con los valores reales y calcular el error de pronóstico MAPE.

Se seguirá el procedimiento sugerido anteriormente para pronosticar los datos:

4.4.1 GRÁFICO DE LOS DATOS

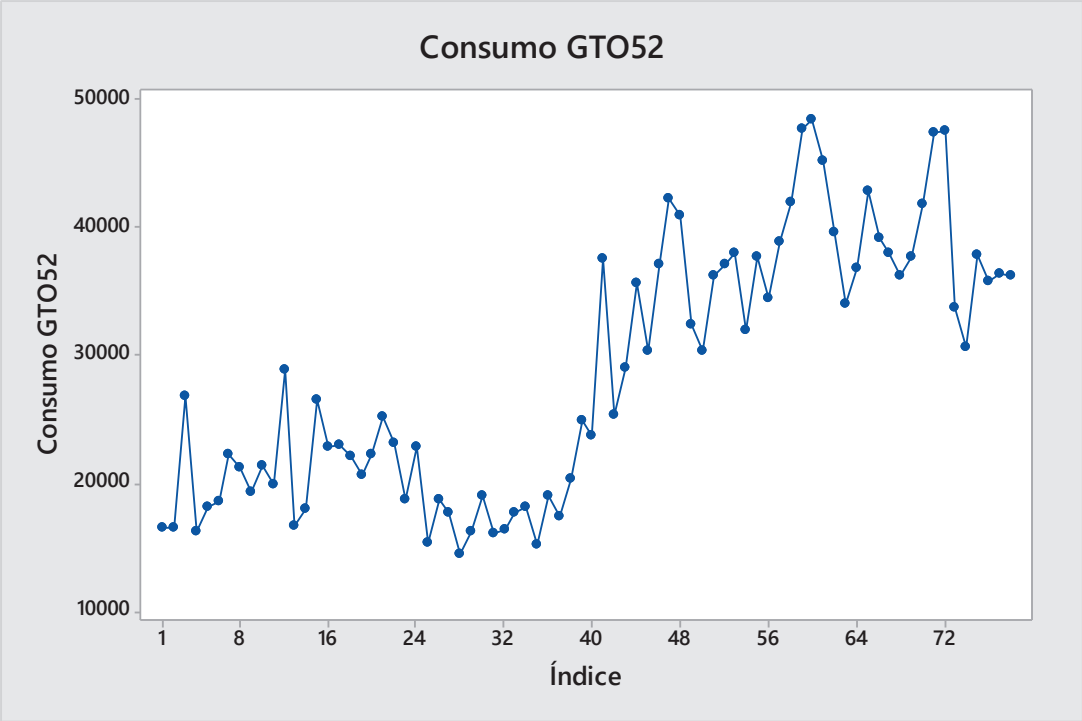


Figura 4.10 Consumo de placas formato 510x400x0.15 (GTO_52)

(Programa Minitab ver 17)

Se puede observar que los datos tienen componentes de tendencia y estacionalidad. Por esta razón se aplicará el método de Holt – Winters ya que este método de suavizamiento exponencial toma en cuenta las dos componentes.

4.4.2 EXAMINE POTENCIALES COMPORTAMIENTOS INUSUALES

Se puede observar que a partir del período 24 hasta período 40 existe una pendiente negativa en la tendencia, pero no hay presencia de discontinuidades (outliers).

4.4.3 DESCRIBIR LA FORMA DE LA VARIACIÓN EXPONENCIAL

En la Industria Gráfica Quiteña se conoce de antemano que los meses finales del año son de intensa actividad, esto se debe a la cantidad de impresos que se elaboran para la época navideña (como son agendas, calendarios, informes de fin de año, etc.).
Se pueden ver claramente picos en los períodos 48, 60 y 72 que corrobora la estacionalidad a fin de cada año.
Además comparando la figura 4.10 con la figura 4.1 se escoge el método de Holt – Winters multiplicativo.

4.4.4 CALCULO DE VALORES INICIALES

Como se va a utilizar el paquete Minitab donde se deben introducir los parámetros de suavizamiento exponencial iniciales, seguiremos la recomendación de Makridakis mencionada anteriormente, es decir se escogen los valores de $\alpha = \beta = \gamma = 0.2$, para luego probar con varios valores de estos parámetros.
En la figura 4.11 y tabla 4.12 se muestran los resultados obtenidos por el programa Minitab ver 17 escogiendo los parámetros arriba mencionados y el método de Holt – Winters Multiplicativo.

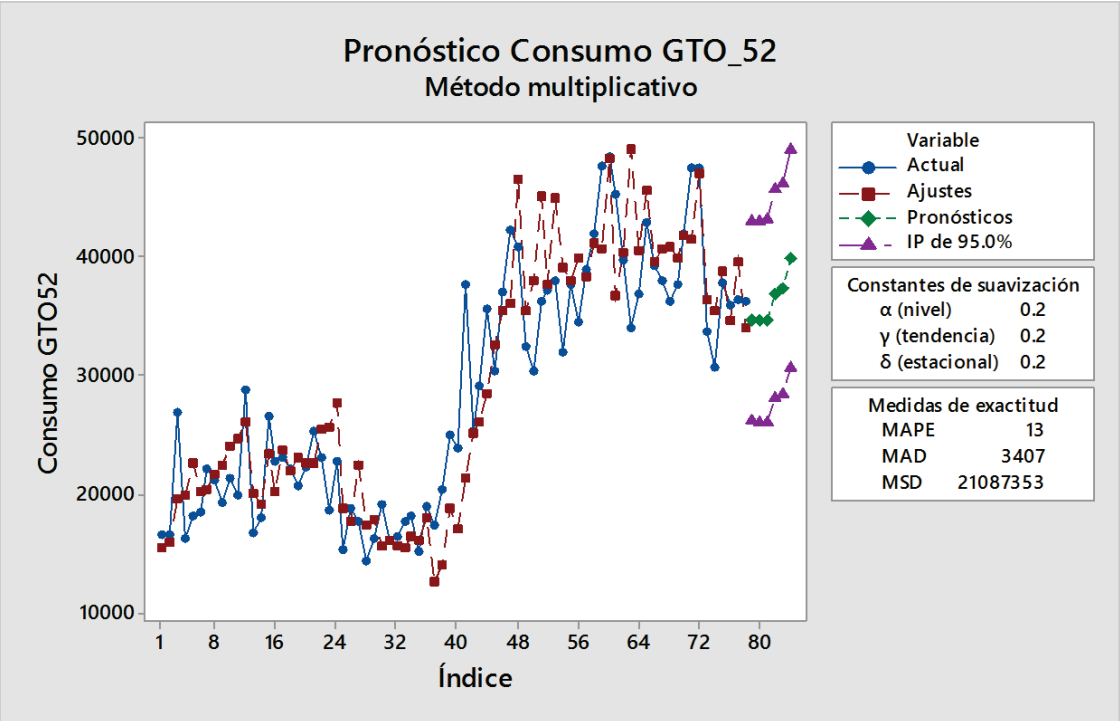


Figura 4.11 Pronóstico de placas formato 510x400x0.15 (GTO_52)

(Programa Minitab ver 17)

4.4.5 CHEQUEO DE ERRORES DE PRONÓSTICO

El programa Minitab Ver 17 nos da información acerca de la distribución de la probabilidad del error mediante dos gráficos: Histograma y la prueba de normalidad. Se muestran en las Figuras 4.12 y 4.13.

Tabla 4.12 Pronóstico de Placas Formato 510x400x0.15 desde el 2009 hasta 2015.

Mes	Consumo GTO52	ACF1	ESTADT1	LBQ1	PACF2	ESTADT2	DSUAV1	NIVEL1	LTEND1	ESTAC1	AJUSTES1	RESID1	PRONOST1	SUPE1	INFERIOR1
1.00	16600.00	0.86	7.58	59.70	0.86	7.58	15008.05	17767.77	592.13	0.89	15487.07	1112.93	34663.27	43010.52	26316.03
2.00	16600.00	0.79	4.45	111.26	0.21	1.87	15471.22	18500.75	620.30	0.88	15986.81	613.19	34583.97	43061.98	26105.95
3.00	26850.00	0.78	3.56	161.68	0.24	2.13	18945.25	20540.84	904.26	1.08	19580.46	7269.54	34681.68	43305.51	26057.85
4.00	16250.00	0.74	2.96	208.37	0.05	0.44	19092.54	20652.62	745.76	0.90	19933.04	-3683.04	36952.83	45736.77	28168.89
5.00	18150.00	0.75	2.68	255.85	0.19	1.63	21742.02	20566.82	579.45	1.02	22527.12	-4377.12	37355.93	46313.49	28398.36
6.00	18550.00	0.71	2.34	299.34	-0.06	-0.49	19616.20	20806.81	511.56	0.94	20168.87	-1618.87	39896.08	49040.03	30752.14
7.00	22200.00	0.66	2.05	338.12	-0.04	-0.32	19938.13	21688.14	585.51	0.97	20428.33	1771.67			
8.00	21250.00	0.59	1.74	369.24	-0.23	-2.04	21089.33	22189.60	568.70	0.97	21658.67	-408.67			
9.00	19300.00	0.56	1.60	398.04	0.05	0.44	21868.87	22123.25	441.69	0.96	22429.35	-3129.35			
10.00	21400.00	0.53	1.46	424.03	-0.08	-0.74	23590.17	22065.81	341.86	1.05	24061.15	-2661.15			
11.00	19950.00	0.51	1.37	448.71	0.11	1.02	24321.00	21546.16	169.56	1.07	24697.80	-4747.80			
12.00	28850.00	0.52	1.35	474.20	0.11	0.94	25862.85	22179.52	262.32	1.22	26066.39	2783.61			
13.00	16750.00	0.44	1.11	492.53	-0.18	-1.59	19832.66	21699.89	113.93	0.87	20067.23	-3317.23			
14.00	18000.00	0.39	0.99	507.65	-0.02	-0.20	19010.17	21560.42	63.25	0.87	19109.98	-1109.98			
15.00	26500.00	0.33	0.83	518.64	-0.20	-1.76	23299.30	22203.39	179.19	1.10	23367.65	3132.35			
16.00	22800.00	0.29	0.70	526.84	-0.06	-0.57	20004.32	22967.34	296.15	0.92	20165.77	2634.23			
17.00	23000.00	0.28	0.69	535.12	0.06	0.50	23396.76	23126.36	268.72	1.01	23698.44	-698.44			
18.00	22150.00	0.25	0.62	541.84	0.04	0.37	21769.55	23422.17	274.14	0.94	22022.50	127.50			
19.00	20650.00	0.21	0.51	546.49	-0.02	-0.14	22750.43	23209.00	176.68	0.96	23016.71	-2366.71			
20.00	22265.00	0.15	0.37	549.02	0.00	0.01	22499.80	23301.90	159.92	0.97	22671.08	-406.08			
21.00	25230.00	0.11	0.27	550.40	-0.08	-0.68	22437.72	24009.80	269.52	0.98	22591.71	2638.29			
22.00	23100.00	0.06	0.16	550.87	-0.09	-0.81	25138.52	23836.01	180.86	1.03	25420.70	-2320.70			
23.00	18700.00	0.06	0.14	551.26	0.04	0.32	25431.75	22718.83	-78.75	1.02	25624.71	-6924.71			
24.00	22800.00	0.06	0.14	551.68	0.04	0.38	27726.67	21848.46	-237.08	1.19	27630.56	-4830.56			
25.00	15300.00	0.01	0.01	551.68	-0.02	-0.18	19002.24	20807.44	-397.86	0.84	18796.05	-3496.05			
26.00	18750.00	-0.03	-0.07	551.77	0.00	-0.04	18056.95	20648.88	-350.00	0.88	17711.67	1038.33			
27.00	17750.00	-0.07	-0.18	552.44	-0.04	-0.36	22780.33	19456.94	-518.39	1.07	22394.19	-4644.19			
28.00	14400.00	-0.12	-0.28	554.12	-0.10	-0.85	17886.95	18283.63	-649.37	0.89	17410.39	-3010.39			
29.00	16300.00	-0.12	-0.28	555.91	-0.01	-0.13	18537.12	17322.82	-711.66	1.00	17878.75	-1578.75			
30.00	19100.00	-0.16	-0.39	559.45	-0.19	-1.69	16321.58	17343.26	-565.24	0.97	15651.06	3448.94			
31.00	16100.00	-0.17	-0.42	563.47	0.13	1.19	16562.89	16794.13	-562.02	0.96	16023.09	76.91			
32.00	16400.00	-0.24	-0.57	571.32	-0.23	-2.05	16234.13	16378.83	-532.67	0.97	15690.85	709.15			
33.00	17700.00	-0.29	-0.68	582.66	0.08	0.67	16059.37	16287.35	-444.44	1.00	15537.09	2162.91			
34.00	18200.00	-0.30	-0.71	595.64	-0.03	-0.26	16799.31	16203.40	-372.34	1.05	16340.90	1859.10			
35.00	15150.00	-0.33	-0.77	611.34	-0.03	-0.22	16497.95	15640.75	-410.40	1.01	16118.84	-968.84			
36.00	19000.00	-0.33	-0.76	627.18	0.01	0.09	18535.10	15390.89	-378.29	1.19	18048.76	951.24			
37.00	17400.00	-0.36	-0.83	646.64	-0.01	-0.11	12972.15	16138.95	-153.02	0.89	12653.31	4746.69			
38.00	20350.00	-0.36	-0.83	667.39	0.03	0.30	14135.42	17435.61	136.91	0.93	14001.39	6348.61			
39.00	24950.00	-0.39	-0.88	691.35	0.06	0.51	18569.50	18743.32	371.07	1.12	18715.32	6234.68			
40.00	23800.00	-0.39	-0.89	716.91	0.05	0.42	16737.14	20622.07	672.61	0.95	17068.50	6731.50			
41.00	37600.00	-0.37	-0.83	740.15	0.05	0.43	20607.28	24561.14	1325.90	1.11	21279.41	16320.59			
42.00	25323.00	-0.40	-0.87	767.30	-0.06	-0.54	23923.03	25909.33	1330.36	0.97	25214.48	108.52			
43.00	29050.00	-0.38	-0.83	793.00	-0.02	-0.18	24762.51	27870.82	1456.59	0.97	26033.98	3016.02			
44.00	35600.00	-0.36	-0.78	816.77	0.21	1.88	27134.54	30775.12	1746.13	1.01	28552.65	7047.35			
45.00	30350.00	-0.35	-0.75	840.10	-0.02	-0.15	30828.76	32076.44	1657.17	0.99	32577.94	-2227.94			
46.00	37100.00	-0.36	-0.76	865.05	-0.02	-0.13	33673.55	34054.96	1721.44	1.06	35413.23	1686.77			
47.00	42260.00	-0.34	-0.71	887.96	0.07	0.59	34336.50	37003.82	1966.92	1.04	36072.17	6187.83			
48.00	40900.00	-0.30	-0.63	906.36	0.01	0.09	44217.37	38022.12	1777.20	1.17	46567.73	-5667.73			
49.00	32390.00	-0.29	-0.60	924.25	-0.01	-0.09	33836.04	39118.89	1641.11	0.88	35417.58	-3027.58			
50.00	30400.00	-0.28	-0.59	941.86	-0.14	-1.27	36541.59	39116.82	1312.48	0.90	38074.57	-7674.57			
51.00	36300.00	-0.29	-0.59	960.89	-0.06	-0.51	43742.56	38835.70	993.76	1.08	45210.25	-8910.25			
52.00	37200.00	-0.28	-0.57	979.35	-0.07	-0.61	36707.23	39734.98	974.86	0.94	37646.53	-446.53			
53.00	37956.00	-0.26	-0.54	996.83	-0.02	-0.20	43931.03	39434.00	719.69	1.08	45008.83	-7052.83			
54.00	32044.00	-0.25	-0.52	1013.56	-0.01	-0.11	38435.91	38698.17	428.59	0.95	39137.38	-7093.38			
55.00	37700.00	-0.25	-0.50	1030.05	-0.07	-0.59	37655.32	39050.23	413.28	0.97	38072.36	-372.36			
56.00	34526.00	-0.24	-0.49	1046.75	0.00	-0.02	39449.38	38406.14	201.81	0.99	39866.89	-5340.89			
57.00	38880.00	-0.24	-0.48	1063.51	-0.03	-0.27	38046.27	38735.91	227.40	0.99	38246.19	633.81			
58.00	41982.00	-0.24	-0.48	1081.02	-0.06	-0.49	40971.58	39108.89	256.52	1.06	41212.10	769.90			
59.00	47700.00	-0.21	-0.43	1095.83	0.11	0.97	40478.59	40709.52	525.34	1.06	40744.09	6955.91			
60.00	48464.00	-0.18	-0.37	1107.27	0.05	0.44	47674.53	41264.61	531.29	1.17	48289.75	174.25			
61.00	45300.00	-0.18	-0.36	1118.80	0.05	0.47	36210.57	43761.26	924.36	0.91	36676.79	8623.21			
62.00	39672.00	-0.16	-0.33	1129.25	-0.04	-0.32	39504.37	44537.89	894.81	0.90	40338.82	-666.82			
63.00	34000.00	-0.16	-0.32	1139.69	0.03	0.25	48169.73	42633.46	334.97	1.02	49137.51	-15137.51			
64.00	36862.00	-0.16	-0.32	1150.85	-0.08	-0.69	40220.19	42189.50	179.18	0.93	40536.20	-3674.20			
65.00	42900.00	-0.14	-0.28	1160.56	-0.09	-0.76	45437.44	41861.63	77.77	1.07	45630.42	-2730.42			
66.00	39233.00	-0.13	-0.27	1169.87	0.00	0.02	39574.37	41851.62	60.22	0.94	39647.89	-414.89			
67.00	38025.00	-0.13	-0.25	1179.10	-0.05	-0.48	40659.94	41357.37	-50.68	0.96	40718.44	-2693.44			
68.00	36300.00	-0.11	-0.22	1186.77	0.05	0.48	40859.89	40393.74	-233.27	0.97	40809.82	-4509.82			
69.00	37680.00	-0.10	-0.20	1193.79	0.02	0.20	40120.99	39715.61	-322.24	0.98	39889.30	-2209.30			
70.00	41911.00	-0.10	-0.20	1201.55	-0.06	-0.51	42132.91	39415.98	-317.72	1.06	41791.06	119.94			
71.00	47457.00	-0.08	-0.16	1207.20	0.03	0.26	41874.01	40212.86	-94.80	1.09	41536.48	5920.52			
72.00	47563.00	-0.05	-0.09	1209.37	-0.04	-0.37	47120.07	40212.62	-75.89	1.17	47008.98	554.02			
73.00	33700.00	-0.04	-0.08	1211.61	-0.05	-0.42	36555.25	39523.72	-198.49	0.90	36486.27	-2786.27			
74.00	30660.00	-0.04	-0.08	1214.10	-0.06	-0.55	35584.37	38271.03	-409.33	0.88	35405.66	-4745.66			
75.00	37900.00	-0.03	-0.05	1215.44	0.02	0.14	39217.67	37686.39	-444.39	1.02	38798.21	-898.21			
76.00	35885.00	-0.02	-0.05	1217.25	0.04	0.33	35028.03	37515.28	-389.74	0.93	34614.98	1270.02			
77.00	36400.00	-0.01	-0.02	1218.13	0.06	0.49	40011.88	36526.19	-509.61	1.05	39596.20	-3196.20			
78.00	36190.00						34472.52	36482.47	-416.43	0.95	33991.57	2198.43			

(Programa Minitab ver 17)

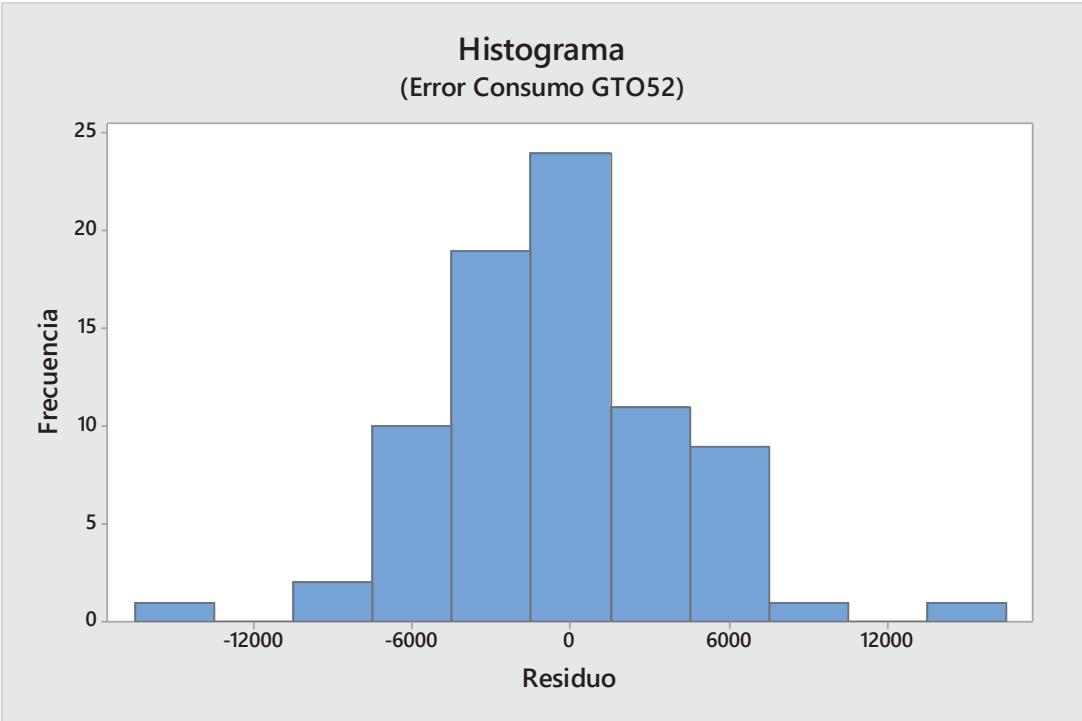


Figura 4.12 Histograma del Error de Pronóstico del Formato 510x400x0.15.

(Programa Minitab ver 17)

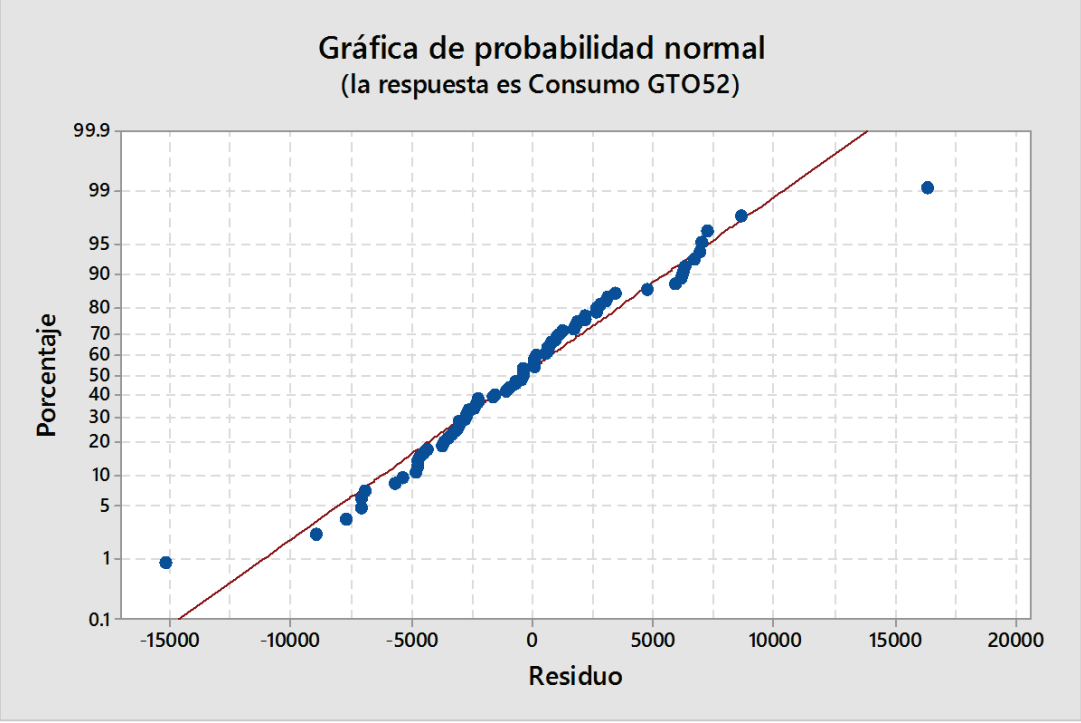


Figura 4.13 Prueba de Normalidad Error de Pronóstico del Formato 510x400x0.15

(Programa Minitab ver 17)

En las figuras 4.12 y 4.13 se puede observar que el error se asemeja mucho a una curva normal. Como se explicó anteriormente esto no es esencial para el pronóstico, pero nos permite calcular los intervalos de predicción de una manera más fácil.

Otro gráfico importante que nos permite realizar el Minitab es el gráfico de la función de autocorrelación ACF, en el próximo capítulo se estudiará en detalle esta función.

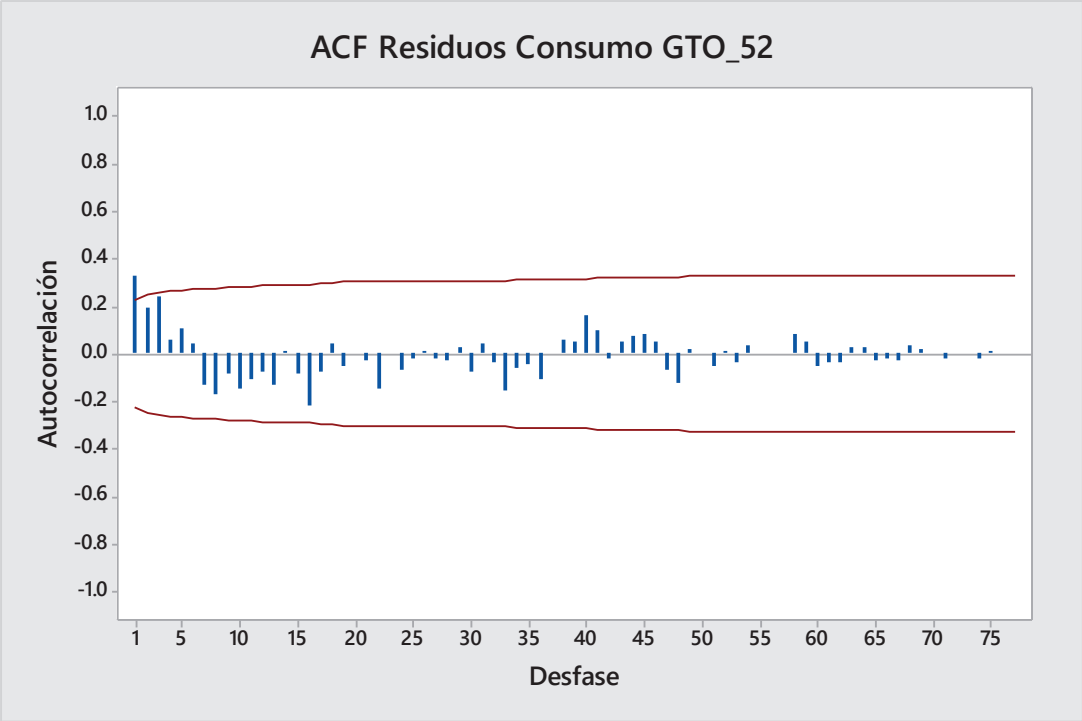


Figura 4.14 Función de Autocorrelación del Error de Pronóstico de Placas Digitales.
(Programa Minitab Ver 17)

La función de autocorrelación es muy útil para determinar si existe un patrón remanente en el error (o residuo) después de que el modelo de pronóstico ha sido aplicado. No es una medida de precisión per se, pero si puede sugerir que el método de pronóstico se podría mejorar. (Makridakis S., 1998).

En la figura 4.14 se puede observar que existe un pico en el primer período que sale del límite de significancia, pero el resto de residuos nos indican que el modelo ha capturado los patrones de datos bastante bien. Sin embargo hay autocorrelación.

Existe una prueba la de Durbin – Watson que permite saber si la autocorrelación en el período uno es significativa.

El coeficiente de Durbin – Watson está definido por:

$$DW = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n}(e_t)^2}$$

En este caso se calcula este coeficiente para saber si se puede mejorar el modelo de pronóstico, es decir si la autocorrelación en el período uno es significativa.

$$DW = \frac{2173522317}{1644813543} = 1.321$$

Para poder saber si es significativo este coeficiente se puede consultar el Anexo C tabla 4. Así para 80 observaciones y tres parámetros de nuestro modelo de Holt – Winters $((k' = 2)$ el límite inferior d_i y el límite superior d_S son 1.59 y 1.69 respectivamente.

Como el valor del índice DW en este caso está por debajo del límite inferior, se puede concluir que hay una autocorrelación significativa y positiva, es decir se puede mejorar el modelo. En la referencia (Makridakis S., 1998) pp: 303-304, se puede consultar la distribución de Durbin – Watson con mejor detalle.

Con el fin de mejorar el modelo de Holt – Winters se realizarón varias simulaciones con el programa Minitab incluyendo varios parámetros para iniciar el método y varias transformaciones para ver si mejora la autocorrelación del error.

A continuación se presenta un resumen de estas simulaciones:

Tabla 4.13 Método de Holt – Winters Multiplicativo (Original)

Método de Holt - Winters Multiplicativo (Original)					
			Alfa	Beta	Gamma
			0.2	0.2	0.2
	Real	Pronóstico		**MAPE**	**MSE**
	33900	34663		2.25	582626.89
	32400	34584		6.74	4769856.00
	38300	34682		9.45	13092094.89
	36000	36953		2.65	907827.84
	39300	37356		4.95	3779524.81
	39300	39896		1.52	355335.21
				4.59	3914544

En la figura 4.14 se muestra la función de autocorrelación correspondiente a los datos de la tabla 4.13, donde se muestra el pronóstico seis períodos adelante y los errores MAPE y MSE.

A continuación se muestra la simulación # 1:

Tabla 4.14 Método de Holt – Winters Multiplicativo (Simulación # 1)

Método de Holt - Winters Multiplicativo					
			Alfa	Beta	Gamma
			0.2	0.1	0.2
	Real	Pronóstico		**MAPE**	**MSE**
	33900	35994		6.18	4382742.25
	32400	36116		11.47	13806426.49
	38300	36337		5.13	3854939.56
	36000	38970		8.25	8820306.01
	39300	39658		0.91	128235.61
	39300	42594		8.38	10851094.81
				6.72	6973957

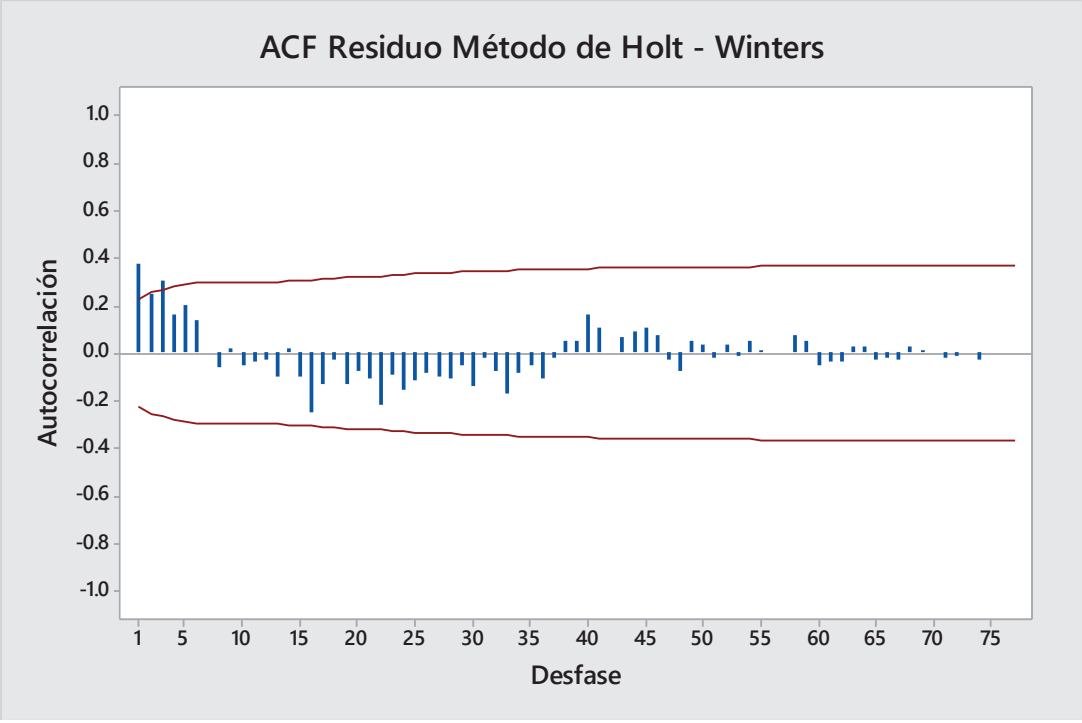


Figura 4.15 ACF del error Simulación # 1

Simulación # 2:

Tabla 4.15 Método de Holt – Winters Multiplicativo (Simulación # 2)

Método de Holt - Winters Multiplicativo					
			Alfa	Beta	Gamma
			0.3	0.1	0.3
	Real	Pronóstico		**MAPE**	**MSE**
	33900	35553		4.88	2732078.41
	32400	35527		9.65	9779379.84
	38300	36012		5.97	5236316.89
	36000	38686		7.46	7215670.44
	39300	39450		0.38	22440.04
	39300	41714		6.14	5827396.00
				5.75	5135547

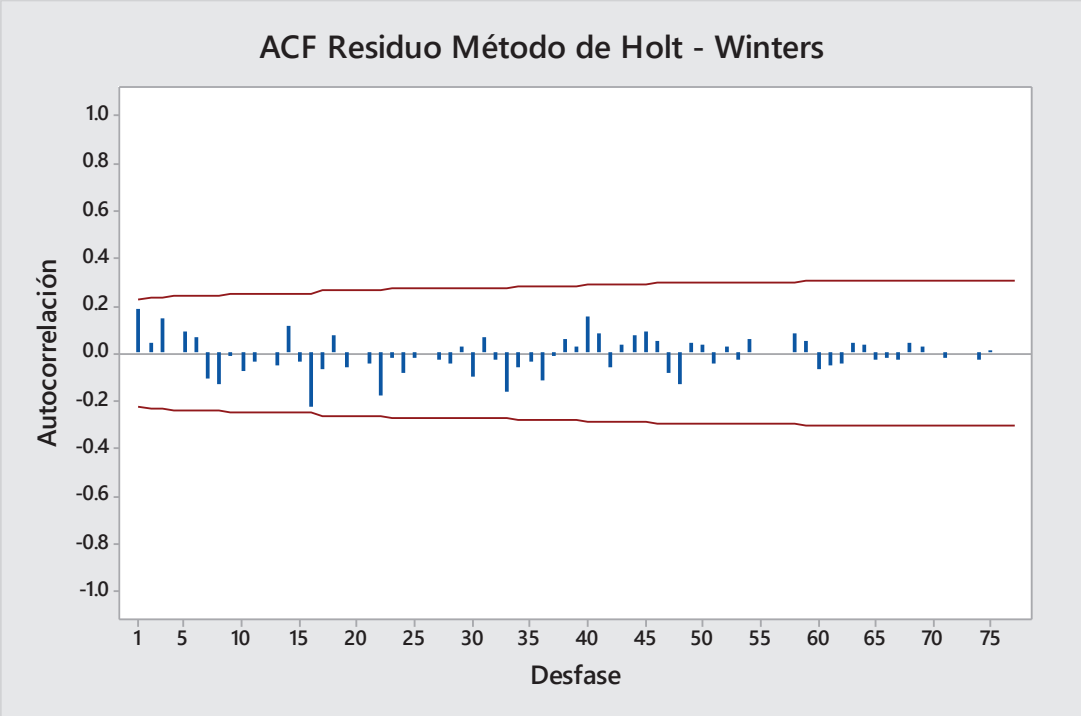


Figura 4.16 ACF del error Simulación # 2

Simulación # 3:

Tabla 4.16 Método de Holt – Winters Multiplicativo (Simulación # 3)

Método de Holt - Winters Multiplicativo					
			Alfa	Beta	Gamma
			0.4	0.1	0.4
	Real	Pronóstico		**MAPE**	**MSE**
	33900	35409		4.45	2277081.00
	32400	35271		8.86	8239770.25
	38300	35940		6.16	5571488.16
	36000	38639		7.33	6965376.64
	39300	39381		0.21	6593.44
	39300	41061		4.48	3101121.00
				5.25	4360238

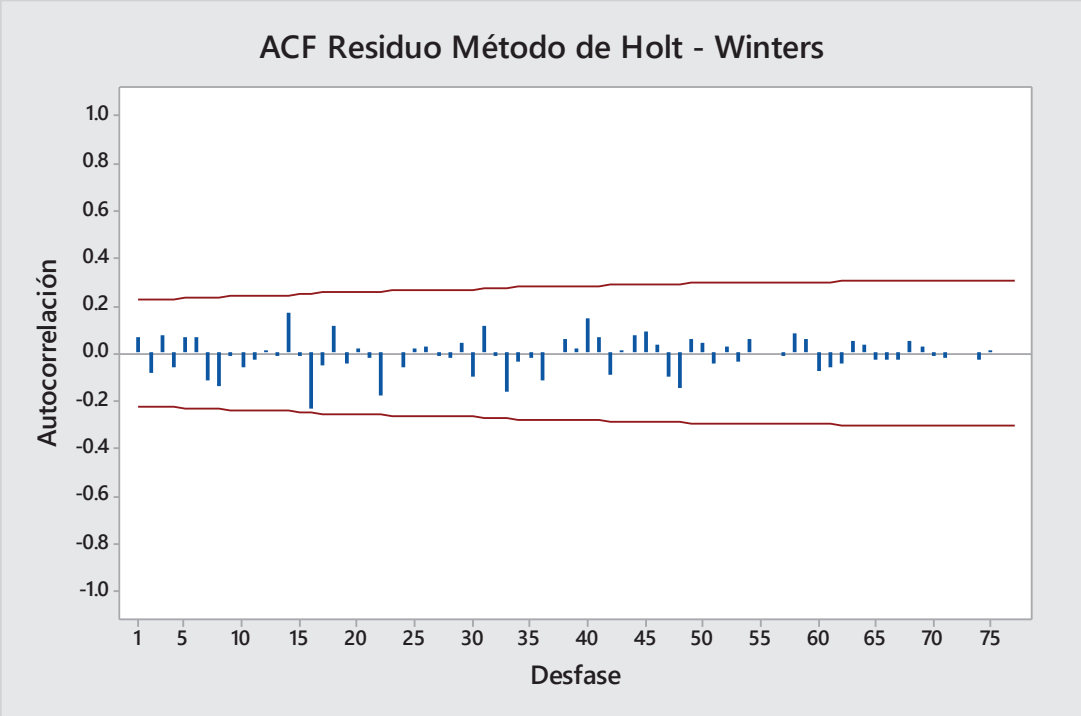


Figura 4.17 ACF del error Simulación # 3

Simulación # 4:

Tabla 4.17 Método de Holt – Winters Multiplicativo (Simulación # 4)

Método de Holt - Winters Multiplicativo					
			Alfa	Beta	Gamma
			0.4	0.2	0.4
	Real	Pronóstico		**MAPE**	**MSE**
	33900	34825		2.73	855269.73
	32400	34427		6.26	4108352.29
	38300	34863		8.97	11814180.54
	36000	37192		3.31	1419691.34
	39300	37599		4.33	2894789.23
	39300	38832		1.19	219046.45
				4.46	3551888

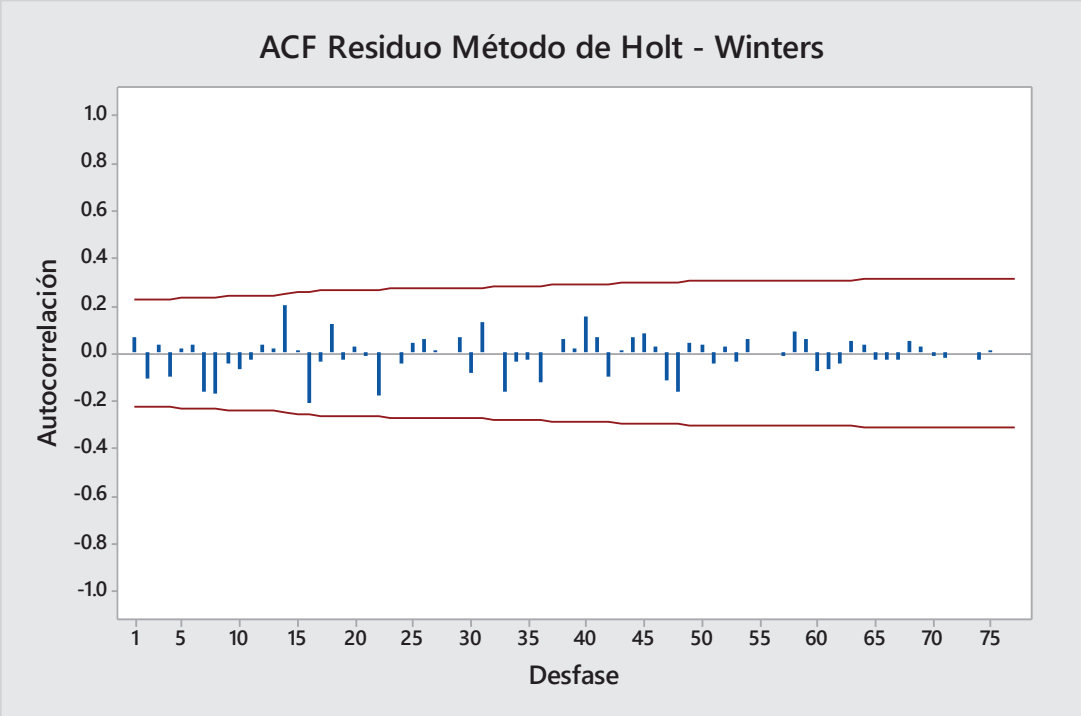


Figura 4.18 ACF del error Simulación # 4

Después de observar los resultados de la simulación # 4, la función de autocorrelación (ACF) del modelo indica que el modelo con estos parámetros ha capturado muy bien la información, no existe información remanente en los residuos que pueda ayudar a mejorar el pronóstico, se puede considerar ruido blanco.

Además los errores MAPE y MSE son los menores de todas las simulaciones, así que esta simulación será la representante del Método de Holt – Winters para la comparación final con el resto de métodos que se verán más adelante.

A continuación se presentan los gráficos obtenidos con esta simulación.

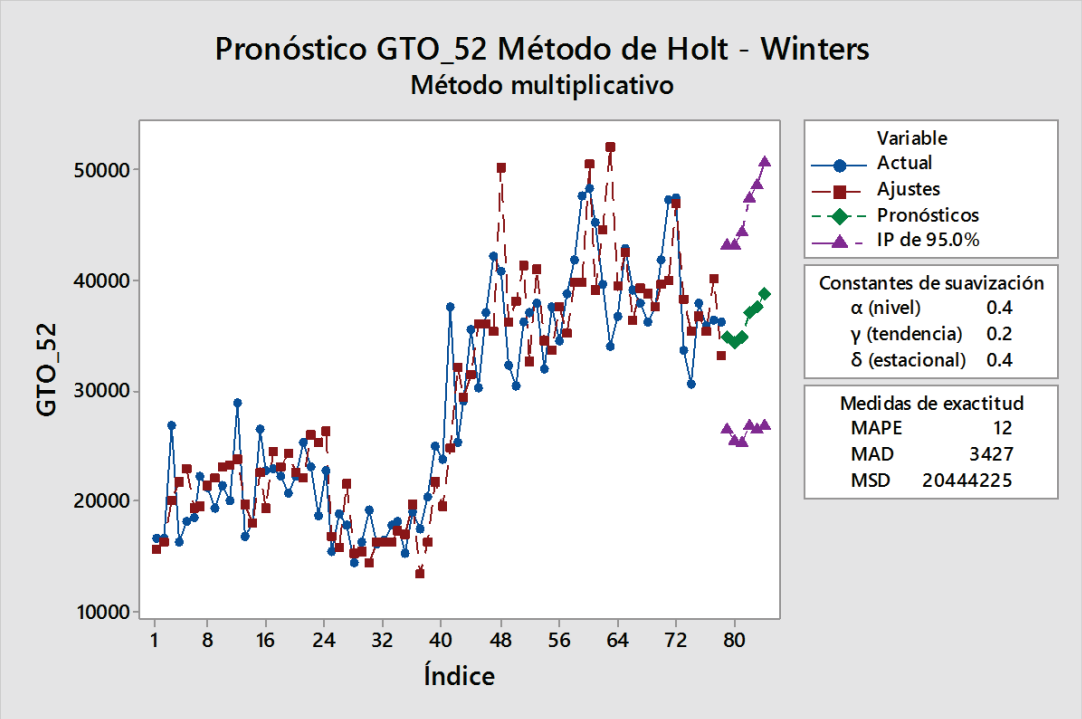


Figura 4.19 Pronóstico de placas formato 510x400x0.15 (GTO_52) Final.

(Programa Minitab ver 17)

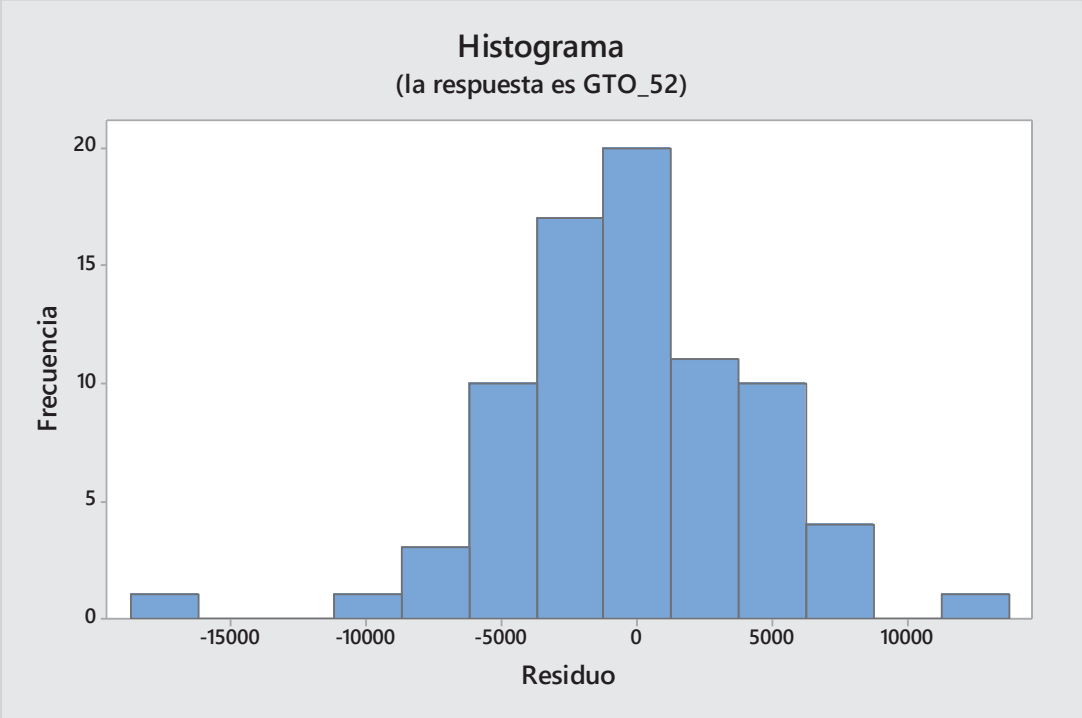


Figura 4.20 Histograma del Error de Pronóstico del Formato 510x400x0.15 Final

(Programa Minitab ver 17)

4.4.6 CÁLCULO DE PRONÓSTICOS A FUTURO

Los pronósticos seis períodos adelante son los siguientes:

Tabla 4.18 Pronóstico a Futuro Método de Holt – Winters Multiplicativo Final

Método de Holt - Winters Multiplicativo					
			Alfa	Beta	Gamma
			0.4	0.2	0.4
L_Inferior	Real	Pronóstico	L_Superior	**MAPE**	**MSE**
26429	33900	34825	43221	2.73	855269.73
25488	32400	34427	43365	6.26	4108352.29
25281	38300	34863	44444	8.97	11814180.54
26885	36000	37192	47498	3.31	1419691.34
26502	39300	37599	48695	4.33	2894789.23
26892	39300	38832	50772	1.19	219046.45
				4.46	3551888

Tabla 4.19 Datos Pronóstico a Futuro Método de Holt – Winters Multiplicativo Final

Mes	GTO_52	DSUAV1	NIVEL1	LTEND1	ESTAC1	AJUSTES1	RESID1	PRONOST1	SUPE1	INFERIOR1
1	16600	15008.05	18019.52	642.48	0.90	15487.07	1112.93	34824.81	43221.01	26428.60
2	16600	15690.42	18822.84	674.65	0.88	16249.86	350.14	34426.91	43365.35	25488.46
3	26850	19275.09	22186.51	1212.45	1.10	19965.95	6884.05	34862.82	44444.15	25281.50
4	16250	20622.17	21032.45	739.15	0.87	21749.13	-5499.13	37191.51	47497.54	26885.47
5	18150	22141.88	19959.19	376.67	1.00	22920.02	-4770.02	37598.59	48695.14	26502.04
6	18550	19036.66	19981.10	305.72	0.94	19395.92	-845.92	38831.98	50771.79	26892.16
7	22200	19146.89	21438.98	536.15	0.99	19439.84	2760.16			
8	21250	20847.05	21926.43	526.41	0.97	21368.39	-118.39			
9	19300	21609.50	21304.92	296.83	0.95	22128.30	-2828.30			
10	21400	22717.59	20988.76	174.23	1.05	23034.10	-1634.10			
11	19950	23133.87	19937.84	-70.80	1.06	23325.91	-3375.91			
12	28850	23932.31	21534.11	262.61	1.26	23847.32	5002.68			
13	16750	19358.93	20530.85	9.44	0.87	19595.02	-2845.02			
14	18000	17968.82	20550.77	11.53	0.88	17977.08	22.92			
15	26500	22574.89	21986.96	296.47	1.14	22587.56	3912.44			
16	22800	19057.00	23892.23	618.23	0.90	19313.96	3486.04			
17	23000	23782.12	23948.87	505.91	0.98	24397.50	-1397.50			
18	22150	22598.59	24062.25	427.40	0.93	23075.98	-925.98			
19	20650	23801.15	23044.41	138.35	0.95	24223.92	-3573.92			
20	22265	22378.29	23080.76	117.95	0.97	22512.64	-247.64			
21	25230	22011.78	24501.34	378.48	0.98	22124.27	3105.73			
22	23100	25668.13	23747.87	152.09	1.02	26064.63	-2964.63			
23	18700	25209.93	21386.17	-350.67	0.99	25371.38	-6671.38			
24	22800	26863.20	19881.86	-581.40	1.21	26422.73	-3622.73			
25	15300	17212.35	18649.44	-711.60	0.85	16709.02	-1409.02			
26	18750	16327.18	19329.45	-433.28	0.91	15704.19	3045.81			
27	17750	22058.78	17559.22	-700.67	1.09	21564.32	-3814.32			
28	14400	15834.19	16502.65	-771.85	0.89	15202.35	-802.35			
29	16300	16195.48	16082.14	-701.58	0.99	15438.00	862.00			
30	19100	15026.87	17404.86	-296.72	1.00	14371.33	4728.67			
31	16100	16568.17	17030.10	-312.33	0.95	16285.71	-185.71			
32	16400	16493.98	16803.89	-295.10	0.97	16191.48	208.52			
33	17700	16536.83	17099.61	-176.94	1.00	16246.41	1453.59			
34	18200	17401.60	17307.27	-100.02	1.03	17221.53	978.47			
35	15150	17077.05	16466.04	-248.26	0.96	16978.36	-1828.36			
36	19000	19962.95	15999.38	-291.94	1.20	19661.96	-661.96			
37	17400	13561.05	17635.89	93.75	0.90	13313.60	4086.40			
38	20350	16106.79	19550.56	457.93	0.96	16192.41	4157.59			
39	24950	21291.85	21168.91	690.02	1.12	21790.57	3159.43			
40	23800	18842.24	23810.90	1080.41	0.93	19456.42	4343.58			
41	37600	23674.01	30061.76	2114.50	1.10	24748.21	12851.79			
42	25323	30049.36	29439.14	1567.08	0.94	32162.99	-6839.99			
43	29050	27946.88	30844.19	1534.67	0.95	29434.52	-384.52			
44	35600	29965.04	34085.11	1875.92	1.00	31455.97	4144.03			
45	30350	34238.79	33662.13	1416.14	0.96	36123.16	-5773.16			
46	37100	34713.36	35437.56	1488.00	1.04	36173.72	926.28			
47	42260	34021.80	39762.76	2055.44	1.00	35450.35	6809.65			
48	40900	47812.37	38696.58	1431.12	1.14	50283.92	-9383.92			
49	32390	34951.10	38421.03	1089.78	0.88	36243.70	-3853.70			
50	30400	37050.70	36316.23	450.87	0.91	38101.61	-7701.61			
51	36300	40851.60	34968.23	91.09	1.09	41358.78	-5058.78			
52	37200	32655.82	36969.28	473.08	0.96	32740.89	4459.11			
53	37956	40549.89	36307.19	246.05	1.08	41068.79	-3112.79			
54	32044	34267.63	35512.43	37.89	0.93	34499.86	-2455.86			
55	37700	33606.07	37265.62	380.95	0.97	33641.92	4058.08			
56	34526	37290.82	36389.01	129.44	0.98	37672.02	-3146.02			
57	38880	35055.28	38054.77	436.70	0.99	35179.97	3700.03			
58	41982	39481.90	39280.68	594.54	1.05	39934.98	2047.02			
59	47700	39325.89	42983.20	1216.14	1.04	39921.11	7778.89			
60	48464	49183.14	43461.49	1068.57	1.13	50574.69	-2110.69			
61	45300	38208.59	47329.17	1628.39	0.91	39148.00	6152.00			
62	39672	43232.21	46747.16	1186.31	0.89	44719.64	-5047.64			
63	34000	50962.14	41235.26	-153.33	0.98	52255.42	-18255.42			
64	36862	39702.09	39963.35	-377.05	0.95	39554.45	-2692.45			
65	42900	43011.65	39695.63	-355.18	1.08	42605.85	294.15			
66	39233	36806.89	40529.13	-117.45	0.94	36477.56	2755.44			
67	38025	39412.70	39887.86	-222.21	0.96	39298.48	-1273.48			
68	36300	39087.18	38616.82	-431.98	0.96	38869.43	-2569.43			
69	37680	38102.56	38186.33	-431.68	0.99	37676.33	3.67			
70	41911	40096.00	38618.75	-258.86	1.06	39642.73	2268.27			
71	47457	40340.56	41188.51	306.87	1.09	40070.16	7386.84			
72	47563	46649.51	41695.25	346.84	1.14	46997.06	565.94			
73	33700	37956.55	40033.03	-54.97	0.88	38272.29	-4572.29			
74	30660	35530.21	37805.08	-489.57	0.86	35481.42	-4821.42			
75	37900	37196.95	37797.16	-393.24	0.99	36715.26	1184.74			
76	35885	35780.66	37605.30	-352.96	0.95	35408.40	476.60			
77	36400	40540.62	35857.20	-631.99	1.05	40160.10	-3760.10			
78	36190	33832.87	36477.27	-381.58	0.96	33236.56	2953.44			

5 METODOLOGIA DE BOX – JENKINS

5.1 INTRODUCCION

Los modelos ARIMA (del inglés Autoregressive Integrated Moving Average) autoregresivos integrados de medias móviles han sido estudiados extensamente. Fueron popularizados por George Box y Gwilym Jenkins en la década de los 70's. Los dos creadores en 1970 compilaron de una manera comprensiva información relevante para entender series de tiempo univariantes y modelos ARIMA.

El fundamento teórico descrito por Box and Jenkins (1970) y luego por Box, Jenkins y Reinsell (1994) es bastante sofisticado, pero es posible para no especialistas comprender la esencia de la metodología ARIMA. (Makridakis S., 1998).

Una serie de tiempo es una secuencia de observaciones tomadas secuencialmente en el tiempo. Muchos datos aparecen como series de tiempo: una secuencia mensual de la cantidad de mercancías embarcadas desde una fábrica, una serie semanal del número de accidentes de tránsito, observaciones cada hora hechas en la producción de un proceso químico, y así. Ejemplos de series de tiempo abundan en algunos campos como: económico, negocios, ingeniería, ciencias naturales (especialmente geofísica y meteorología) y ciencias sociales. (Box G., 2008 4th Edition).

Una característica intrínseca de las series de tiempo es que normalmente observaciones adyacentes *son dependientes*. El análisis de series de tiempo estudia las técnicas para analizar esta dependencia. Esto requiere el desarrollo de modelos estocásticos para series de tiempo y el uso de estos modelos en importantes áreas de aplicación. (Box G., 2008 4th Edition).

Cabe indicar que los métodos que se discutirán en este capítulo son apropiados para sistemas discretos (datos muestreados), donde las observaciones ocurren a intervalos de tiempo igualmente espaciados.

El área de aplicación en el presente trabajo será el pronóstico de valores futuros de una serie de tiempo a partir de valore pasados desde el año 2009 al 2015 de la demanda de placas digitales en el mercado gráfico quiteño.

5.1.1 PRONOSTICO DE SERIES DE TIEMPO

Se supone que observaciones están disponibles en intervalos de tiempo igualmente espaciados. Por ejemplo en un problema de predicción de ventas, las ventas Z_t en el mes actual t. y las ventas $Z_{t-1}, Z_{t-2}, Z_{t-3}$,… de meses anteriores deberán ser usadas para predecir las ventas para los tiempos de espera $l = 1,2,3, \dots 12$ meses hacia el futuro. Se denota $\hat{Z}_t(l)$ el pronóstico hecho al origen t de las ventas, Z_{t+l} para algún tiempo futuro $t + l$, esto es, el tiempo de espera l. La función $\hat{Z}_t(l)$ que proveerá el pronóstico a partir de origen t para valores futuros en base de valores pasados se denominará función de pronóstico al origen t. El objetivo será obtener la función de pronóstico tal que minimice el error entre el valor actual y el valor predicho ($Z_{t+l} - \hat{Z}_t(l)$, para cada tiempo de espera l. (Box G., 2008 4th Edition).

Además de obtener el mejor pronóstico es necesario especificar su precisión. La precisión del pronóstico puede expresarse por los ***límites de probabilidad*** a los dos lados de cada pronóstico. Estos límites pueden ser calculados para algún conjunto de probabilidades por ejemplo 50% o 95%. Así al valor pronosticado se deberá incluir estos límites con su respectiva probabilidad. (Box G., 2008 4th Edition).

Métodos de pronósticos serán desarrollados asumiendo que la serie de tiempo Z_t sigue un modelo estocástico de una forma conocida. Este tipo de modelos se denominan ARIMA autoregresivo integrado de medias móviles. (Box G., 2008 4th Edition).

Los modelos ARIMA son especialmente seguros para la predicción a corto plazo, ya que la mayoría de modelos ARIMA ponen mucho énfasis en el pasado reciente en lugar del pasado distante. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.1.2 MODELOS MATEMATICOS DINAMICOS ESTOCASTICOS Y DETERMINISTICOS

La idea de utilizar un modelo matemático para describir el comportamiento de un fenómeno físico está muy bien establecido. En particular, es a veces posible derivar

un modelo basado en leyes físicas que nos permiten calcular el valor de alguna cantidad dependiente del tiempo exactamente en algún instante del tiempo. Así se puede calcular la trayectoria de un misil lanzado en una dirección conocida y con una velocidad conocida. Si un cálculo exacto es posible, tal modelo es enteramente ***determinístico***. (Box G., 2008 4th Edition).

En muchos otros problemas se debe considerar fenómenos dependientes del tiempo como las ventas mensuales de un periódico, en el cual existen muchos factores desconocidos y por lo que no es posible escribir un modelo determinístico que permita un cálculo exacto del comportamiento futuro del fenómeno. Sin embargo es posible derivar un modelo que pueda ser usado para calcular ***la probabilidad*** de que un valor futuro se encuentre entre límites específicos. Tal modelo es llamado modelo probabilístico o ***modelo estocástico***. Los modelos de series de tiempo que se necesitan para alcanzar un pronóstico óptimo son en efecto modelos estocásticos. (Box G., 2008 4th Edition).

5.1.2.1 Modelos Estocásticos Estacionarios y No Estacionarios para Pronósticos

En los ***modelos estacionarios*** se asume que el proceso permanece en un *equilibrio estadístico* con propiedades probabilísticas que no cambian en el tiempo, en particular variando alrededor de una *media constante* y con una *varianza constante*. (Box G., 2008 4th Edition).

Sin embargo, los pronósticos han adquirido importancia en la industria, los negocios y la economía donde muchas series de tiempo son mejor representadas como no estacionarias y en forma particular las que no tienen una media constante al pasar el tiempo. No es ninguna sorpresa que métodos que utilizan suavizamiento exponencial, como los vistos en el capítulo anterior, sean apropiados para un tipo particular de procesos no estacionarios. Aunque esos métodos son muy limitados para manejar con eficiencia todas las series de tiempo.

Estos modelos estocásticos de pronóstico con suavizamiento exponencial que producen un error cuadrático medio mínimo son miembros de una clase de *procesos no estacionarios* llamados autoregresivos integrados de medias móviles (ARIMA). El enfoque será primero derivar un modelo estocástico para la serie de

tiempo en estudio, una vez obtenido el modelo para la serie inmediatamente seguirá un procedimiento óptimo de pronóstico. (Box G., 2008 4th Edition).

Dentro de los modelos estacionarios tenemos a los modelos Autoregresivos (AR), de medias móviles (MA) y Modelos autoregresivos de medias móviles (ARMA).

Muchas series encontradas en la industria o los negocios tienen un comportamiento **no estacionario** en particular no varían alrededor de una media fija. Tales series podrían sin embargo tener un comportamiento homogéneo a través del tiempo.

Así este comportamiento no estacionario homogéneo puede ser transformado a estacionario tomando las diferencias de orden d al proceso original, en la práctica d=0,1 o máximo 2 es necesaria, donde d=0 corresponde a un proceso estacionario. Los procesos ARIMA de orden (p, d, q) son capaces de describir estacionariedad y no estacionariedad. (Box G., 2008 4th Edition).

Normalmente se considera que los valores sucesivos de la serie de tiempo bajo consideración están disponibles para el análisis. Si es posible se recomienda *al menos entre 50 y 100 observaciones* que deberían usarse. (Box G., 2008 4th Edition). Otros analistas recomiendan una muestra mayor a 50 observaciones cuando se trabaja con datos estacionales o cíclicos. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La metodología de Box–Jenkins es un proceso iterativo que consiste de tres fases: identificación, estimación y diagnóstico - chequeo, para finalmente realizar el Pronóstico con el modelo apropiado, como se muestra en figura 5.1.

En la **etapa de identificación** se selecciona uno o más modelos tentativos analizando dos gráficos derivados de los datos disponibles. Estos gráficos son llamados función de autocorrelación (acf) estimada y función de autocorrelación parcial (pacf) estimada. Se elige el modelo comparando las funciones acf y pacf teóricas con las funciones estimadas acf y pacf calculadas de los datos.

En la **etapa de estimación** se obtienen los parámetros del modelo ARIMA tentativo elegido en la etapa de identificación.

En la **etapa de diagnóstico – chequeo** se realizan pruebas para ver si el modelo estimado es estadísticamente adecuado. Si el modelo no es satisfactorio se regresa a la etapa de identificación para tentativamente seleccionar otro modelo.

Finalmente un modelo ARIMA construido adecuadamente produce un pronóstico óptimo para una sola variable, es decir se obtiene la menor varianza en el error de pronóstico. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).
Box, Jenkins y Reinsel enfatizan en un importante principio, el de **parsimonia**. Acorde con este principio, se necesita encontrar el modelo adecuado más simple, es decir el que contenga el menor número de coeficientes y explique satisfactoriamente el comportamiento de los datos observados.

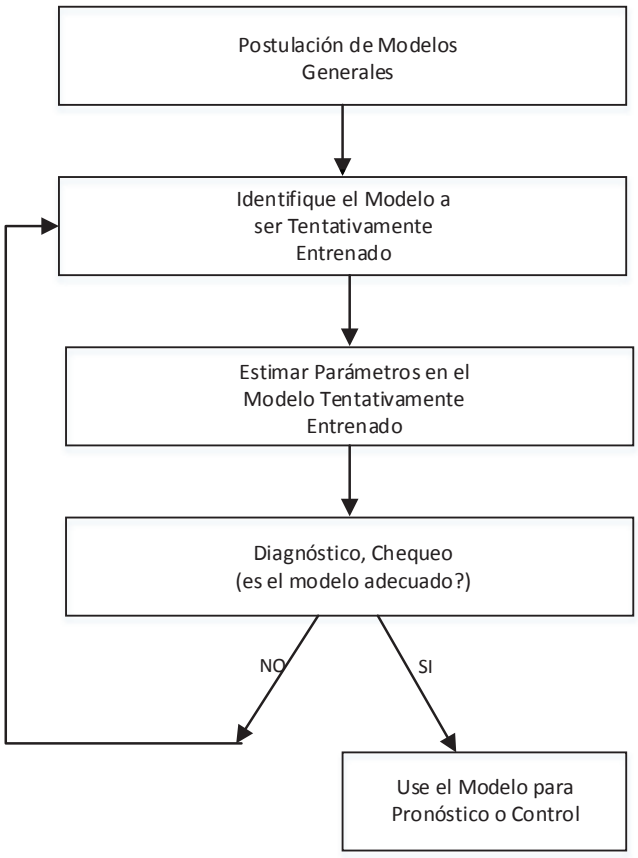


Figura 5.1 Etapas en un enfoque iterativo en la construcción de modelos

(Box G., 2008 4th Edition Pág.18)

Sin embargo seleccionar un modelo ARIMA apropiado no es tarea fácil, varios autores sugieren que construir un modelo ARIMA apropiado es un arte que requiere de buen juicio y mucha experiencia. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

En la presente investigación se analizarán varios ejemplos para desarrollar ese buen juicio y mejorar la experiencia en la modelación ARIMA.

5.1.3 MODELOS ARIMA COMPARADOS CON OTROS MODELOS

Los modelos ARIMA tienen tres ventajas sobre otros métodos tradicionales de series de tiempo. **Primero**, los conceptos asociados con los modelos ARIMA están derivados de una base sólida de la teoría de probabilidad clásica y matemáticas estadísticas.

Segundo, Los modelos ARIMA son una familia de modelos, no un simple modelo. Box y Jenkins han desarrollado una estrategia que guía al analista a escoger un modelo apropiado de una amplia familia de modelos.

Tercero, se puede demostrar que un modelo ARIMA apropiado produce un pronóstico univariante óptimo (menor varianza en el error de pronóstico). (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2 FUNDAMENTO TEÓRICO DE LA METODOLOGÍA DE BOX-JENKINS

5.2.1 CONCEPTOS FUNDAMENTALES

Se repasarán conceptos básicos que se utilizarán a lo largo de este capítulo, que servirán para el pronóstico de las placas digitales.

5.2.1.1 Correlación

El concepto de correlación se utiliza cuando se necesita examinar y medir el grado de relación entre dos variables.

La correlación mide la fuerza y la dirección de la relación lineal entre dos variables aleatorias. (Pankratz, Forecasting with Dynamic Regression Models, 1991)

5.2.1.1.1 Coeficiente de Correlación de la Población

El coeficiente de correlación de la población para dos variables aleatorias está dado por:

$$\rho = \frac{cov(X,Y)}{\sigma_x \sigma_y} \quad (5.1)$$

Donde $\sigma_x \sigma_y$ son las desviaciones estándar de la población de las variables aleatorias X e Y respectivamente y $cov(X,Y)$ la covarianza de la población de X e Y. Como se conoce σ_x es la raíz cuadrada de la varianza de la población ${\sigma_x}^2$, que se define como el valor esperado

$${\sigma_x}^2 = E(X_i - \mu_x)^2 \quad (5.2)$$

El símbolo μ_x es el valor esperado (media de la población) de la variable aleatoria X,

$$\mu_x = E(X_i) \quad (5.3)$$

Dividiendo la $cov(X,Y)$ para el producto $\sigma_x \sigma_y$ en (5.1) se estandariza la covarianza convenientemente así ρ puede tomar valores entre -1 y +1.

La covarianza de X e Y se define con el valor esperado,

$$cov(X,Y) = E[(X_i - \mu_x)(Y_i - \mu_y)] \quad (5.4)$$

Si valores altos de Y_i tienden a valores altos de X_i y valores bajos de Y_i tienden a valores bajos de X_i, entonces la $cov(X,Y)$ es positiva y X e Y son positivamente correlacionadas ($\rho > 0$). Pero si valores altos de Y_i tienden a valores bajos de X_i y valores bajos de Y_i tienden a valores altos de X_i, entonces la $cov(X,Y)$ es negativa y X e Y son negativamente correlacionadas ($\rho < 0$).

Se puede entender la covarianza de una manera más clara si se interpreta la definición (5.4). Si se supone que Y_i tiende a estar sobre μ_y cuando X_i está sobre μ_x o que Y_i tiende a estar bajo μ_y cuando X_i está bajo μ_x, por lo tanto el producto $(X_i - \mu_x)(Y_i - \mu_y)$ y la $cov(X,Y)$ tenderán a ser positivos. Alternativamente si se supone que Y_i tiende a estar bajo μ_y cuando X_i está sobre μ_x o que Y_i tiende a estar sobre μ_y cuando X_i está bajo μ_x, entonces el producto $(X_i - \mu_x)(Y_i - \mu_y)$ y la $cov(X,Y)$ tenderán a ser negativos. Sin embargo las cantidades $cov(X,Y)$ y ρ no

son observables. Se necesitan las estadísticas de una muestra para estimar estos valores. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

5.2.1.1.2 Coeficiente de Correlación Muestral

El coeficiente de correlación muestral provee un estimado de ρ, se calcula como

$$r = \frac{C(X,Y)}{S_x S_y} \quad (5.5)$$

Donde $S_x S_y$ son las desviaciones estándar muestrales de X_i , Y_i respectivamente y $C(X,Y)$, covarianza muestral entre X e Y. Además S_x es la raíz cuadrada de la *varianza muestral*, cuya fórmula es

$$S_x^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \quad (5.6)$$

S_y, se obtiene de una manera similar. La varianza muestral S_x^2 de una serie de tiempo sirve para estimar la varianza de la población σ_x^2. Como es usual la varianza mide la dispersión de las observaciones alrededor de la media. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La *covarianza muestral* se calcula como

$$c(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (5.7)$$

La covarianza y el coeficiente de correlación son estadísticos que miden el grado de relación lineal entre dos variables. (Makridakis S., 1998).

5.2.1.1.3 Autocovarianza y Autocorrelación

La autocovarianza y la autocorrelación son medidas comparables con la covarianza y la correlación y sirven para el mismo propósito pero para una sola serie de tiempo. Es decir su cálculo se puede realizar como si fueran dos series separadas, luego se aplica las fórmulas de la covarianza y la correlación teniendo en cuenta que se trata de la misma serie pero retrasada una o varios períodos. (Makridakis S., 1998). Así la autocovarianza para el rezago k se representa por c_k y la autocorrelación para el rezago k por r_k. Sus fórmulas son:

$$c_k = \frac{\sum_{t=1}^{n-k}(Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{n} \quad (5.8)$$

y

$$r_k = \frac{\sum_{t=1}^{n-k}(Y_t-\bar{Y})(Y_{t+k}-\bar{Y})}{\sum_{t=1}^{n}(Y_t-\bar{Y})^2} \quad (5.9)$$

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Las autocorrelaciones juntas para los rezagos 1,2,…, forman la ***función de autocorrelación ACF*** $\boldsymbol{\rho(k)}$ (del inglés ***Autocorrelation Function***), si esta función se grafica contra los rezagos de tiempo se obtiene un gráfico muy útil para la identificación de modelos ARIMA el mismo que se denomina ***correlograma***.

En la práctica para obtener un estimado útil de la función de autocorrelación se necesitan al menos 50 observaciones y la autocorrelación estimada r_k debería calcularse para $k = 0,1,2,\dots,K$ donde K no debe ser mayor a $\frac{n}{4}$, donde n es el número de observaciones. (Box G., 2008 4th Edition).

Teóricamente todos los coeficientes de autocorrelación para una serie de números aleatorios deben ser cero. Pero ya que se tienen muestras finitas, cada coeficiente de autocorrelación no será exactamente cero. Ha sido demostrado por varios autores (Anderson (1942), Barlett (1946), Quenouille (1949) y otros) que los coeficientes de autocorrelación del ruido blanco tienen una distribución de muestreo que puede ser aproximada por una curva normal de media cero y error estándar de $\frac{1}{\sqrt{n}}$ donde n es el número de observaciones.

Por ejemplo, 95% de todos los coeficientes de autocorrelación muestral deben caer en un rango especificado por la media más o menos 1.96 veces el error estándar[1]. Como la media es cero y el error estándar es $\frac{1}{\sqrt{n}}$ para ruido blanco, se espera que el 95% de los coeficientes de autocorrelación muestral estarán dentro del intervalo $\pm \frac{1.96}{\sqrt{n}}$. Si este no es el caso, la serie probablemente no es ruido blanco. Por esta razón es común graficar líneas a $\pm \frac{1.96}{\sqrt{n}}$ cuando graficamos la acf. Estos límites son conocidos como ***valores críticos***. (Makridakis S., 1998)

En resumen la distribución de muestreo y el error estándar permiten interpretar los resultados del análisis de autocorrelación así se puede distinguir que es un patrón, o que es aleatorio o ruido blanco, en los datos.

---

[1] El valor de 1.96 se encuentra buscando en el Anexo C tabla C1 correspondiente a la curva de distribución Normal

5.2.1.1.4 Pruebas de Portmanteau

En lugar de estudiar los valores de r_k uno a la vez, un enfoque alternativo es considerar todo un conjunto de valores a la vez, digamos los 15 primeros de ellos ($r_{1\ hasta}\ r_{15}$) todos al mismo tiempo y desarrollar una prueba que nos permita determinar si el conjunto de valores es significativamente diferente de cero. Estas pruebas se conocen como Pruebas de ***Portmanteau***. (Makridakis S., 1998)

Una prueba común de Portmanteau es la prueba de *Box – Pierce* basada en el estdístico Q de Box – Pierce:

$$Q = n\sum_{i=1}^{h} r_k^2 \qquad (5.10)$$

Donde h es el máximo rezago a ser considerado y n el número de observaciones de la serie. Usualmente se selecciona $h = 20$.

La prueba de Box – Pierce fue diseñada por Box y Pierce (1970) para probar los residuos de un modelo de Pronóstico. Si los residuos son ruido blanco, el estadístico Q tiene una distribución chi - cuadrada (χ^2) con ($h - m$) grados de libertad donde m es el número de parámetros del modelo que han sido ajustados con los datos. El estadístico Q puede ser comparado con la tabla chi-cuadrada (Anexo C Tabla C2) para evaluar si es significativo.

Una prueba Portmanteau alternativa es la prueba *Ljung – Box* en honor a Ljung y Box (1978). Ellos crearon un estadístico alternativo,

$$Q^* = n(n+2)\sum_{k=1}^{h} \frac{r_k^2}{(n-k)} \qquad (5.11)$$

Tiene una distribución más cercana a la distribución chi-cuadrada que el estadístico Q.

Si los datos son ruido blanco, el estadístico Q^* tiene exactamente la misma distribución que el estadístico Q de Box – Pierce denominada chi – cuadrada con ($h - m$) grados de libertad. El estadístico Q^* puede ser comparado con la tabla chi-cuadrada (Anexo C Tabla C2) para evaluar si es significativo.

Se puede concluir que los datos no son ruido blanco si el valor de Q o (Q^*) cae al extremo del 5% de la cola derecha de la distribución chi – cuadrada. (Esto es, el valor de Q o (Q^*) es mayor que el valor dado en la columna del Anexo C Tabla 2 cuyo encabezado es 0.05). Desafortunadamente, estas pruebas a veces fallan en

rechazar modelos con un ajuste pobre. Se debe tener cuidado en no aceptar un modelo solamente basado en las pruebas de Portmanteau. (Makridakis S., 1998)

5.2.1.1.5 Función de Autocorrelación Parcial (pacf)

Otra función muy útil para la identificación de modelos ARIMA es la denominada función de autocorrelación parcial (PACF del inglés Partial Autocorrelation Function).

La idea del análisis de la función PACF es medir como $Z_t \ y \ Z_{t+k}$ están relacionadas, pero con los efectos de las variables que intervienen $Z's$ tomadas en cuenta. Por ejemplo se quiere mostrar la relación existente entre ($Z_t \ y \ Z_{t+2}$) tomando en cuenta los efectos de $Z_{t+1} \ en \ Z_{t+2}$ Luego se quiere la relación entre ($Z_t \ y \ Z_{t+3}$), pero con los efectos de $Z_{t+1} \ y \ Z_{t+2}$ en Z_{t+3} tomados en cuenta, y así en adelante, cada rezago ajustado por el impacto de los $Z's$ que caen entre los pares en cuestión.

El coeficiente de autocorrelación parcial que mide esta relación entre $Z_t \ y \ Z_{t+k}$ es representado por $\widehat{\emptyset}_{kk}$. (Recuerde $\widehat{\emptyset}_{kk}$ es un estadístico que es calculado de la información muestral y provee un estimado del verdadero coeficiente de autocorrelación parcial $\emptyset_{kk}$).

A diferencia del coeficiente de autocorrelación acf que solo examina pares ordenados de $Z's$ pero sin tomar en cuenta los efectos de las $Z's$ que intervienen. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

El ***coeficiente de autocorrelación parcial*** de orden k se representa por $\emptyset_{kk}$ y es calculado mediante la regresión múltiple de Y_t contra $Y_{t+1,\dots,}Y_{t+k,}$:

$$Y_t = b_0 + \ b_1 Y_{t+1} + b_2 Y_{t+2} + \cdots + b_k Y_{t+k} + u_t \qquad (5.12)$$

Esta regresión múltiple es inusual ya que las variables independientes del lado derecho de la ecuación (5.12) son los valores previos de la variable a predecir Y_t, ya que estos son simples valores corridos en el tiempo de la variable a predecir por lo tanto se denomina ***autoregresión*** (AR). (Makridakis S., 1998)

Siendo u_t el término de error que representa todas las cosas que afectan a Y_t que no aparecen en ningún lugar de la ecuación de regresión.

El coeficiente de autocorrelación parcial $\emptyset_{kk}$ corresponde a b_k en la regresión múltiple. Como en cualquier regresión múltiple para su cálculo se consideran

constantes o no se toman en cuenta los efectos de las variables corridas en el tiempo $Y_{t+1}\ Y_{t+2},\ \ldots Y_{t+k-1}$.

Existe una forma iterativa (ligeramente menos precisa) que facilita la implementación computacional para estimar los coeficientes $\emptyset_{kk}$. Se necesita haber calculado el coeficiente de autocorrelación (r_k). Los datos deben ser estacionarios y el siguiente conjunto de ecuaciones recursivas, basadas en el conjunto de ecuaciones de Yule – Walker, nos dan un muy buen estimado de la autocorrelación parcial:

$$\widehat{\emptyset}_{11} = r_1$$

$$\widehat{\emptyset}_{kk} = \frac{r_k - \sum_{j=1}^{k-1} \widehat{\emptyset}_{k-1,j}\, r_{k-j}}{1 - \sum_{j=1}^{k-1} \widehat{\emptyset}_{k-1,j}\, r_j} \qquad (k = 2,3,\ldots) \qquad (5.13)$$

Donde

$$\widehat{\emptyset}_{kj} = \widehat{\emptyset}_{k-1,j} - \widehat{\emptyset}_{kk}\widehat{\emptyset}_{k-1,k-j} \qquad (k = 3,4,\ldots; j = 1,2,\ldots,k-1)$$

Ilustrar como el conjunto de ecuaciones (5.13) pueden calcular los coeficientes de autocorrelación parcial sería un proceso engorroso en su lugar se utilizarán programas computacionales.

5.2.1.1.6 Función de Autocorrelación Extendida (EACF)

Identificar el orden de un proceso AR puro o MA puro es relativamente fácil. El orden de un AR(p) puro es igual al retardo del último pico significativo de la función Spacf. El orden de un MA(q) puro es igual al último pico significativo en la función Sacf, es decir se aprovecha la propiedad de corte (en inglés ”cutting off”) de las dos funciones. Sin embargo, es una tarea más complicada identificar $p\ y\ q$ a partir de las funciones Sacf y Spacf de un modelo mixto (componentes AR y MA). La función de autocorrelación extendida (Eacf) y su contraparte muestral (Esacf) son especialmente útiles para identificar el orden de modelos mixtos no estacionales, ya que la identificación del orden de modelos estacionales es muy difícil mediante la función Esacf. La función Esacf podría también ser usada para identificar el orden de modelos MA y AR puros, pero la experiencia sugiere que es mejor el uso de las tres herramientas (Sacf, Spacf y Seacf). (Pankratz, Forecasting with Dynamic Regression Models, 1991).

No se entrará en detalles de la función Eacf. Si el lector desea profundizar podría utilizar la siguiente referencia, (Tiao G., 2001) pp: 61-86. En esta investigación se mostrarán solamente los resultados y como utilizar la función Esacf.

En el trabajo de Tsay y Tiao (1984) mostrado en la referencia arriba citada, se demuestra que para un modelo ARMA (p, q) *la función de autocorrelación extendida* $\rho(l, k)$ tiene la propiedad de corte arriba citada, así para $l = p$,

$$\rho(l,k) = \begin{cases} -\theta_q\left(1 + \theta_1^2 + \cdots + \theta_q^2\right)^{-1}, & k = q \\ 0, & k > q \end{cases} \qquad (5.14)$$

Donde $\rho(l, k)$ es la $l - ésima$ autocorrelación extendida del rezago k para Z_t. Además se demuestra que para un modelo ARMA (p,q) cuando $l \geq p$,

$$\rho(l,k) = \begin{cases} c, & k = q + l - p \\ 0, & k > q + l - p \end{cases} \qquad (5.15)$$

$Donde \ -1 < c < 1$. Esta propiedad será utilizada después. (Tsay, 2010 3rd Edition).

La Eacf (y Esacf) pueden ser presentadas en una forma tabular simplificada, como se muestra en la tabla 5.1. Celdas marcadas con x son teóricamente diferentes de cero (significativas en la función Esacf) y celdas marcadas con 0 son teóricamente cero (no significativas en la función Esacf).

Tabla 5.1 EACF Teórica par un modelo ARIMA(1,1,1), o ARIMA(2,0,1), o ARIMA(0,2,1)

$(q \rightarrow)$	0	1	2	3	4	5	6	7	8
$(p' = 0)$	X	X	X	X	X	X	X	X	X
$(p' = 1)$	X	X	X	X	X	X	X	X	X
$(p' = 2)$	X	0	0	0	0	0	0	0	0
$(p' = 3)$	X	X	0	0	0	0	0	0	0
$(p' = 4)$	X	X	X	0	0	0	0	0	0
$(p' = 5)$	X	X	X	X	0	0	0	0	0

(Pankratz, Forecasting with Dynamic Regression Models, 1991) pp:63

La idea es buscar un triángulo de ceros en la parte inferior derecha de la tabla 5.1. (En la práctica se puede formar un rectángulo en lugar de un triángulo.) El vértice superior izquierdo (o esquina del rectángulo) corresponde al orden p' del

componente AR del modelo a la izquierda de la tabla 5.1 y al orden q del componente MA del modelo arriba de la tabla 5.1.

Otra característica especial de la función eacf es que no es necesario tomar la diferencia de los datos a usar con la función esacf para identificar el orden del modelo. La función eacf incorpora el orden d de la diferencia en el orden p' autoregresivo. Entonces el orden p' autoregresivo dado por eacf es $p' = p + d$. Es decir la función eacf incorpora la naturaleza de la media no estacionaria de la serie en el orden AR. La característica no estacionaria presumiblemente será vista en la etapa de estimación cuando se chequeen si los coeficientes AR cumplen con las condiciones de estacionariedad. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

5.2.1.1.7 Estacionariedad y Función de Autocorrelación

La estacionariedad implica que no exista ni crecimiento ni decrecimiento en los datos. Los datos deben fluctuar alrededor de una media constante, independiente del tiempo y la varianza de la fluctuación debe permanecer constante en el tiempo. (Makridakis S., 1998)

Una forma de evaluar la estacionariedad es utilizando un gráfico.

- Si en el gráfico no hay evidencia de un cambio de la media con el tiempo, se dice que la serie es de media estacionaria.
- Si en el gráfico no existe un cambio obvio en la varianza, se dice que la serie es de varianza estacionaria.

El análisis visual de un gráfico de la serie de tiempo es muy importante para saber si una serie es estacionaria o no-estacionaria. (Makridakis S., 1998)

La función de autocorrelación estimada (acf) es muy útil para decidir si la media de una serie es estacionaria. Si la media es estacionaria la acf estimada decae rápidamente a cero. Si la media no es estacionaria la acf estimada decae lentamente hacia cero. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

El Dr Holger Capa de la EPN en la referencia (Capa, Un Primer Curso en Series Temporales, 2008) pp: 3-5, da una definición completa de un proceso denominado

débilmente estacionario. Así un proceso cuyas variables aleatorias toman valores reales, su media es constante (independiente del tiempo), la varianza del proceso es constante y la covarianza depende solamente de la longitud del rezago, se dice *débilmente estacionario*. En esta investigación al igual que en la referencia arriba indicada a un proceso débilmente estacionario se lo llamará estacionario.
Además en la misma referencia sugiere utilizar la prueba de Dickey - Fuller aumentada para evaluar la estacionariedad de la serie.

5.2.1.2 Diferencia

Como se ha indicado antes, la metodología ARIMA exige que las series de tiempo sean estacionarias. Afortunadamente muchas series de tiempo *no estacionarias* pueden ser transformadas a estacionarias. Así la metodología ARIMA puede ser utilizada aunque los datos no sean estacionarios. Esta transformación se denomina *diferencia* y es una operación relativamente simple.
La diferencia se utiliza cuando la media de la serie está cambiando a lo largo del tiempo.
Para tomar la diferencia de una serie de tiempo, se define una nueva variable (w_t) que cambia z_t de la siguiente forma,

$$w_t = z_t - z_{t-1}, \qquad t = 2,3,\dots,n \tag{5.16}$$

Esta diferencia normalmente produce una media constante en la serie de tiempo, la serie w_t es llamada *primera diferencia* de z_t. Si con esta primera diferencia no se obtiene una media constante, se redefine w_t como la primera diferencia de la primera diferencia:

$$w_t = (z_t - z_{t-1}) - (z_{t-1} - z_{t-2}) \qquad t = 3,4,\dots,n \tag{5.17}$$

Esta serie redefinida w_t se denomina *segunda diferencia* de Z_t, ya que resulta de tomar la diferencia dos veces Z_t. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)
Cuando la diferencia es necesaria para inducir una media estacionaria, se construye una serie w_t que es diferente de la serie original z_t. Entonces se construye un modelo ARIMA de la serie estacionaria w_t. Sin embargo lo que se necesita es predecir la serie original z_t es decir se necesita un modelo ARIMA para

esta serie. Afortunadamente esto no representa un problema serio ya que w_t y z_t están enlazadas por la definición (5.16) o (5.17) en el caso de una segunda diferencia.

Finalmente, si a una serie a la que se ha tomado la diferencia se ha vuelto estacionaria tiene una media cuyo valor es virtualmente cero. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.3 Desviación de la Media

Cuando la media de una serie de tiempo es estacionaria (constante en el tiempo) se podría tratar a la media como una componente determinística (significa fija o no estocástica) de la serie de tiempo. Para mantener un enfoque estocástico del comportamiento de la serie se expresan los datos en *desviaciones de la media*. Es decir se define una nueva serie $\tilde{z}_t$ como:

$$\tilde{z}_t = z_t - \bar{z} \qquad (5.18)$$

Donde $\bar{z}$ es un estimado del parámetro μ.

Esta nueva serie ($\tilde{z}_t$) se comporta exactamente igual que la serie (z_t) excepto que la media de la serie $\tilde{z}_t$ es precisamente cero en lugar de $\bar{z}$. Ya que conocemos $\bar{z}$ se podrá regresar al nivel original de la serie después de terminar el análisis con la serie $\tilde{z}_t$.

Cabe destacar que las dos series $\tilde{z}_t$ y z_t tienen las mismas propiedades estadísticas excepto por sus medias. Por ejemplo las varianzas de las dos series son idénticas.

5.2.1.4 Proceso, Realización y Modelo

Al igual que en la estadística básica es necesario familiarizarse con cierta terminología utilizada en la metodología de Box – Jenkins.

Se inicia preguntando ¿Qué clase de mecanismo produce una observación?

En la metodología de Box – Jenkins o ARIMA se asume que las observaciones son producidas por un *proceso* ARIMA. Cuyo concepto equivalente en la estadística clásica es la *población*. La población es el conjunto de todas las posibles observaciones de una variable; correspondientemente, un proceso ARIMA consiste en todas las posibles observaciones de una variable con una secuencia temporal.

Un proceso ARIMA no solo consiste en todas las posibles observaciones de una variable con una secuencia temporal, pero este también incluye una expresión algebraica a veces denominado *mecanismo de generación*, la cual especifica de que manera estas posibles observaciones se relacionan unas con otras.

En la estadística clásica se distingue entre *población* (todas las observaciones posibles) y *muestra* (un conjunto de observaciones actuales). Una muestra es un subconjunto particular de la población. La terminología ARIMA se refiere a la muestra como una *realización*. Una realización es un subconjunto de observaciones que se derivan de un proceso subyacente.

El objetivo de la metodología ARIMA es encontrar una buena representación del proceso que ha producido una realización dada. Esta representación es denominada *modelo*. Un modelo ARIMA es una expresión algebraica elegida a la luz de una realización disponible. El deseo es que el modelo que ajusta los datos disponibles (la realización) sea una buena representación del mecanismo de generación subyacente desconocido. Las tres etapas de la metodología ARIMA son diseñadas para guiarnos a este modelo apropiado.

Un proceso ARIMA se refiere a un conjunto de posibles observaciones de una variable con secuencia temporal junto con una expresión algebraica (mecanismo de generación) que describe como se relacionan estas observaciones. Los dos procesos ARIMA más comunes son: Proceso Autoregresivo (AR) y Proceso de medias móviles (MA) que se analizarán en detalle más adelante.

Aquí cabe una pregunta ¿***Cómo se decide si un modelo es bueno***?

Es importante recordar la diferencia entre modelo y proceso. En la práctica nunca se conoce que proceso ARIMA ha generado una realización dada. Lo que se hace es seguir un procedimiento de ensayo – error. En la parte del ensayo (etapa de identificación) el estadista se guía por la acf y pacf estimadas, calculadas de la realización. Se seleccionan algunos mecanismos de generación ARIMA hipotéticos con la esperanza de que ellos se ajusten a los datos adecuadamente. Estos posibles mecanismos de generación (ensayos) son *modelos*. Un modelo es diferente de un proceso: un proceso es el verdadero pero desconocido mecanismo que ha generado una realización, mientras un modelo es solamente una *imitación o representación* de un proceso. Ya que el proceso es desconocido, nunca se sabe

si un modelo seleccionado es el mismo que el verdadero proceso. Todo lo que se puede hacer es seleccionar un modelo que parezca adecuarse a los datos disponibles. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Box y Jenkins enfatizan en un principio clave, el de *parsimonia*, que significa "*ahorro"* un modelo con parsimonia ajusta los datos disponibles adecuadamente sin utilizar coeficientes innecesarios, este principio es muy importante ya que un modelo con parsimonia generalmente produce mejores pronósticos.

Otro punto clave es que un buen modelo de pronóstico usualmente ajusta bien el pasado, pero es más importante que pronostique satisfactoriamente el futuro.

Bien para responder a la pregunta planteada es importante tomar en cuenta los siguientes puntos resumidos en la tabla 5.2.

Un buen modelo es:

Tabla 5.2 Características de un buen modelo ARIMA

1. Tiene Parsimonia (utiliza el menor número de coeficientes necesarios para explicar los datos disponibles).
2. Estacionario (Tiene coeficientes AR que satisfacen algunas desigualdades matemáticas).
3. Invertible (Tiene coeficientes MA que satisfacen algunas desigualdades matemáticas)
4. Tiene coficientes ($\widehat{\emptyset}_s \; y \; \hat{\theta}_s$) de alta calidad: (a) Valores t absolutos mayores o cercanos a 2, (b) ($\widehat{\emptyset}_s \; y \; \hat{\theta}_s$) no son altamente correlacionados.
5. Tiene residuos no correlacionados
6. Ajusta los datos disponibles (el pasado) suficientemente bien para satisfacer al estadista: (a) Error cuadrático medio (RMSE) es aceptable, (b) Error porcentual medio absoluto (MAPE) es aceptable.
7. El pronóstico a futuro es satisfactorio.

(Pankratz, Forecasting with Dynamic Regression Models, 1991) pp:81

5.2.1.5 Inferencia Estadística

En la estadística básica se desea conocer algo acerca de la población, pero obtener toda la información relevante acerca de la población es frecuentemente imposible o demasiado costoso. Por lo tanto se infiere algo acerca de la población usando una muestra de la misma, utilizando conceptos de probabilidad, fórmulas y teorías de la estadística. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La inferencia estadística es la rama de la estadística que se ocupa del uso de los conceptos de probabilidad para manejar la incertidumbre en la toma de decisiones. La inferencia estadística está basada en la estimación y en las pruebas de hipótesis. Tanto en la estimación como en las pruebas de hipótesis, se harán inferencias acerca de las características de la población a partir de la información obtenida de las muestras. (Levin, 2010).

5.2.1.5.1 Pruebas para los Coeficientes de Autocorrelación

Los coeficientes de autocorrelación se calculan para una realización conocida. El coeficiente de autocorrelación estimado (r_k) es un estimado de su correspondiente (desconocido) coeficiente de autocorrelación teórico $\rho(k)$. No se espera que cada r_k sea exactamente igual a su correspondiente ρ_k debido al *error de muestro*.

Así para una realización digamos con $n = 100\ y\ k = 1$ aplicamos la fórmula de la Ec. (5.9) y se obtiene un cierto valor de r_1. Pero si se obtiene otra realización con $n = 100$ y recalculamos r_1 se obtiene probablemente un valor distinto. Y así otra realización daría todavía otro valor de r_1. Estas diferencias entre los valores se deben al error de muestreo.

Si se calcularan los valore de r_1 para todas las realizaciones posibles con $n = 100$ se obtendría una colección de los posibles valores de r_1. Esta distribución de los posibles valores es llamada *distribución de muestreo*.

R.L. Anderson (1942) demostró que los valores de r_k están normalmente distribuidos cuando $\rho_k = 0$ y n no es muy pequeño.

M.S. Barlett (1946) encontró la siguiente expresión para calcular el error estándar de la distribución de muestreo de los valores de r_k (este error es la raíz cuadrada de su varianza). Este error estándar, denominado $s(r_k)$ se calcula de la siguiente manera:

$$s(r_k) = \frac{(1+2\sum_{j=1}^{k-1} r_j^2)^{1/2}}{n^{1/2}} \quad (5.19)$$

Esta aproximación es apropiada para procesos estacionarios con variaciones aleatorias *normalmente distribuidas*.

El error estándar estimado se utiliza para probar la hipótesis nula $H_0: \rho_k = 0$ para $k = 1,2,3, \ldots$. Es común cuando se usa el estimado $s(r_k)$ en lugar del verdadero error estándar $\sigma(r_k)$ utilizar la distribución t en lugar de la distribución normal. Se prueba la hipótesis nula encontrando cuán lejos está es estadístico muestral r_k del valor hipotético $\rho_k = 0$, donde cuán lejos significa que el estadístico t es igual a un cierto número de errores estándar estimados. El estadístico t se calcula como:

$$t_{r_k} = \frac{r_k - \rho_k}{s(r_k)} \quad (5.20)$$

Debido a la hipótesis nula $\rho_k = 0$. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Una vez calculado el estadístico t se puede ver en el Anexo C Tabla C3 cuántos errores estándar bajo cero se encuentra, si el número calculado es mayor al valor de la tabla t, se rechaza la hipótesis nula, ya que el valor será significativamente diferente de cero.

Una ***regla práctica*** indica que solo el 5% de los posibles valores de r_k caerían ***dos o más errores estándar*** lejos de cero si $\rho_k = 0$. Así sí el estadístico t es mayor a 2 rechazamos la hipótesis nula ya que ρ_k es significativamente diferente de cero con un nivel del 5%. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

En la mayoría de programas aparecen líneas punteadas dos errores estándar arriba y abajo de cero, de esta manera visualmente se puede saber si los coeficientes de autocorrelación son significativos o no.

Cabe recalcar que estos cálculos son solo aproximaciones ya que se utiliza la fórmula de Bartlett (5.19) para calcular el error estándar de la distribución de muestreo de r_k.

5.2.1.5.2 Pruebas para los Coeficientes de Autocorrelación Parcial

Se puede también probar la significancia estadística de los coeficientes de autocorrelación parcial. El error estándar estimado es:

$$s(\emptyset_{kk}) = \frac{1}{n^{\frac{1}{2}}} \quad (5.21)$$

El estadístico t en este caso sería:

$$t_{\emptyset_{kk}} = \frac{\widehat{\emptyset}_{kk} - \emptyset_{kk}}{s(\widehat{\emptyset_{kk}})} \quad (5.22)$$

Así mismo, la hipótesis nula $\emptyset_{kk} = 0$.

De igual manera en la mayoría de programas aparecen líneas punteadas dos errores estándar arriba y abajo de cero, de esta manera visualmente se puede saber si los coeficientes de autocorrelación parcial son significativamente diferentes de cero a un nivel del 5%.

5.2.1.6 Notación con el Operador de Retardo B

El operador de retardo opera de la siguiente manera:

$$BZ_t = Z_{t-1} \quad (5.23)$$

Este operador es diferente a otros operadores utilizados por el álgebra, ya que este operador retrasa un período de tiempo a la variable si su exponente es 1, si su exponente es 2 retrasará dos períodos de tiempo, así:

$$B^2 Z_t = Z_{t-2} \quad (5.24)$$

Generalizando:

$$B^k Z_t = Z_{t-k} \quad (5.25)$$

Multiplicando una constante por B^k esta no es afectada.

$$B^k C = C \quad (5.26)$$

Se puede definir el operador de diferencia $(1 - B)$:

$$Z'_t = Z_t - Z_{t-1} = Z_t - BZ_t = (1 - B)Z_t \quad (5.27)$$

Se puede notar que la primera diferencia es representada por $(1 - B)$.

Nuevamente no se debe pensar al operador B como un número. Así (1- B) no es un valor numérico; es un operador que tiene sentido cuando se multiplica por una variable.

De manera similar las *diferencias de segundo orden* (es decir las primeras diferencias de las primeras diferencias) se calculan como:

$$Z_t'' = Z_t' - Z_{t-1}'$$
$$Z_t'' = (Z_t - Z_{t-1}) - (Z_{t-1} - Z_{t-2})$$
$$Z_t'' = (Z_t - 2Z_{t-1} + Z_{t-2})$$
$$Z_t'' = (1 - 2B + B^2)\, Z_t$$
$$Z_t'' = (1 - B)^2\, Z_t \qquad (5.28)$$

Se debe notar que la *diferencia de segundo orden* es representada por $(1 - B)^2$ mientras que la *segunda diferencia* se representa por $(1 - B^2)$.

En general la diferencia de orden d se escribe como:

$$(1 - B)^d\, Z_t \qquad (5.29)$$

(Makridakis S., 1998)

Algunos autores representan en forma condensada la diferencia de orden d mediante:

$$\nabla^d = (1 - B)^d \qquad (5.30)$$

(Box G., 2008 4th Edition), (Brockwell, 2009), (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Una diferencia regular seguida de una diferencia estacional se puede escribir como:

$$(1 - B)(1 - B^S)\, Z_t \qquad (5.31)$$

En el ítem 5.2.1.3 se sugirió la idea de expresar los datos de una realización en desviaciones de la media, definiendo $\tilde{z}_t = z_t - \bar{z}$. Donde $\bar{z}$ es la media de la realización del modelo. Cuando se escriba un proceso mediante el operador de retardo la variable aleatoria z_t se expresará en desviaciones de su media μ, definiendo $\tilde{z}_t = z_t - \mu$. Así el símbolo $\tilde{z}_t$ tendrá doble interpretación. Cuando se refiera a la *realización* basada en un modelo $\tilde{z}_t$ representará la desviación de la media de la realización $\bar{z}$. Cuando se trate de un *proceso* $\tilde{z}_t$ representará la desviación de la media del proceso μ. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

De manera más general se pueden definir series con el operador de retardo B, para esto se consideran solamente *procesos estacionarios*. (Capa, Un Primer Curso en Series Temporales, 2008).

Algunas propiedades de estas series se incluyen en el Apéndice C.

Ahora ya se puede escribir un proceso ARIMA (p, d, q) de manera compacta:

$$\emptyset(B)\nabla^d \widetilde{z_t} = \theta(B)a_t \quad (5.32)$$

Donde:

$\emptyset(B) = (1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p)$ Operador AR

$\theta(B) = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)$ Operador MA

$\nabla^d = (1 - B)^d$. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.7 Prueba de Dickey – Fuller

Como se analizó en el ítem (5.2.1.2) la metodología de Box – Jenkins exige que antes de la fase de identificación, las series sean estacionarias caso contrario se debe tomar los correctivos necesarios para lograr la estacionariedad de las series de tiempo. Uno de estos correctivos es tomar la primera diferencia de la serie.

Se había llamado w_t a la serie con una diferencia:

$w_t = z_t - z_{t-1} = (1 - B)z_t$

El polinomio autoregresivo asociado $\emptyset(B) = (1 - B)$ tiene una raíz de valor 1, se dice que el proceso w_t tiene una raíz unitaria, se dice además que la serie w_t es integrada de orden 1 y se representa por $I(1)$. En otros procesos se pueden presentar raíces unitarias de orden superior también. (Capa, Un Primer Curso en Series Temporales, 2008).

Para tomar la decisión de tomar la primera diferencia normalmente el estadista se basa en el gráfico de la serie de tiempo o en sus correlogramas, estos instrumentos empíricos para detectar la presencia de raíces unitarias pueden resultar un poco imprecisos. (Capa, Un Primer Curso en Series Temporales, 2008).

La metodología Dickey y Fuller se ha desarrollado para detectar si una serie tiene o no raíces unitarias (es o no estacionaria). En el programa estadístico EViews 9 se han implementado varias pruebas como son: Pruebas de Dickey – Fuller Aumentada y de Phillips – Perron. (Capa, Un Primer Curso en Series Temporales, 2008).

Dos referencias muestran en detalle esta prueba: (Capa, Un Primer Curso en Series Temporales, 2008) pp: 197-203 y (Yaffee, 2000) pp: 81-86, para el lector que desee profundizar.

En el presente trabajo se utilizará el paquete EViews para correr esta prueba, antes de la etapa de identificación, más adelante cuando se muestren algunos ejemplos.

5.2.1.8 Etapas de la Metodología de Box-Jenkins

A continuación se revisarán las tres etapas iterativas de la metodología de Box – Jenkins que son: Identificación, Estimación y Diagnóstico–Chequeo, para finalmente abordar la última etapa, Pronóstico.

5.2.1.8.1 Identificación de Modelos Estacionarios

Antes de analizar esta etapa podría ayudar el revisar algunas ideas básicas vistas anteriormente (sección 5.2.1.4):

1. Se inicia con un conjunto de n observaciones secuenciales en el tiempo de una sola variable $(Z_1, Z_2, Z_3, \dots, Z_n)$. Idealmente, debe tener al menos 50 observaciones. La realización se asume es generada por un proceso ARIMA desconocido.
2. Se supone las observaciones deberían ser autocorrelacionadas. Se mide la relación estadística entre un par de observaciones separadas por varios lapsos de tiempo (Z_t, Z_{t+k}), $k = 1,2,3, \dots$ calculando los coeficientes estimados de autocorrelación y autocorrelación parcial. Estos coeficientes son mostrados gráficamente por la función de autocorrelación estimada (sacf del inglés sample acf) y la función de autocorrelación parcial estimada (spacf del inglés sample pacf).
3. La metodología ARIMA es apropiada solamente para series de tiempo *estacionarias*. Una serie estacionaria tiene la media, varianza y coeficientes de autocorrelación que son esencialmente constantes en el tiempo. A menudo las series no-estacionarias pueden convertirse en estacionarias con trasformaciones apropiadas. El tipo más común de no-estacionariedad ocurre cuando la media de la realización cambia con el tiempo. Una serie no-estacionaria de este tipo puede frecuentemente convertirse en estacionaria por diferencia.

4. El objetivo es encontrar un buen modelo. Esto es, una representación estadísticamente adecuada y con parsimonia de una realización dada. (En la tabla 5.2 se enumeraron las características de un buen modelo).
5. En la etapa de identificación se comparan las funciones estimadas sacf y spacf con varias acfs y pacfs teóricas para encontrar una coincidencia. Se elige como modelo tentativo al proceso ARIMA cuyas acf y pacf teóricas mejor coincidan con las funciones sacf y spacf estimadas. Al elegir el modelo tentativo, se debe tener en mente el *principio de parsimonia*: se requiere un modelo que se ajuste a la realización dada con el menor número de parámetros estimados.
6. En la etapa de estimación se ajusta el modelo a los datos para obtener un estimado preciso de sus parámetros. Se examinan estos coeficientes para comprobar: estacionariedad, invertibilidad, significancia estadística entre otros indicadores de su calidad.
7. En la etapa de diagnóstico-chequeo se examinan los residuos del modelo estimado para ver si estos son independientes. Si no lo son se regresa a la etapa de identificación para tentativamente seleccionar otro modelo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La identificación es claramente una etapa crítica en el modelamiento ARIMA y el conocimiento de las funciones teóricas acfs y pacfs más comunes es requerido para una identificación efectiva. La familiarización con las funciones de autocorrelación teóricas aumentan las posibilidades de encontrar un buen modelo rápidamente.
Afortunadamente en la práctica ocurren comúnmente un reducido número de modelos de esa gran familia de modelos propuestos por Box y Jenkins. Así se necesita examinar las propiedades de unos pocos procesos comunes para poder identificar hasta los modelos más inusuales. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.1.1 Cinco Procesos ARIMA Estacionarios

Un ***modelo*** ARIMA se basa en los datos disponibles, su contraparte teórica (población) es un ***proceso*** ARIMA. Cada proceso ARIMA tiene su función de *autocorrelación teórica (acf) y su función de autocorrelación parcial (pacf)* asociada a este. A continuación se presentan algunos procesos ARIMA estacionarios más comunes y sus acfs y pacfs teóricas.

5.2.1.8.1.1.1 Procesos Autoregresivos

Dos procesos ARIMA muy comunes y simples son:

$$Z_t = C + \emptyset_1 Z_{t-1} + a_t \quad (5.33)$$

$$Z_t = C + \emptyset_1 Z_{t-1} + \emptyset_2 Z_{t-2} + a_t \quad (5.34)$$

En estos procesos los valores pasados de Z $(Z_{t-1}, Z_{t-2}, \dots)$ con sus coeficientes asociados se denominan términos autoregresivos (AR). El orden p de un proceso AR o (modelo) es el mayor retardo entre los términos AR.

Así (5.33) es un proceso AR(1) y (5.34) es un proceso AR(2). C es una constante que depende de la media μ de la serie Z_t y de sus coeficientes AR ($\emptyset_s$) como sigue:

$$C = \mu_z(1 - \sum_{i=1}^{p} \emptyset_i) \quad (5.35)$$

a_t se asume *ruido blanco de media cero y normalmente distribuido*. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Se debe tomar en cuenta en esta sección que se está tratando con acfs y pacfs teóricas obtenidas de procesos. Las funciones sacfs y spacfs estimadas calculadas de realizaciones nunca van a coincidir con las teóricas plenamente debido al *error de muestreo*. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

5.2.1.8.1.1.1.1 Acfs y Pacfs Teóricas

La figura 5.2 muestra las funciones acfs y pacfs teóricas asociadas con un proceso **AR(1)** (Proceso de Markov)**.**

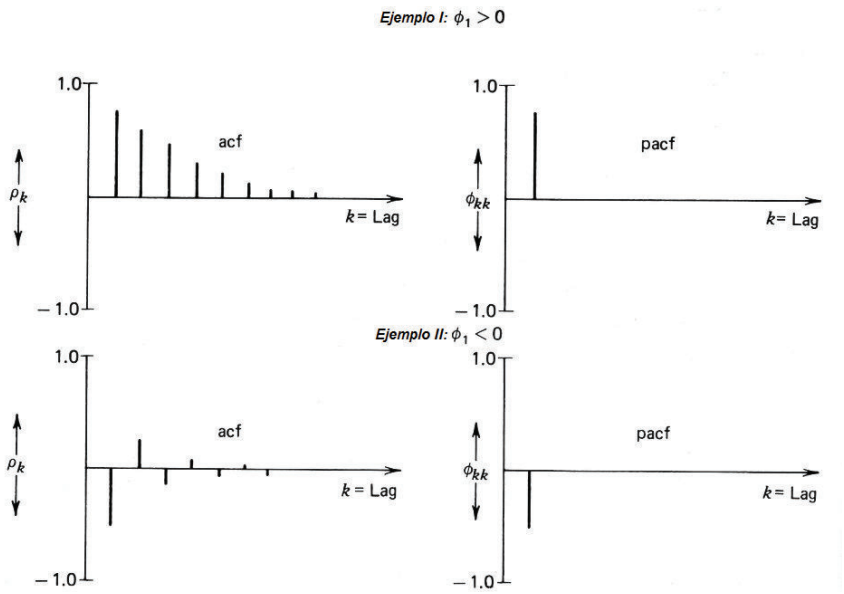


Figura 5.2 Ejemplos de acfs y pacfs teóricas para dos procesos AR(1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:124

Todos los procesos AR tienen una ***acf*** que decrece hacia cero. Este decrecimiento puede seguir un patrón exponencial, una onda seno amortiguada o una mezcla de los dos, más complicado.

Pero en todos los casos existe un decrecimiento hacia cero.

La ***pacf*** teórica para un proceso AR tiene picos hasta el retardo p sin ningún otro patrón a continuación. En la práctica p no es mayor a 2 o 3 para modelos no estacionarios.

Cuando $\emptyset_1 > 0$ como en la parte superior de la figura 5.2, la función acf decrece todo en el lado positivo y la función pacf tiene un solo pico positivo.

Cuando $\emptyset_1 < 0$ como en la parte inferior de la figura 5.2, la función acf decrece alternando el signo iniciando en el lado negativo y la función pacf tiene un solo pico negativo.

Los valores numéricos exactos de los coeficientes en las dos funciones teóricas acf y pacf de un AR(1) se determinan mediante el valor de $\emptyset_1$. Así en el rezago 1, los dos coeficientes de autocorrelación (ρ_1) y autocorrelación parcial $(\emptyset_{11})$, son iguales a $\emptyset_1$.El resto de coeficientes de autocorrelación parcial son cero.

Los coeficientes de autocorrelación siguientes son iguales a $\emptyset_1$ elevado a la potencia k donde k es la longitud del rezago. En general $\rho_k = \emptyset_1^k$. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La figura 5.3 muestra ejemplos de acfs y pacfs asociados a un proceso **AR(2)** como el de la ecuación (5.34).

La función acf de un proceso AR(2) tiene una mayor variedad de patrones que un AR(1), esta función muestra un decrecimiento exponencial y una onda seno amortiguada que también decrece. Sin embargo, en todos los casos la función acf decrece hasta cero y la función pacf tiene dos picos en los rezagos 1 y 2. El patrón exacto depende de los signos y tamaños de los coeficientes $\emptyset_1 \; y \; \emptyset_2$.

Para determinar la naturaleza de estos patrones se analiza la ecuación característica del proceso AR(2): $(1 - \emptyset_1 B - \emptyset_2 B^2) = 0$, donde el operador de retardo B es tratado como una variable ordinaria.

Tomar en cuenta lo siguiente para la función *acf de un AR(2):*

1. Si las raíces de $(1 - \emptyset_1 B - \emptyset_2 B^2) = 0$ son reales, es decir su discriminante $\Delta = \emptyset_1^2 + 4\emptyset_2 \geq 0$, y la *raíz dominante es positiva*, entonces la función acf decrece a cero desde el lado positivo.
2. Si las raíces son reales ($\Delta = \emptyset_1^2 + 4\emptyset_2 \geq 0$), pero la *raíz dominante es negativa*, entonces la función acf decrece a cero mientras alterna el signo.
3. Si las raíces son complejas ($\Delta = \emptyset_1^2 + 4\emptyset_2 < 0$) y $\emptyset_1$ *es positivo*, la función acf tiene forma de una onda seno amortiguada iniciando en el lado positivo.

4 Si las raíces son complejas, pero $\emptyset_1$ *es negativo*, la función acf tiene forma de una onda seno amortiguada iniciando en el lado negativo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

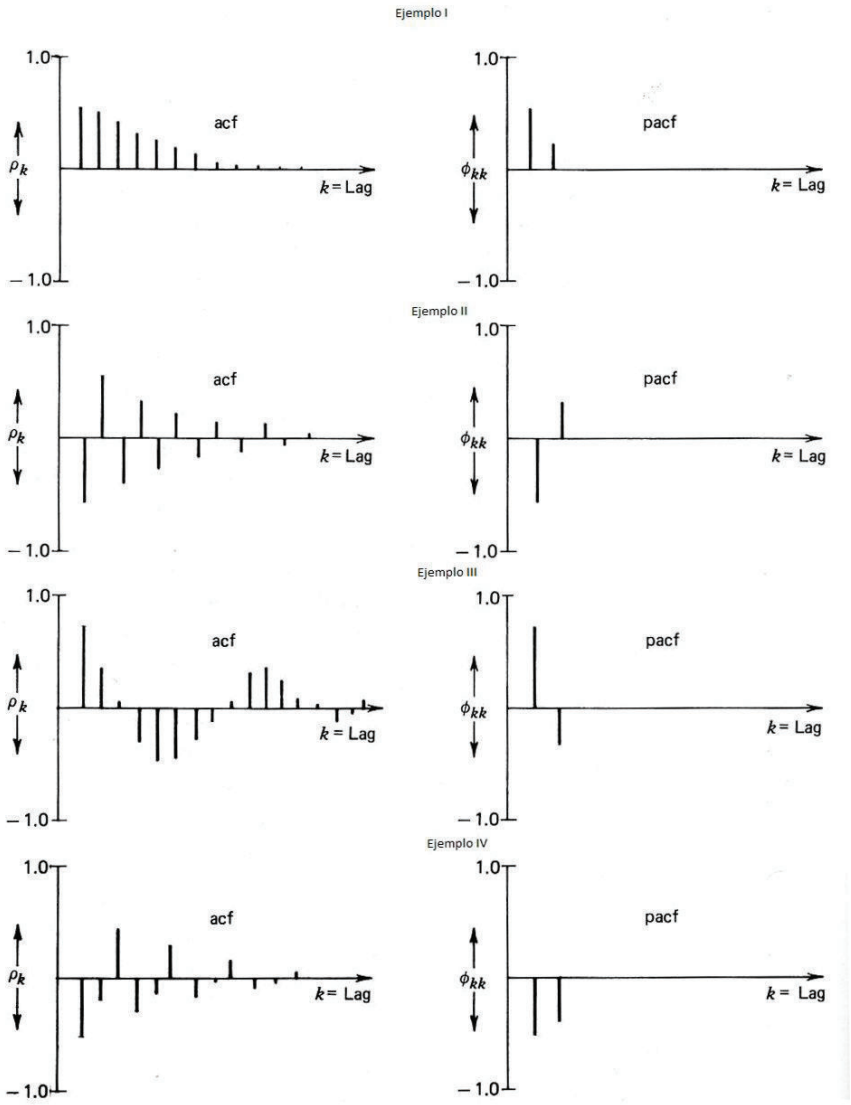


Figura 5.3 Ejemplos de acfs y pacfs teóricas para procesos AR(2)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:126

En la figura 5.4 se muestran ejemplos de funciones de autocorrelación ρ_k y autocorrelación parcial $\emptyset_{kk}$ para varios modelos estacionarios AR(2). Además se muestran los valores permitidos para $\emptyset_1$ y $\emptyset_2$ en la misma figura.

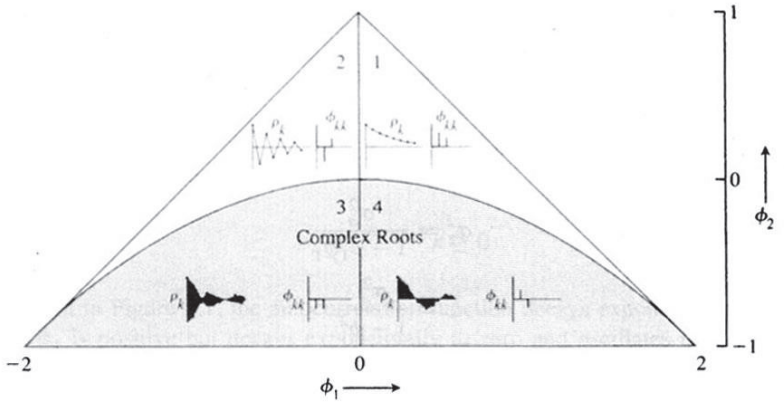


Figura 5.4 Funciones típicas acfs y pacfs teóricas para procesos AR(2)

(Box G., 2008 4th Edition) pp: 62

Ciclos Estocásticos

Un caso especial se da cuando el discriminante de la ecuación característica es menor a cero ($\Delta = \emptyset_1^2 + 4\emptyset_2 < 0$) en un modelo AR(2) ya que se genera un ciclo estocástico cuyo *período promedio* (p^*) se calcula mediante:

$$p^* = \frac{360^o}{cos^{-1}\left(\frac{\emptyset_1}{2(-\emptyset_2)^{\frac{1}{2}}}\right)} \tag{5.36}$$

Donde el coseno inverso (cos^{-1}) está en grados. Si el modelo es no estacional, las unidades de (p^*) son las mismas que los datos originales. Por ejemplo si los datos son mensuales y $p^* = 3$ el ciclo se repite cada tres *meses* en promedio. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Si se escribe la solución compleja como $a \pm bi$, se tiene $\emptyset_1 = 2a, \emptyset_2 = -(a^2 + b^2)$ y

$$p^* = \frac{360^\circ}{cos^{-1}\left(a/\sqrt{a^2+b^2}\right)}$$

Donde $\sqrt{a^2 + b^2}$ es módulo del número complejo $a \pm bi$. (Tsay, 2010 3rd Edition).

Procesos AR estacionarios de orden $p > 2$ tienen patrones de acfs y pacfs similares a los de las figuras 5.2 y 5.3.

Resumiendo, ***procesos AR estacionarios*** de orden p tienen las siguientes características:

1. La función acf teórica decrece o decae, con patrones exponenciales u ondas seno amortiguadas o los dos a la vez.

2. La función pacf teórica tiene *picos* hasta el rezago p, luego todos son cero. (Algunos valores antes del rezago p podrían ser cero; lo importante es que el último valor diferente de cero ocurre en el rezago p).

Para recordar el uso de las funciones acf y pacf teóricas en la práctica: Si se ve un par de funciones sacf y spacf de una muestra que se parecen mucho a una de las funciones teóricas mostradas en las figuras 5.2 y 5.3 se identifica un modelo AR(1) o AR(2) respectivamente, como un modelo razonablemente tentativo para los datos disponibles. El modelo que se elija corresponde al proceso cuyas acf y pacf coincidan con las sacf y spacf de la muestra. Luego se estima y cuequea el modelo tentativo para ver si este es adecuado. Esto se ilustrará más adelante. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Condiciones de Estacionariedad

Los coeficientes $\emptyset$ deben satisfacer ciertas condiciones para que el proceso sea estacionario. Para un AR(1) se requiere:

$$|\emptyset_1| < 1 \tag{5.37}$$

Para un AR(2) las raíces de su ecuación característica $(1 - \emptyset_1 B - \emptyset_2 B^2) = 0$ deben estar fuera del círculo unitario, es decir se deben cumplir ***todas*** las siguientes condiciones:

$$|\emptyset_2| < 1$$
$$\emptyset_2 + \emptyset_1 < 1 \tag{5.38}$$
$$\emptyset_2 - \emptyset_1 < 1$$

En la práctica no se pueden observar los coeficientes del proceso AR, en su lugar se observa que los coeficientes estimados del modelo satisfagan las condiciones de estacionariedad.

Las condiciones de estacionariedad se vuelven más complicadas cuando $p > 2$. Afortunadamente, modelos ARIMA con $p > 2$ no ocurren frecuentemente en la práctica. Cuando $p > 2$ se puede chequear al menos la siguiente condición de estacionariedad necesaria (pero no suficiente):

$$\emptyset_1 + \emptyset_2 + \cdots + \emptyset_p < 1 \tag{5.39}$$

Además las raíces de su ecuación característica deben estar fuera del círculo unitario. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.1.1.2 Procesos de Medias Móviles y Mixtos

Los otros tres procesos comunes son:

$$Z_t = C - \theta_1 a_{t-1} + a_t \quad (5.40)$$

$$Z_t = C - \theta_1 a_{t-1} - \theta_2 a_{t-2} + a_t \quad (5.41)$$

$$Z_t = C + \emptyset_1 Z_{t-1} - \theta_1 a_{t-1} + a_t \quad (5.42)$$

Los términos pasados de a_t con sus coeficientes son llamados términos de Medias Móviles (MA del inglés Moving Average).

El término pasado a_{t-k} no es el valor pasado de Z_t, pero es la componente aleatoria de Z_{t-k}; así un término MA representa parte de un valor pasado de Z_t. Como una convención *los términos MA se escriben con signo negativo*.

El orden (q) de un proceso MA (o modelo) es el rezago de más alto grado de los términos MA. Por ejemplo en (5.40) tenemos un MA de primer orden $(q = 1)$, también llamado MA(1), mientras (5.41) es un proceso MA(2). El proceso (5.42) es mixto (AR y MA) con $p = 1\, y\, q = 1$, se denomina ARMA (1,1) (del inglés autoregresive moving average). C es una constante que se relaciona con μ_z y los coeficientes AR ($\emptyset_s$) si existe alguno como se muestra en la ecuación (5.35). (Pankratz, Forecasting with Dynamic Regression Models, 1991).

5.2.1.8.1.1.2.1 Acfs y Pacfs Teóricas

La figura 5.5 muestra ejemplos de funciones acfs y pacfs asociadas con procesos MA(1) tal como (5.40).

En los dos casos pacfs decrecen hacia cero y la función acf tiene un pico en el rezago 1.

Cuando $\theta_1 < 0$, la parte superior de la figura 5.5, el pico en la función acf es positivo y pacf decae exponencialmente alternando el signo desde el lado positivo para $k = 1$.

Cuando $\theta_1 > 0$, la parte inferior de la figura 5.5, el pico en la función acf es negativo y pacf decae exponencialmente, todo en el lado negativo.

La figura 5.7 muestra ejemplos de funciones acfs y pacfs asociadas con procesos MA(2) tal como (5.41).

Los procesos MA(2) tienen una mayor variedad de patrones de acfs y pacfs que los MA(1). Sin embargo, en todos los casos la función acf tiene picos hasta el rezago 2 luego es cero y pacf decrece.

En *la figura 5.6* se muestran ejemplos de funciones de autocorrelación ρ_k y autocorrelación parcial $\emptyset_{kk}$ para varios modelos estacionarios MA(2). Además se muestran los valores permitidos para $\theta_1 \, y \, \theta_2$ en la misma figura.

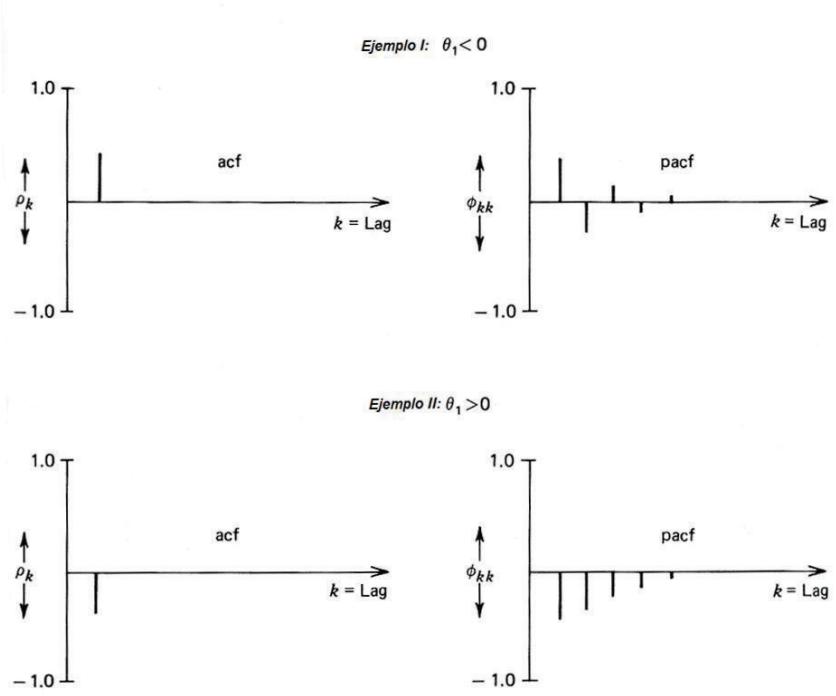


Figura 5.5 Ejemplos de acfs y pacfs teóricas para procesos MA(1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:127

Las figuras 5.8 y 5.9 muestran ejemplos de funciones acfs y pacfs asociadas con procesos ARMA(1,1) tal como (5.42). La característica principal es que las dos funciones acf y pacf decaen. Las funciones acf y pacf pueden decrecer alternando el signo también.

Procesos mixtos de ordenes más altos ($p > 1 \, y \, q > 1$) también tienen fuciones acf y pacfs que decaen. Identificar los órdenes exactos de un modelo mixto es más difícil que identificar el orden de un modelo puramente AR o puramente MA. Una

herramienta de identificación conocida en estos casos es la *función de autocorrelación extendida* (EACF del inglés Extended autocorrelation function), esta función se verá más adelante. (Pankratz, Forecasting with Dynamic Regression Models, 1991). Resumiendo:

1. La función acf teórica de un MA tiene picos hasta el rezago q, luego todos son cero.
2. A diferencia de la función pacf de un modelo AR, la función acf provee información de los retardos del modelo MA diferentes de cero. (Tsay, 2010 3rd Edition). Por lo tanto sugiere coeficientes MA en esos retardos.
3. La función pacf teórica de un MA es decreciente.
4. Las funciones pacf y acf teóricas de un proceso mixto ambas decrecen.

Para recordar el uso de las funciones acf y pacf teóricas en la práctica: Si se ve un par de funciones acf y pacf de una muestra que se parecen mucho a una de las funciones teóricas mostradas en las figuras 5.5, 5.7 y 5.9 se identifica un modelo MA(1), MA(2) o ARMA(1,1) respectivamente, como un modelo razonablemente tentativo para los datos disponibles. El modelo que se elija corresponde al proceso cuyas acf y pacf coincidan con las sacf y spacf de la muestra. Luego se estima y cuequea el modelo tentativo para ver si este es adecuado. (Pankratz, Forecasting with Dynamic Regression Models, 1991)

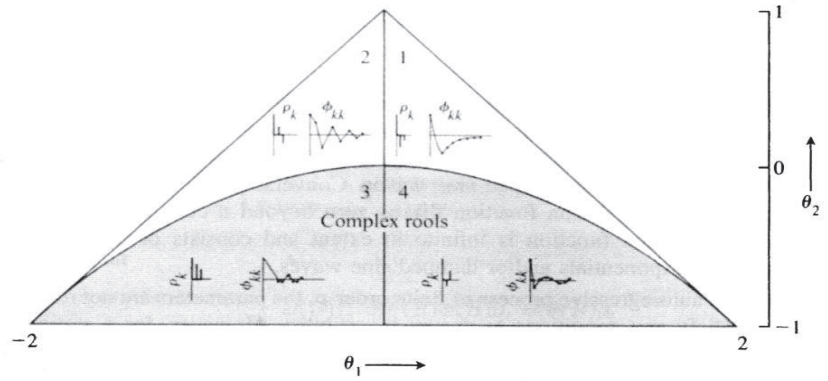


Figura 5.6 Funciones típicas acfs y pacfs teóricas para procesos MA(2)

(Box G., 2008 4th Edition) pp: 62

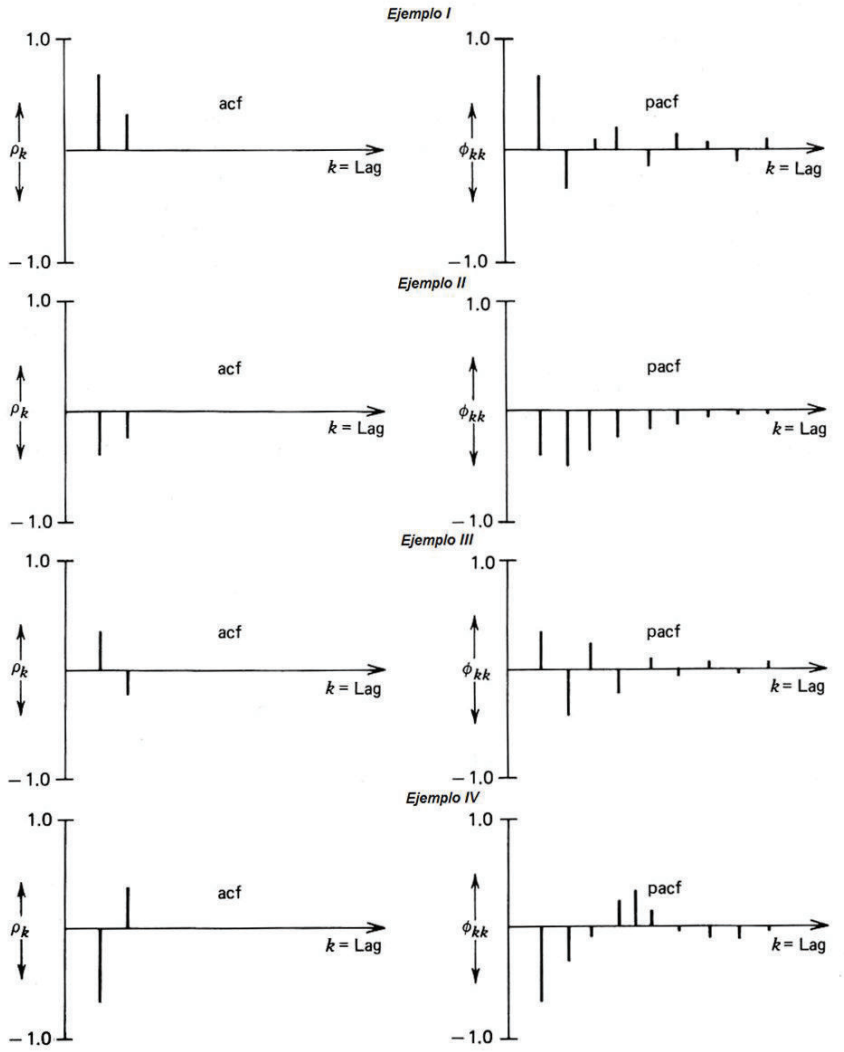


Figura 5.7 Ejemplos de acfs y pacfs teóricas para procesos MA(2)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:128

Invertibilidad

Anteriormente se analizaron condiciones de estacionariedad para procesos AR. Procesos MA requieren condiciones similares para *invertibilidad*. Los coeficientes MA deben satisfacer ciertas condiciones, estas son algebraicamente idénticas a las condiciones de estacionariedad de los coeficientes AR.

Si $q = 0$ tenemos procesos AR puros o series de ruido blanco. Todos los procesos AR puros (o ruido blanco) son invertibles y no se necesita chequear nada adicional.

Para procesos MA(1) o ARMA($p, 1$), la invertibilidad requiere que el valor absoluto de θ_1 sea menor a uno:

$$|\theta_1| < 1 \tag{5.43}$$

Para un MA(2) o ARMA $(p, 2)$ las raíces de su ecuación característica $(1 - \theta_1 B - \theta_2 B^2) = 0$ deben estar fuera del círculo unitario, es decir se deben cumplir ***todas*** las siguientes condiciones:

$$|\theta_2| < 1$$
$$\theta_2 + \theta_1 < 1 \tag{5.44}$$
$$\theta_2 - \theta_1 < 1$$

Y es estacionario para todos los valores de $\theta_1 \; y \; \theta_2$. (Box G., 2008 4th Edition).

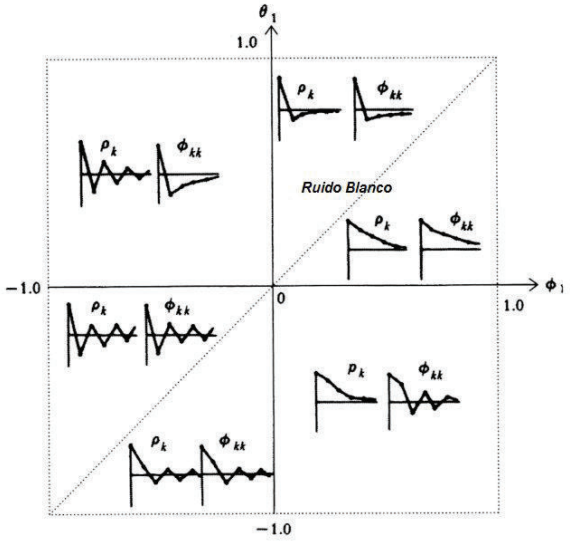


Figura 5.8 Funciones acfs y pacfs teóricas para varios procesos ARMA(1,1)

(Box G., 2008 4th Edition) pp: 84

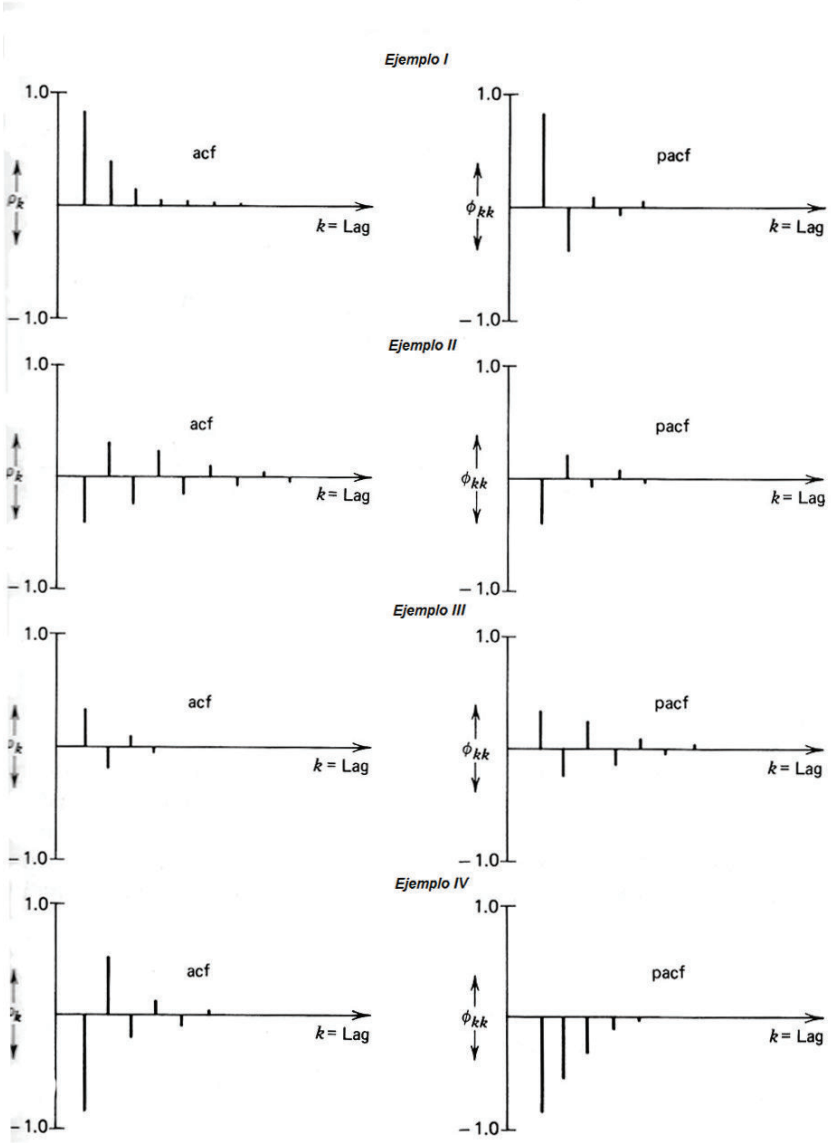


Figura 5.9 Ejemplos de acfs y pacfs teóricas para procesos ARMA(1,1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:129

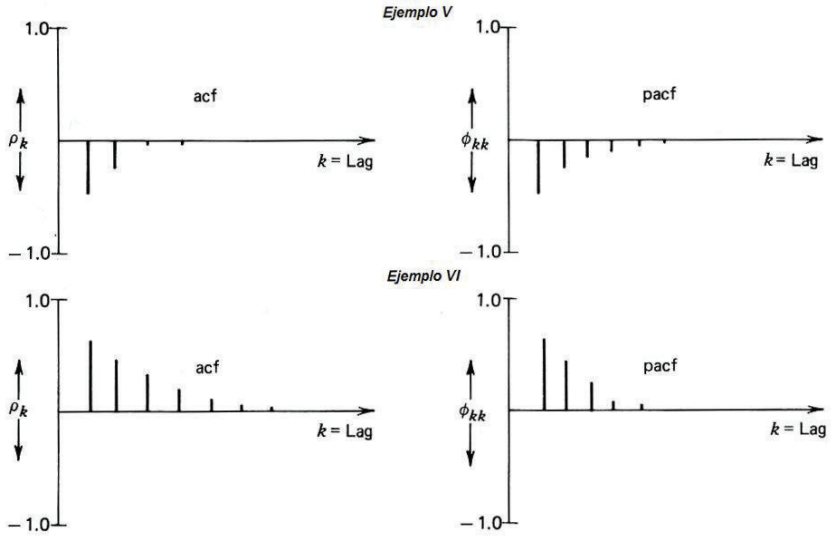


Figura 5.9 Continuación

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:130

En la práctica no se pueden observar los coeficientes del proceso MA, en su lugar se observa que los coeficientes estimados del modelo satisfagan las condiciones de invertibilidad.

Las condiciones de invertibilidad se vuelven más complicadas cuando $q > 2$. Afortunadamente, modelos ARIMA con $q > 2$ no ocurren frecuentemente en la práctica. Cuando $q > 2$ se puede chequear al menos la siguiente condición de invertibilidad necesaria (pero no suficiente):

$$\theta_1 + \theta_2 + \cdots + \theta_q < 1 \qquad (5.45)$$

Además las raíces de su ecuación característica deben estar fuera del círculo unitario. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Se debe recalcar que si se tiene un modelo ARIMA se deben chequear las condiciones de *estacionariedad* solo con los coeficientes AR y las condiciones de *invertibilidad* solo se aplican a los coeficientes MA.

Razones para la invertibilidad

El sentido común de la invertibilidad se puede explicar en dos pasos:

1. Un proceso MA invertible tiene su forma equivalente AR (aunque sin cumplir el principio de parsimonia). El lector puede ver la demostración en la referencia (Box G., 2008 4th Edition) pp: 52.
2. La invertibilidad asegura que los pesos de las observaciones pasadas de Z_t en la forma equivalente AR disminuyan mientras se mueve más hacia el pasado. Un ejemplo explicativo se puede observar en la referencia (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp: 136.

Además la invertibilidad garantiza que, para un proceso estacionario, a cualquier función acf teórica dada le corresponde un único proceso ARIMA.

Finalmente existe un teorema que expresa lo siguiente: En la práctica, la forma no invertible de un modelo no puede producir mejor pronóstico que la forma invertible, basado en el criterio de mínimos cuadrados. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.1.2 Identificación de Modelos No Estacionarios

En la práctica algunas realizaciones son no estacionarias. En esta sección se analizará cómo transformar (no siempre es posible) una realización no estacionaria a estacionaria. Si esta transformación se logra encontrar, se aplica la metodología de Box – Jenkins (identificación, estimación y diagnóstico – chequeo) a la serie transformada a estacionaria. Después de modelar la serie transformada, se reversa el procedimiento de transformación para obtener el pronóstico de la serie original, no estacionaria. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.1.2.1 Media No Estacionaria

El tipo más común de no estacionariedad se da cuando la media de la serie no es constante. En algunas series de tiempo el nivel general de la serie puede tender hacia arriba o hacia abajo, en otras pueden cambiar el nivel y la pendiente de la serie a la vez. Un ejemplo se muestra en la figura 5.10.

Una característica importante de una realización es la llamada ***no estacionariedad homogénea***. Esto es diferentes segmentos de la serie se comportan de manera

muy similar al resto de la serie después de permitir los cambios en el nivel y/o pendiente. Esta característica es importante ya que una realización homogéneamente no estacionaria puede convertirse a estacionaria simplemente tomando la diferencia.

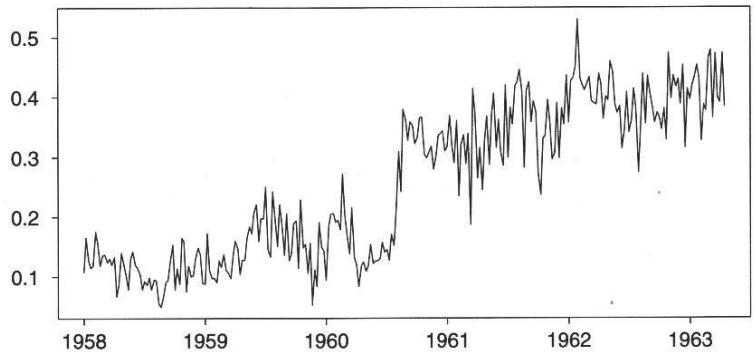


Figura 5.10 Serie con media No Estacionaria

(Tiao G., 2001) pp:7

Se debe tomar en cuenta que tomar *la diferencia* es un procedimiento para tratar con medias no estacionarias, más no con *varianza* no estacionaria.

Se pierde una observación cada vez que se diferencia una serie. Mientras más se diferencia una serie la media se acerca a cero. Mientras $w's$ son las diferencias de $Z's$, las $Z's$ son las sumas de $w's$. Se obtiene la serie Z_t integrando (sumando) sucesivamente los valores de w_t.

Procedimiento de Identificación

Sea w_t una serie diferenciada:

$w_t = \nabla^d Z_t$

$w_t = (1 - \mathrm{B})^d Z_t$ (5.46)

Luego de que una serie no estacionaria Z_t ha sido transformada a una serie diferenciada estacionaria w_t, entonces w_t es modelada con la metodología de Box – Jenkins vista anteriormente para series estacionarias.

Un modelo ARMA (p, q) para una serie diferenciada w_t es también un modelo ARIMA (p, d, q) integrado para la serie no diferenciada o integrada Z_t, con $p\ y\ q$ iguales para los dos modelos. El enlace entre los dos modelos es la definición (5.46)

que significa que las $w's$ se obtienen diferenciando las $Z's$ d veces, y las $Z's$ se obtienen integrando las $w's$ d veces. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Para cualquier realización se debe seleccionar el grado apropiado de diferencia (el valor de d) antes de elegir los términos AR y MA a incluir en el modelo. Si la serie original es estacionaria, no necesita diferencia así $d = 0$. Cuando segmentos de la serie difieren solo en nivel una sola diferencia es suficiente para inducir una media estacionaria. Cuando una serie tiene nivel y pendiente variables en el tiempo tomar dos veces la diferencia inducirá una media estacionaria, así $d = 2$.

En la práctica, una primera diferencia es requerida frecuentemente mientras una segunda diferencia es necesaria solo ocasionalmente. Tomar la diferencia más de dos veces es virtualmente imposible. Diferencias innecesarias crean patrones artificiales que tienden a reducir la precisión del pronóstico. Por otro lado, Box y Jenkins sugieren que, en una situación de pronóstico, a una serie se debe tomar la diferencia si existe duda de si la formulación estacionaria o no estacionaria es apropiada. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Cómo se elige el valor de d ? Existen varios procedimientos complementarios:

1. Examine los datos visualmente. Esto nos da una pista del grado apropiado de diferencia. Aunque el análisis visual puede ser de mucha ayuda, no confíe en este exclusivamente para determinar el grado de diferencia.
2. Examine los acfs de las series originales y de las series diferenciadas. La función sacf de una serie no estacionaria decrecerá lentamente hacia cero. No necesariamente debe decaer de valores altos cercanos a 1 ($r = 1$) sino el punto importante es que la función sacf decae a cero muy lentamente.
3. Chequear los coeficientes estimados para ver si satisfacen las condiciones de estacionariedad vistas anteriormente. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).
4. Utilizar la prueba de Dickey Fuller aumentada.

5.2.1.8.1.2.2 Varianza No Estacionaria

Algunas realizaciones tienen una varianza que cambia con el tiempo. Esto ocurre frecuentemente en series económicas y de negocios que cubren un largo espacio de tiempo, especialmente cuando hay elementos estacionales en los datos. Tales series deben ser transformadas para inducir una varianza constante antes de ser modeladas con la metodóloga de Box – Jenkins. También es posible que no se encuentre una transformación segura para algunas series.

Series con varianza no estacionaria tienen una media no estacionaria también. Estas series deben ser transformadas para inducir una varianza constante y luego diferenciadas para inducir una media constante antes de ser modeladas. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La figura 5.11 muestra un ejemplo de este tipo de series.

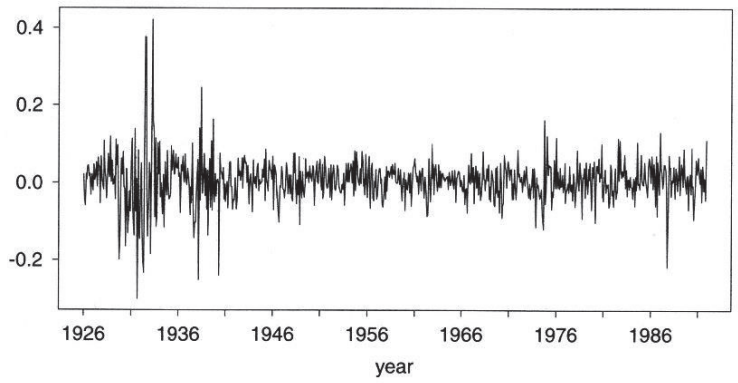


Figura 5.11 Serie con Varianza No Estacionaria

(Tiao G., 2001) pp:8

La varianza en la figura 5.11 es variable a medida que pasa el tiempo. Es necesario aplicar una transformación para inducir una varianza constante. Así por ejemplo:

1. Si la *desviación estándar* de la serie es proporcional a su nivel, tomar el logaritmo natural produce una nueva serie con varianza constante.
2. Si *la varianza* de la serie es proporcional a su nivel, tomar la raíz cuadrada induce a una varianza constante.

En la práctica algunas transformaciones son posibles pero estas dos son las más comunes especialmente el logaritmo natural (Ln). La transformación logarítmica es además interpretable: cambios en los valores logarítmicos son relativos

(porcentaje) de las medidas originales. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Transformaciones de Box – Cox

El logaritmo y la raíz cuadrada son miembros de una familia de poderosas transformaciones llamadas *transformaciones de Box – Cox*.

Con la transformación se define una nueva serie (trasformada) Z_t' como:

$$Z_t' = \frac{Z_t^{\lambda}-1}{\lambda} \quad \text{Si } \lambda \neq 0 \qquad (5.47)$$

$$Z_t' = Ln\, Z_t \quad \text{Si } \lambda = 0 \qquad (5.48)$$

Donde λ es un número real. Note que Z_t no debe ser negativa. Si algunos valores de Z_t son negativos, se puede incluir una constante positiva a Z_t para que todos sus valores sean positivos.

Restando 1 de Z_t^{λ} y dividiendo el resultado para λ no altera la estructura de tiempo de la varianza de la serie, además (5.47) tiene varias propiedades atractivas:

1. Preserva el orden de los datos cuando λ es negativo,
2. Es continua cuando $\lambda \to 0$,
3. Nos produce una transformación logarítmica cuando $\lambda \to 0$.

El análisis del gráfico de los datos nos da una idea de la trasformación apropiada. Si la varianza tiende a aumentar y el nivel de la serie también aumenta, se recomienda un $\lambda < 1$. Si la varianza tiende a disminuir y el nivel de la serie aumenta se recomienda un $\lambda > 1$.

Transformaciones simples e interpretables se prefieren; así, se tiende a elegir 0.5 (Raiz cuadrada) en lugar de 0.54 por ejemplo, y $\lambda = 0$ (transformación logarítmica) en lugar de $\lambda = -0.08$. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Además se recomienda utilizar la transformación logarítmica Ln cuando:

Existe un crecimiento exponencial de la tendencia o

Un crecimiento de la variabilidad (ΔZ_t) de la serie junto con una estabilidad en la media. (Capa, Un Primer Curso en Series Temporales, 2008).

Normalmente se necesita pronosticar los datos originales en lugar de los valores logarítmicos.

Al re-transformar la serie pronosticada Z_t' a la serie original Z_t se lo debe realizar con cuidado. Ya que si el error de la serie Z_t' está normalmente distribuido (o se asume típicamente) el error de la serie original Z_t no tiene una distribución normal. Es decir al realizar la conversión al pronóstico de la serie original a partir de la serie transformada se debe tomar en cuenta un sesgo existente.

5.2.1.8.1.2.3 Tendencia Determinista

Cuando la media $\hat{\mu}$ de la serie de datos original Z_t es estacionaria es decir no se requiere ninguna diferencia, $\hat{\mu}$ generalmente es diferente de cero. Por lo tanto el modelo de la serie no diferenciada tendrá un término constante estimado ($\hat{C}$) como se indicó anteriormente en (5.35) es igual a $\hat{C} = \hat{\mu}\,(1 - \sum \emptyset_i)$, en la práctica si $\hat{\mu}$ es estadísticamente diferente de cero entonces $\hat{C}$ será típicamente diferente de cero.
Suponga que una serie ha sido diferenciada $(d > 0)$ para lograr una media estacionaria. La serie resultante W_t a menudo tiene una media μ_w ($\widehat{\mu_w}$) que no es significativamente diferente de cero. El modelo que representa a esta serie diferenciada tendrá un término constante igual a cero.
Pero ocasionalmente cuando $(d > 0)$ la serie diferenciada resultante W_t tiene una media que es significativamente diferente de cero $\mu_w \neq 0$. El modelo resultante para W_t por lo tanto usualmente tendrá un término constante diferente de cero. Es decir la serie integrada Z_t tendrá una *tendencia determinista*. En la práctica modelos con tendencia determinista no son muy comunes a excepción de las ciencias físicas. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.2 Estimación

En la etapa de identificación se seleccionan uno o varios modelos que parecen estadísticamente adecuados y con el menor número de parámetros (principio de parsimonia) para la representación de los datos disponibles.
La etapa de identificación es necesaria para tener una idea preliminar y poder decidir el modelo a estimar. En esta etapa (Identificación) se obtiene un estimado

preciso de unos pocos parámetros que se ajustan al modelo tentativo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La estimación de modelos ARIMA se la realiza mediante programas estadísticos, los principales algoritmos son: Mínimos Cuadrados no condicionales, Mínimos Cuadrados Condicionales y Máximo de Verosimilitud.

El método de mínimos cuadrados es un algoritmo iterativo donde se elige un modelo preliminar y el programa estadístico va actualizando iterativamente hasta que la suma del cuadrado de los errores es minimizada.

Los algoritmos de mínimos cuadrados condicional y no condicional son casi idénticos, la diferencia entre ellos es la manera de escoger los valores iniciales. El algoritmo de mínimos cuadrados no condicional escoge los valores iniciales mediante Retro predicción(backcasting) y sus valores son muy cercanos a los reales, mientras el algoritmo de mínimos cuadrados condicional simplemente encera los valores iniciales. Si la serie es suficientemente grande los dos algoritmos producen similares resultados en la estimación. (Yaffee, 2000).

Para series estacionales la aproximación condicional no es muy satisfactoria y el cálculo no condicional se vuelve más necesario. (Box G., 2008 4th Edition).

El otro algoritmo que es frecuentemente utilizado es el *Máximo de Verosimilitud*. La función de Verosimilitud se representa por L y es proporcional a la probabilidad de obtener los datos dado un modelo. El algoritmo del máximo de verosimilitud encuentra los valores de los parámetros que maximizan la función de verosimilitud L. Al igual que los algoritmos de mínimos cuadrados la estimación se la realiza de manera iterativa. (Makridakis S., 1998).

EL algoritmo del máximo de verosimilitud se basa en el principio de verosimilitud. Este principio dice que (dado que el modelo asumido es correcto) todo lo que los datos tienen que decirnos acerca de los parámetros está contenido en la función de verosimilitud, todos los otros aspectos de los datos son irrelevantes. (Box G., 2008 4th Edition).

Aunque existe gran controversia sobre que algoritmo produce los mejores resultados y bajo cuales circunstancias, mínimos cuadrados condicionales generalmente da una mejor ejecución que el máximo de verosimilitud con conjuntos de datos pequeños.

Para grandes cantidades de datos, mínimos cuadrados condicionales es más rápido que el máximo de verosimilitud, pero el máximo de verosimilitud es más preciso. (Yaffee, 2000).

En general, si el conjunto de datos es pequeño, es aconsejable evitar la estimación mediante el máximo de verosimilitud. Pero si la estimación de parámetros está cerca de los límites de estacionariedad o estabilidad o si modelos estacionales multiplicativos son estimados, entonces mínimos cuadrados no condicionales podrían producir mejores resultados. (Yaffee, 2000).

5.2.1.8.2.1 Resultados de la Etapa de Estimación:

Se ha visto anteriormente que las características de un buen modelo son: Tiene parsimonia (cumple con el principio de parsimonia), es estacionario, es invertible, tiene coeficientes de alta calidad, sus residuos son estadísticamente independientes, se ajusta a los datos disponibles satisfactoriamente y produce pronósticos suficientemente precisos.

En esta etapa se verifican las características de: estacionariedad, invertibilidad, coeficientes de alta calidad y ajuste de datos satisfactoriamente. En la siguiente etapa (diagnóstico – chequeo) se verificará si sus residuos son estadísticamente independientes. La precisión de los pronósticos se evaluará utilizando el modelo para producir predicciones reales. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

El principio de parsimonia, primordial en la metodología de Box – Jenkins, será analizado en cada uno de los ejemplos del presente trabajo.

Se considerará el siguiente ejemplo para analizar las características arriba mencionadas. Tomado de la referencia (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Sea el modelo ARMA(1,1) y su media $\bar{Z}$ = 99.44,

$$(1 - 0.908B)\tilde{Z}_t = (1 - 0.605B)\hat{a}_t \qquad (5.49)$$

En la figura 5.12 se muestra un ejemplo de los resultados de una simulación, con el fin de explicar las características principales para un buen modelo.

Coeficiente	Estimado	Std.Error	Estadístico t
Phi	0.908	0.07	13.04
Theta	0.605	0.145	4.18
Constante	9.15733		
RMSE Ajustado:	0.92857	MAPE:	0.71
Correlaciones:			
	1	2	
1	1.00		
2	0.68	1.00	

Figura 5.12 Resultados Estimación Ejemplo ARMA(1,1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:201

Invertibilidad y Estacionariedad

Dado el modelo en función del operador de retardo B en (5.49), se puede calcular el valor de la constante C de este modelo por medio de la fórmula dada en (5.35). Por lo tanto $\hat{C} = \hat{\mu}(1 - \widehat{\emptyset}_1) = 99.44(1 - 0.908) = 9.15$, como se puede ver en la figura 5.12.

Los coeficientes estimados nos sirven para chequear estacionariedad e invertibilidad.

La condición de estacionariedad se chequea en la porción autoregresiva del modelo ARMA(1,1) así $|\emptyset_1| < 1$. Ya que $\emptyset_1$ es desconocido se examina $\widehat{\emptyset}_1$ estimado en su lugar. Como se puede observar en la Figura 5.12 esta condición se satisface ya que $|\widehat{\emptyset_1}| = 0.908 < 1$. Se debe tener cuidado con este coeficiente ya que es bastante cercano a 1. Esto hace al operador AR de este modelo $(1 - 0.908B)$ casi idéntico al operador de diferencia $(1 - B)$. Este es un ejemplo de cómo los resultados de la estimación nos dan pistas de la estacionariedad y como un modelo podría ser reformulado.

Los requerimientos de invertibilidad se aplican solo a la parte de medias móviles MA del modelo. Los requerimientos de un ARIMA (1,1) son los mismos que un MA(1): $|\theta_1| < 1$. El coeficiente $\hat{\theta}_1 = 0.605$ estimado como se muestra en la figura 5.12 satisface plenamente este requerimiento.

Calidad de los Coeficientes: Significancia Estadística

Como se indica en la figura 5.12 cada coeficiente estimado tiene su error estándar y su valor t, ya que es una estadística basada en una muestra. Cada coeficiente estimado tiene una distribución muestral con su error estándar el mismo que es estimado por el programa estadístico. La mayoría de programas automáticamente realizan la prueba de hipótesis de que el verdadero coeficiente es igual a cero. El valor t aproximado para probar la hipótesis de cada coeficiente es calculado por:

$$\frac{(Coeficiente\ Estimado) - (Valor\ del\ Coeficiente\ para\ la\ Hipótesis)}{Error\ estándar\ del\ coeficiente\ estimado}$$

En el presente ejemplo sería:

$$t_{\hat{\emptyset}_1} = \frac{0.908 - 0}{0.070} = 13.04$$

y

$$t_{\hat{\theta}_1} = \frac{0.605 - 0}{0.145} = 4.18$$

Como regla práctica se excluirá cualquier coeficiente con un valor $t\ absoluto$ menor a 2.0. Cualquier coeficiente cuyo valor absoluto de t es 2.0 o mayor será significativamente diferente de cero a un nivel aproximado del 5%. Incluir coeficientes con valores $t\ absolutos$ menores a 2.0 tiende a producir modelos sin parsimonia y pronósticos menos precisos. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Calidad de los Coeficientes: Matriz de Correlación

La mayoría de programas muestran la correlación entre los coeficientes estimados. No se puede evitar obtener estimados que estén correlacionados, pero una alta correlación entre los coeficientes estimados sugiere una baja calidad. Si los coeficientes estimados tienen una alta correlación los coeficientes estimados finales dependen fuertemente de una realización en particular; una realización ligeramente diferente podría producir coeficientes estimados muy diferentes. Si realizaciones diferentes del mismo proceso podrían producir coeficientes estimados muy diferentes, estas estimaciones son de baja calidad. Bajo estas condiciones, estimados basados en una realización dada podrían ser inapropiados para períodos de tiempo futuro a menos que observaciones futuras se comporten de una manera muy similar a una realización dada. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Como regla práctica, se pensará que los coeficientes estimados serán algo inestables cuando el coeficiente de correlación entre dos coeficientes ARIMA estimados es mayor a 0.9. Si esto pasa se debería considerar algún modelo alternativo basado en la función acf estimada y pacf. Una de estas alternativas podría producir un ajuste adecuado con parámetros estimados más estables. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

En la figura 5.12 se muestra que la correlación entre $\widehat{\emptyset}_1 \; y \; \hat{\theta}_1$ es de 0.68, por lo tanto el modelo estimado es satisfactorio respecto a la correlación.

Calidad de los Coeficientes: Redundancia en los coeficientes

Los modelos ARIMA presentan a veces un problema conocido como coeficientes redundantes. Un modelo con coeficientes redundantes presenta dos problemas: Tiende a no cumplir el principio de parsimonia y es difícil realizar una buena estimación. Los coeficientes estimados normalmente son de baja calidad.

Para aclarar la idea de coeficientes redundantes se presenta el siguiente ejemplo: Considere el proceso ARMA(1,1) con $\emptyset_1 = 0.6 \; y \; \theta_1 = 0.6$:

$$(1 - 0.6B)\tilde{Z}_t = (1 - 0.6B)a_t \qquad (5.50)$$

Teóricamente no hay nada inaceptable en el proceso (5.50), pero note que el operador AR en el lado izquierdo cancela exactamente al operador MA en el lado derecho dejando a $\tilde{Z}_t$ como ruido blanco, $\tilde{Z}_t = a_t$. Los parámetros $\emptyset_1 \, y \, \theta_1$ son perfectamente redundantes. No se puede distinguir a (5.50) como un proceso de ruido blanco basado en la función acf estimada, ya que con una muestra suficientemente grande la función acf estimada nos dará evidencia de la existencia de un proceso ARMA(1,1). Además los resultados de la estimación en este caso son muy inestables, dependiendo en gran medida de esta realización en particular. Además se podría estar mejor con un modelo que cumple el principio de parsimonia como $\tilde{Z}_t = a_t$ que con un modelo sin parsimonia ARMA (1,1) con coeficientes estimados de baja calidad. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

El modelo estimado de la figura 5.12 no parece sufrir de coeficientes redundantes. El operador AR $(1 - 0.908B)\tilde{Z}_t$ no está muy cerca para cancelar al operador MA $(1 - 0.605B)\hat{a}_t$.

Como regla práctica, se debería construir modelos mixtos con mucho cuidado, evitando incluir términos AR y MA sin una evidencia sólido de su necesidad y chequear por coeficientes redundantes. Esto ayudará a producir mejores pronósticos evitando modelos sin parsimonia con coeficientes estimados inestables.

Cercanía del Ajuste: Raíz del error cuadrático medio (RMSE)

No existe garantía de que un modelo ARIMA apropiadamente construido ajuste los datos disponibles correctamente. A veces los datos tienen una gran cantidad de ruido (estadístico) que no puede ser removido con los términos AR o MA. Es decir la varianza de los residuos puede ser grande.

Ya que no se puede observar los residuos no se puede medir su varianza. Pero se tiene los residuos estimados $\widehat{(a_t)}$ y se puede estimar su varianza mediante la fórmula:

$$\hat{\sigma}_a^2 = \frac{\sum \hat{a}_t^2}{(n-m)} \qquad (5.51)$$

Donde la sumatoria es para los n residuos al cuadrado disponibles y m es el número de parámetros estimados. Restando $m\ de\ n$ se ajusta $\hat{\sigma}_a^2$ a los grados de libertad.

La raíz cuadrada de $\hat{\sigma}_a^2$ se interpreta como la desviación estándar estimada de los residuos. En la figura 5.12 se puede ver como: RMSE Ajustado y su valor es 0.92857.

RMSE Ajustado es muy útil para comparar modelos estimados de una misma realización. Dos o más modelos pueden dar resultados similares en la mayoría de aspectos. Pero si un modelo tiene un menor RMSE ajustado, se prefiere este ya que ajusta los datos mejor y producirá una menor varianza en el error de pronóstico.

Cercanía del Ajuste: Error porcentual medio absoluto (MAPE)

Otra medida de cuan bien un modelo ajusta los datos es el error porcentual medio absoluto (MAPE). Si el residuo es dividido para su correspondiente valor observado se tiene un residuo porcentual. El MAPE es simplemente la media de los valores absolutos de este residuo porcentual:

$$MAPE = \frac{100}{n}\sum\left|\frac{\hat{a}_t}{Z_t}\right| \qquad (5.52)$$

Aplicando está fórmula al ejemplo de la figura 5.12 nos da un MAPE=0.71%.

El MAPE es muy utilizado por gerentes y usuarios no técnicos para tener una idea aproximada de la precisión de un modelo. La forma preferida para dar una idea de la precisión de un pronóstico es definir intervalos de confianza para dicho pronóstico. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Criterios de Selección de Modelos

Una vez terminada la fase de verificación se pueden tener varios modelos válidos. Varios criterios se han propuesto para seleccionar modelos de series de tiempo así: Akaike (1969, 1974) con su trabajo inicial (AIC Akaike's Information Criterion), Schwarz (1978) y Akaike(1979) con su criterio de información bayesiana (BIC por sus siglas en inglés), los modelos de penalidad de Hannan y Quinn (1979), el criterio de mínimos cuadrado predictivo de Rissanen (1986), ampliado por Lai y Lee (1977)

y el AIC modificado de Hurvich - Tsai (1989) y Cavanaugh – Shumway (1977). (Tiao G., 2001).

Se escoge entonces el modelo que tenga el valor numérico más pequeño de los criterios.

El criterio de Akaike (AIC) tiene la siguiente forma general:

$$AIC = -2(\ln función\ de\ verosimilitud\ maximizada) + 2(número\ de\ parámetros)$$

Donde el primer término de AIC mide la bondad del ajuste del modelo, mientras el segundo término llamado *función de penalidad,* penaliza al modelo candidato por el número de parámetros utilizados. (Tsay, 2010 3rd Edition).

Para un modelo ARMA se reduce a:

$$AIC = log\hat{\sigma}^2 + \frac{2(p+q}{n}) \qquad (5.53)$$

Donde $(p+q)$ es el número de parámetros del modelo.

El problema del AIC es que tiende a sobreestimar el número de parámetros aunque asintóticamente. Para resolver este problema, Akaike (1979) propuso una modificación de este criterio y lo llamó BIC y es equivalente, para muestras grandes, al criterio de Schwarz. (Tiao G., 2001).

El criterio de Schwarz (1978) presenta la forma Bayesiana de estimar la dimensión de un modelo. La forma general de este criterio es:

$$BIC = -2(\log función\ de\ verosimilitud\ maximizada) + (\ln n)\ (número\ de\ parámetros)$$

Para un modelo ARMA se reduce a:

$$BIC = log\hat{\sigma}^2 + (\log n)\frac{(p+q)}{n} \qquad (5.54)$$

En este criterio la penalidad por introducir nuevos parámetros es mayor que AIC así BIC tiende a seleccionar modelos más simples que los elegidos por AIC. La diferencia entre los dos criterios puede ser muy grande si n es grande. Se puede demostrar que el criterio BIC es asintóticamente consistente bajo condiciones generales. (Tiao G., 2001).

Finalmente el ***criterio de Hannan – Quinn*** (1979) tiene la siguiente forma para un modelo ARMA:

$$\varphi = log\hat{\sigma}^2 + c\log[(\log n)]\frac{(p+q)}{n} \quad con\ c > 2 \qquad (5.55)$$

(Capa, Un Primer Curso en Series Temporales, 2008).

El primero de estos criterios es el más utilizado. Aunque los dos restantes son convergentes y conducen a una elección asintóticamente correcta del modelo. (Capa, Un Primer Curso en Series Temporales, 2008).
Estos criterios de selección de modelos se utilizarán mediante el programa EViews 9 en el presente trabajo, más adelante.

5.2.1.8.3 Diagnóstico - Chequeo

En esta etapa se decide si el modelo estimado es estadísticamente adecuado. Esta etapa está relacionada con la etapa de identificación por dos razones importantes: La primera, cuando en el diagnóstico – chequeo se descubre un modelo inadecuado, se regresa a la etapa de identificación para tentativamente seleccionar otro u otros modelos. Segunda, el diagnóstico – chequeo nos da pistas acerca de cómo un modelo inadecuado puede ser reformulado.
La prueba más importante de que un modelo ARIMA es adecuado radica en asumir que sus residuos son independientes. Con la función acf de los residuos se puede demostrar si estos son o no independientes. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.3.1 Diagnóstico - Chequeo Aplicado a los Residuos

Un modelo es estadísticamente adecuado si sus residuos son estadísticamente independientes, lo que significa no autocorrelacionados. En la práctica no se puede observar los residuos (a_t), pero tenemos los residuos $(\hat{a}_t)$ calculados del modelo estimado. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).
Un primer paso indispensable en este proceso es inspeccionar visualmente un gráfico de los residuos. A medida que el tamaño de la serie se incrementa los residuos estimados $(\hat{a}_t)$ se vuelven cercanos al ruido blanco (a_t). Por lo tanto se espera que el estudio de los $(\hat{a}_t)$ estimados podrían indicar la existencia y la naturaleza de un modelo no adecuado. En particular, patrones reconocibles de la función de autocorrelación estimada de los $(\hat{a}_t)$ podrían sugerir modificaciones apropiadas en el modelo. (Box G., 2008 4th Edition).

Hay una razón práctica para preocuparse de la independencia de los residuos que es la siguiente: Los residuos son una componente de la variable a modelar Z_t. Si los residuos están serialmente correlacionados, entonces existe un patrón autocorrelacionado en Z_t que no ha sido tomado en cuenta por los términos AR y MA del modelo. La idea completa de la modelación ARIMA es tomar en cuenta cualquier patrón de autocorrelación en Z_t con una combinación de términos AR y MA con el menor número de parámetros (parsimonia), dejando así los residuos como ruido blanco. Si los residuos están autocorrelacionados ellos no serán ruido blanco, se debe buscar otro modelo con residuos que sean independientes. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.3.1.1 Función de Autocorrelación de los Residuos (acf)

Esta función es una herramienta analítica básica para esta etapa. La función acf de los residuos es básicamente la misma que cualquier otra función acf. La única diferencia es que se utilizan los residuos $(\hat{a}_t)$ de un modelo estimado en lugar de las observaciones de una realización (Z_t) para calcular los coeficientes de autocorrelación.
Al igual que (5.9) la fórmula para calcular la función acf de los residuos es:

$$r_k(\hat{a}) = \frac{\sum_{t=1}^{n-k}(\hat{a}_t-\bar{a})(\hat{a}_{t+k}-\bar{a})}{\sum_{t=1}^{n}(\hat{a}_t-\bar{a})^2} \qquad (5.56)$$

La idea detrás del uso de la función acf de los residuos es la siguiente: Si un modelo estimado está bien formulado, entonces sus residuos (a_t) deberían ser no correlacionados. Si los residuos son no correlacionados, entonces sus estimados $(\hat{a}_t)$ deberían también ser no correlacionados en promedio. Por lo tanto la función acf de los residuos de un modelo ARIMA apropiadamente construido tendrá idealmente coeficientes de autocorrelación que sean estadísticamente igual a cero. No se puede esperar que todos los coeficientes de autocorrelación residuales sean exactamente cero, aunque el modelo haya sido construido apropiadamente. La razón es que los residuos se calculan de una realización (no del proceso) utilizando solamente estimados de los coeficientes ARIMA (no sus valores reales). Por lo

tanto se espera que el *error de muestreo* cause alguna autocorrelación residual diferente de cero aunque el modelo encontrado sea bueno.

Prueba t

Una vez calculados y graficados los coeficientes de autocorrelación residuales, es importante determinar si cada uno es significativamente diferente de cero. Se puede utilizar la fórmula aproximada de Barlett (5.19) para estimar el error estándar de la autocorrelación residual. La fórmula quedaría:

$$s(r_k(\hat{a})) = \frac{(1+2\sum_{j=1}^{k-1}(r_j^2\hat{a}))^{1/2}}{n^{1/2}} \qquad (5.57)$$

Una vez estimado el error estándar $s(r_k(\hat{a}))$ mediante (5.57), se puede probar la hipótesis nula H_0: $\rho_k(a) = 0$ para cada coeficiente de autocorrelación residual. Cabe destacar que no se tiene los valore de $\rho_k(a)$ disponibles pero se tienen sus estimados $r_k(\hat{a})$. Se prueba la hipótesis nula calculando a cuantos errores estándar (t) desde cero caen los coeficientes de autocorrelación residual, mediante:

$$t = \frac{r_k(\hat{a})-0}{s(r_k(\hat{a}))} \qquad (5.58)$$

En la práctica, si el valor absoluto de t es menor que 1.25 (aproximadamente) para los retardos 1,2 y 3, y menor a 1.6 para retardos mayores, se puede concluir que los residuos en esos retardos son independientes. Si el valor de t es mayor que los valores críticos sugeridos arriba, se rechaza la hipótesis nula y se concluye que los residuos del modelo estimado están correlacionados y que el modelo estimado es inadecuado. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Prueba de Portmanteau

En lugar de considerar los valores $r_k(\hat{a})$ individualmente, un indicador es a menudo necesario para saber si por ejemplo las primeras 15 o 20 autocorrelaciones residuales $(\hat{a}_t)$ tomadas como un todo indican que un modelo es inadecuado.

Si se tienen K autocorrelaciones residuales. Se prueba la siguiente hipótesis nula H_0 conjunta acerca de las correlaciones entre los residuos

$$H_0: \rho_1(a) = \rho_2(a) = \rho_3(a) = \cdots = \rho_k(a) = 0 \quad (5.59)$$

Mediante el siguiente estadístico Q^* de Ljung – Box mostrado antes:

$$Q^*(h) = n(n+2)\sum_{k=1}^{h} \frac{r_k^2(\hat{a})}{(n-k)} \quad (5.60)$$

Donde n es el número de observaciones del modelo estimado. El estadístico $Q^*(h)$ sigue aproximadamente la distribución chi-cuadrada con $(K-m)$ grados de libertad, donde m es el número de parámetros estimados del modelo ARIMA.

Si Q^* es grande (significativamente diferente de cero) nos indica que la autocorrelación residual como un todo es significativamente diferente de cero y los residuos del modelo estimado están probablemente autocorrelacionados. Se debe considerar reformular el modelo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

La mayoría de programas estadísticos calculan el valor de p (p value) de $Q^*(h)$. La regla de decisión práctica es entonces rechazar H_0 si el valor de p es menor o igual a α, el nivel de significancia. (Tsay, 2010 3rd Edition).

En la práctica, la elección de h podría afectar el desempeño del estadístico $Q^*(h)$. Varios valores de h se utilizan a menudo. Estudios de simulación sugieren la elección de $h = Ln(n)$ proporciona la mejor ejecución. Cabe destacar que esta regla debe ser modificada cuando se analice series de tiempo estacionales donde las autocorrelaciones de retardos múltiplos de la estacionalidad son los más importantes. (Tsay, 2010 3rd Edition).

En lugar del estadístico de Portmanteau basado en la autocorrelación de los residuos, como en (5.53) se podría basar alternativamente en los residuos de la autocorrelación parcial. Si el modelo ajustado es adecuado, el error asociado al proceso a_t será ruido blanco y se debería esperar que la autocorrelación parcial de los residuos a cualquier retardo k, que se representa por $\widehat{\emptyset}_{kk}(\hat{a})$, no debe ser significativamente diferente de cero. Por lo tanto, la prueba de si el modelo es adecuado puede basarse en el siguiente estadístico:

$$\tilde{Q} = n(n+2)\sum_{k=1}^{h} \frac{[\widehat{\emptyset}_{kk}(\hat{a})]^2}{(n-k)} \quad (5.61)$$

El estadístico $\tilde{Q}$ sigue aproximadamente la distribución chi-cuadrada con $(K - m)$ grados de libertad, donde m es el número de parámetros estimados del modelo ARIMA. (Box G., 2008 4th Edition).

Si $\tilde{Q}$ es grande los residuos del modelo estimado están probablemente autocorrelacionados. Entonces se debe reformular el modelo.

5.2.1.8.3.1.2 Otros Diagnósticos

Gráfico de los Residuos

Un análisis visual de los residuos ayuda a detectar problemas con el modelo ajustado.

Por ejemplo, los residuos podrían mostrar una varianza cambiante con el tiempo, sugiriendo una transformación logarítmica (u otra) de los datos originales. Es mucho más fácil ver un cambio en la varianza en el gráfico de los residuos que en el gráfico de los datos originales.

El gráfico de los residuos pude ser útil en la detección de datos erróneos o eventos inusuales (outliers) que impactan la serie de tiempo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Sobreajuste (Overfitting)

Un método muy útil para chequear un modelo es el *sobreajuste*, esto es, estimar los parámetros de un modelo algo más general que lo que se piensa será el modelo real. Este método asume que se conoce en qué dirección el modelo probablemente no es adecuado. En otras palabras se necesita descubrir de qué manera el modelo no es adecuado para sugerir una modificación apropiada. (Box G., 2008 4th Edition).

El sobreajuste se justifica especialmente si las funciones acf y pacf estimadas son ambiguas. Es decir estas funciones nos pueden dar pistas acerca de la dirección hacia donde expandir el modelo.

Se debe tener en cuenta una precaución especial: Ser cuidadoso con el sobreajuste y no adicionar a los dos lados del modelo. Es decir no sobreajustar a los términos

AR y MA simultáneamente, ya que haciendo esto no solo va en contra del principio de parsimonia sino que también se producen problemas serios de estimación debido a la redundancia de los coeficientes. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Modelo no adecuado debido al cambio en los valores de los parámetros

Una forma interesante de un modelo no adecuado ocurre cuando la *forma* del modelo permanece igual pero los parámetros cambian durante un período de tiempo prolongado. (Box G., 2008 4th Edition).

Una forma de chequear este problema en un modelo es dividir los datos en dos por ejemplo y estimar el mismo modelo para cada mitad. Luego ejecutar una prueba estadística para ver si los parámetros de los dos conjuntos de datos son significativamente diferentes.

Existe otra forma menos formal para chequear si los coeficientes cambian. Se recorta una parte anterior de la realización (Por Ejemplo el último 10% de las observaciones) y se reestima el mismo modelo para la realización acortada. Si los coeficientes son cercanos a los estimados usando la realización completa (por ejemplo con (± 0.1) se puede concluir que las observaciones más recientes han sido generadas por el mismo proceso que los datos anteriores. Este enfoque aunque es informal tiene la ventaja que enfatiza al pasado más reciente. Si los datos más recientes se comportan de diferente manera que el resto de la realización, surge una seria duda acerca de la predicción del modelo en un futuro cercano. La desventaja de esta prueba es que es informal. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.3.1.3 Reformulación del Modelo

Se supone que un modelo es estadísticamente no adecuado debido a: (i) los valores del estadístico t exceden los valores críticos sugeridos anteriormente o (ii) cualquiera de los coeficientes de la prueba de portmanteau es demasido grande. Acorde con la metodología de Box-Jenkins se debe regresar a la etapa de identificación para tentativamente seleccionar otro u otros modelos. No existe

garantía de encontrar un mejor modelo ya que las autocorrelaciones residuales del modelo original pueden ser significativas solamente por un error de muestreo.

Una forma de reformular un modelo aparentemente no adecuado es reexaminando las funciones sacf y spacf calculadas de la realización original. Reexaminando las funciones originales sacf y spacf podrían sugerir uno o más modelos alternativos que no eran obvios al inicio. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Otra forma de reformular un modelo es utilizar la función acf residual como guía.

Por ejemplo, se supone que la función sacf original decrece hacia cero, por tal razón escogemos un modelo AR(1) inicialmente de la forma:

$$(1 - \emptyset_1 B)\tilde{Z}_t = b_t \qquad (5.62)$$

Pero b_t resulta estar correlacionada. Se supone además que el acf residual para (5.62) tiene un solo pico en el retardo 1. Esto sugiere un modelo MA(1) para b_t igual a:

$$b_t = (1 - \theta_1 B)a_t \qquad (5.63)$$

Donde a_t resulta no estar autocorrelacionado. Utilizando (5.63) para sustituir en (5.62) el modelo ARMA(1,1) resultante para Z_t es:

$$(1 - \emptyset_1 B)\tilde{Z}_t = (1 - \theta_1 B)a_t \qquad (5.64)$$

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

El modelo (5.64) dará mejores resultados que (5.62).

No siempre la modificación de un modelo es tan directa como en el ejemplo anterior. Es decir el modelo resultante (5.64) es una composición obvia del modelo inicial de Z_t y el modelo de los residuos $\hat{b}_t$. Se ha reformulado el modelo por simple adición de un coeficiente al modelo original determinado mediante la sacf residual. Sin embargo en otros casos la información de la función scaf residual no es tan clara como en el ejemplo anterior.

Así, se supone que el modelo inicial es un AR(1):

$$\left(1 - \acute{\emptyset}_1 B\right)\tilde{Z}_t = b_t \qquad (5.65)$$

Donde b_t resulta estar autocorrelacionado. Pero se supone que la función residual sacf sugiere un modelo AR(1) para b_t esta vez:

$$(1 - \emptyset_1^* B)b_t = a_t \qquad (5.66)$$

Donde a_t ya no está autocorrelacionado.

En este caso ya no se puede adicionar un coeficiente al modelo, sugerido por la función sacf residual, ya que es imposible aumentar un coeficiente AR al retardo 1 cuando ya existe este coeficiente en el modelo. Sin embargo se puede utilizar el mismo procedimiento algebraico utilizado en el ejemplo anterior. Resolviendo (5.66) para obtener b_t:

$b_t = (1 - \emptyset_1^* B)^{-1} a_t$ (5.67)

Se Sustituye (5.67) en (5.65) se tiene:

$(1 - \acute{\emptyset}_1 B)\tilde{Z}_t = (1 - \emptyset_1^* B)^{-1} a_t$ (5.68)

Luego se multiplica ambos lados por $(1 - \emptyset_1^* B)$ y se opera dando como resultado un modelo AR(2) como:

$(1 - \emptyset_1 B - \emptyset_2 B^2)\tilde{Z} = a_t$ (5.69)

Donde $\emptyset_1 = \acute{\emptyset}_1 + \emptyset_1^*$ y $\emptyset_2 = -\acute{\emptyset}_1 \emptyset_1^*$.

Este ejemplo muestra que la información contenida en la función sacf residual puede ser sutil. En particular no es apropiado aumentar un coeficiente al modelo inicial tal como sugiere la sacf residual. Además muestra que la scaf residual puede ser muy importante para encontrar un modelo adecuado.

5.2.1.8.4 Pronósticos

Un modelo ARIMA correcto da un pronóstico con el menor error medio cuadrático, para modelos de una variable (una sola serie) con coeficientes fijos.

Para cada período de tiempo se produce un solo valor del pronóstico llamado *pronóstico puntual*. Además se construye un intervalo de confianza alrededor de cada punto del pronóstico para obtener un *intervalo del pronóstico*. El intervalo del pronóstico es muy útil ya que nos da la posibilidad de asociar un error al pronóstico puntual. Más adelante se mostrará que con la desviación estándar del error del pronóstico se crean intervalos de confianza alrededor del pronóstico puntual. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

En esta sección se asumirá, por simplicidad, que cualquier modelo ARIMA a considerar es conocido, es decir se conocen su media μ, todos sus coeficientes $\emptyset s$, θs y todos sus residuos pasados a_s.

Afortunadamente esta suposición en la práctica es esencialmente correcta si se ha identificado y estimado un modelo ARIMA con el suficiente número de observaciones ya que las propiedades del pronóstico ARIMA son poco afectadas por el ordinario error de muestreo cuando el tamaño de la muestra es el apropiado. Box, Jenkins y Reinsel en su libro (Box G., 2008 4th Edition) (pp: 103-113) dan tres formas explícitas de expresar un modelo ARIMA:

1- Por ecuaciones de diferencias es decir en términos de los valores anteriores de Z_s y valores anteriores y actual de sus residuos a_s,

2- En términos de valores anteriores y actual de los residuos a_s solamente y

3- En términos de la suma ponderada de los valores previos Z_{t-j} del proceso y valor actual del residuo a_t.

En esta sección se utilizarán las dos primeras formas para calcular el pronóstico puntual y su intervalo del pronóstico.

Modelo en forma de ecuación de diferencias

La manera más conveniente de calcular pronósticos *puntuales* mediante un modelo ARIMA es expresándolo con una ecuación de diferencias.

Sea t el período de tiempo actual. Cuando se pronostica son de interés los valores futuros de una serie de tiempo, representados por Z_{t+l} donde $l \geq 1$. El período t es denominado *origen* del pronóstico y l es llamado el *tiempo de espera* (del inglés *lead time*) del pronóstico. El conjunto de observaciones disponibles $(Z_t, Z_{t-1}, \dots)$ se designará como I_t. El pronóstico de Z_{t+l}, designado por $\hat{Z}_t(l)$ es la esperanza condicional de Z_{t+l}. Esto es, $\hat{Z}_t(l)$ es la esperanza matemática de Z_{t+l} dado I_t:

$$\hat{Z}_t(l) = E(Z_{t+l}|I_t) \qquad (5.70)$$

Como una ilustración, se considera un modelo ARIMA (1,0,1) y se desarrollará en forma manual los primeros pronóstico obtenidos a partir del modelo. La mayoría de programas estadísticos para identificar y estimar los modelos ARIMA tienen la opción de generar pronósticos de un modelo estimado, así normalmente el pronóstico no se calcula a mano.

Sea el modelo ARIMA (1,0,1)

$$(1 - \emptyset_1 B)\tilde{Z}_t = (1 - \theta_1 B)a_t$$

o

$$Z_t = \mu(1 - \emptyset_1) + \emptyset_1 Z_{t-1} - \theta_1 a_{t-1} + a_t \quad (5.71)$$

Sea $l = 1$ utilizando (5.71) se obtiene Z_{t+1}:

$$Z_{t+1} = \mu(1 - \emptyset_1) + \emptyset_1 Z_t - \theta_1 a_t + a_{t+1} \quad (5.72)$$

Aplicando (5.71) a (5.72) se obtiene el pronóstico de Z_{t+1}:

$$\hat{Z}_t(1) = E(Z_{t+1}|I_t)$$

$$\hat{Z}_t(1) = \mu(1 - \emptyset_1) + \emptyset_1 Z_t - \theta_1 a_t \quad (5.73)$$

Ya que a_{t+1} es desconocido al tiempo t, se le asigna el valor esperado de cero. En este ejemplo $Z_t \; y \; a_t$ constituyen I_t. Es decir $Z_t \; y \; a_t$ es toda la información relevante acerca del pasado de Z_t necesaria para el pronóstico Z_{t+1}.

Para $l = 2$:

$$\hat{Z}_t(2) = E(Z_{t+2}|I_t)$$

$$\hat{Z}_t(2) = \mu(1 - \emptyset_1) + \emptyset_1 Z_{t+1} - \theta_1 a_{t+1} \quad (5.74)$$

Ya que Z_{t+1} es desconocido al origen t se reemplaza por su esperanza condicional $\hat{Z}_t(1)$ de (5.73), al igual a_{t+1} es desconocido al origen t es reemplazado por su valor esperado de cero. Con estos dos reemplazos, (5.74) queda:

$$\hat{Z}_t(2) = \mu(1 - \emptyset_1) + \emptyset_1 \hat{Z}_t(1) \quad (5.75)$$

Siguiendo el procedimiento anterior se obtiene:

$$\hat{Z}_t(3) = \mu(1 - \emptyset_1) + \emptyset_1 \hat{Z}_t(2)$$

$$\hat{Z}_t(4) = \mu(1 - \emptyset_1) + \emptyset_1 \hat{Z}_t(3)$$

$$\hat{Z}_t(5) = \mu(1 - \emptyset_1) + \emptyset_1 \hat{Z}_t(4)$$

Pronósticos para otros modelos ARIMA se calcular de la misma manera anterior. En la práctica μ es desconocido y es reemplazado por su estimado $\hat{\mu}$. De igual manera los coeficientes $\theta \; y \; \emptyset$ son reemplazados por sus estimados $\hat{\theta} \; y \; \widehat{\emptyset}$. Como se indicó arriba, las observaciones pasadas de Z_t se emplean cuando están disponibles. Ellas están disponibles hasta el tiempo t, el origen del pronóstico de ahí en adelante se reemplazan por sus pronósticos (valores de su esperanza condicional). Los valores pasados de a_t son reemplazados por sus correspondientes estimados, los residuos estimados $\hat{a}_t$, cuando estos residuos están disponibles. Pero cuando exceden el origen del pronóstico t estos residuos

son reemplazados por sus valores esperados de cero. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Para *modelos estacionarios* el pronóstico converge hacia *la media* de la serie. Cuán rápido o lento esto suceda depende del modelo y de que tan cerca estén las observaciones más recientes de la media. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

En términos generales pronósticos de modelos MA convergen más rápido hacia la media, ya que pierden información de los residuos pasados como el pronóstico se va hacia el futuro. Esto es en los modelos MA se reemplazan los residuos con el valor esperado de cero cuando exceden el origen del pronóstico.

Cuando el tiempo de espera (lead time) del pronóstico excede el valor de q, el máximo retardo de los términos MA, el pronóstico de un modelo MA puro es igual a la constante estimada $\hat{C}$, la misma que es igual a la media estimada $\hat{\mu}$, en los modelos MA puros. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

En la figura 5.13 se muestra el pronóstico de un modelo AR(1) en rojo se pueden ver sus pronósticos y sus respectivos intervalos de confianza. Se puede notar que los valores de los pronósticos convergen hacia la media del modelo (4.04).

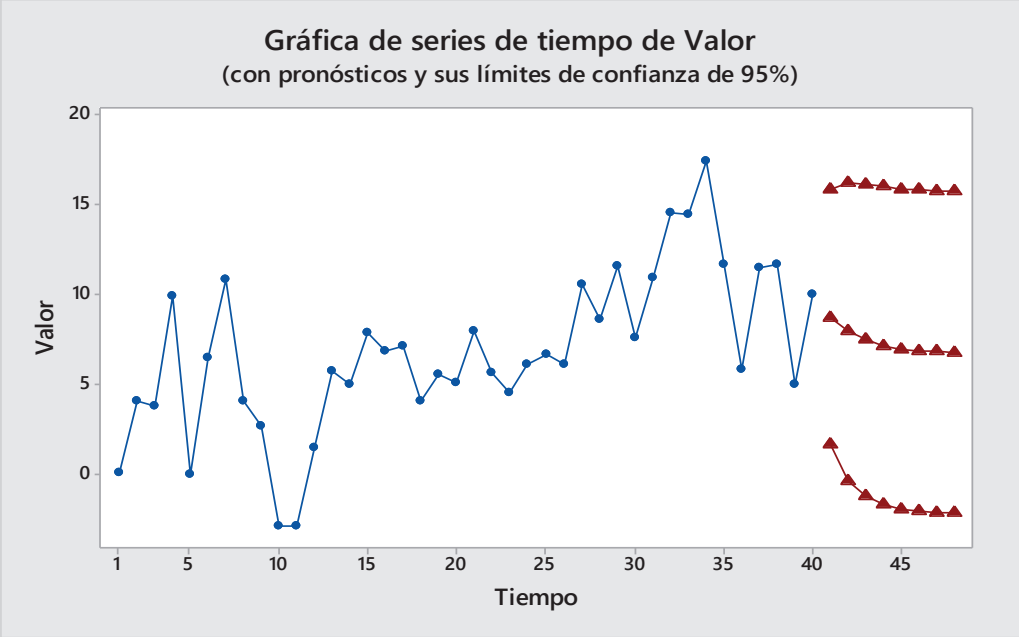


Figura 5.13 Pronóstico de un Modelo AR(1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:245

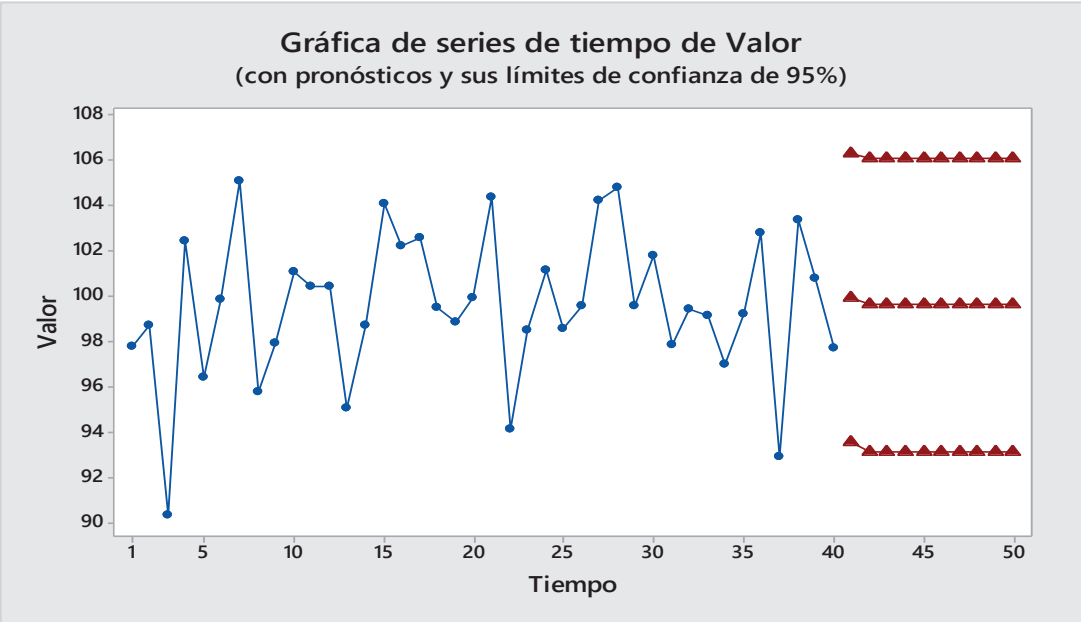


Figura 5.14 Pronóstico de un Modelo MA(1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:246

En la figura 5.14 se muestra el pronóstico producido por un modelo MA(1). Se puede notar que converge hacia la media $(\widehat{\mu} = \hat{C} = 99.83)$ al igual que el anterior pero de una manera más rápida.

En la figura 5.15 se muestra el pronóstico de un modelo ARIMA (0,1,1) no estacionario. Note que el pronóstico no converge hacia media de la serie (193.3) justamente porque el modelo no es estacionario. La diferencia $(d = 1)$ ha liberado al pronóstico de la media fija. Si la media de la serie está cambiando significativamente con el tiempo no se debe atar el pronóstico con la media general de la serie; Aunque su valor puede ser calculado, este no es útil para describir el cambio del nivel de la serie.

Finalmente en la figura 5.16 se muestra el pronóstico producido por otro modelo no estacionario, un ARIMA (0,2,1). La realización utilizada para identificar y estimar este modelo muestra cambios en el nivel y la pendiente. Como se discutió anteriormente, tal serie requiere una segunda diferencia $(d = 2)$ para inducir a una media constante. Se puede notar que el pronóstico no converge hacia la media de la realización (62.7). El pronóstico está dominado por la componente de diferencia del modelo. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

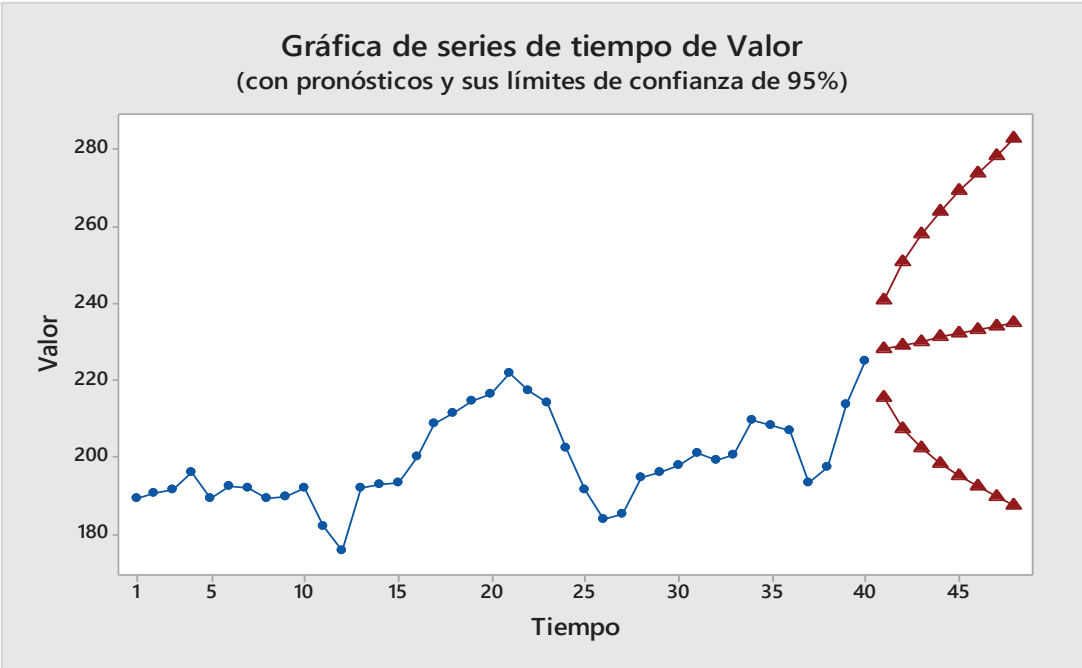


Figura 5.15 Pronóstico de un Modelo ARIMA(0,1,1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:248

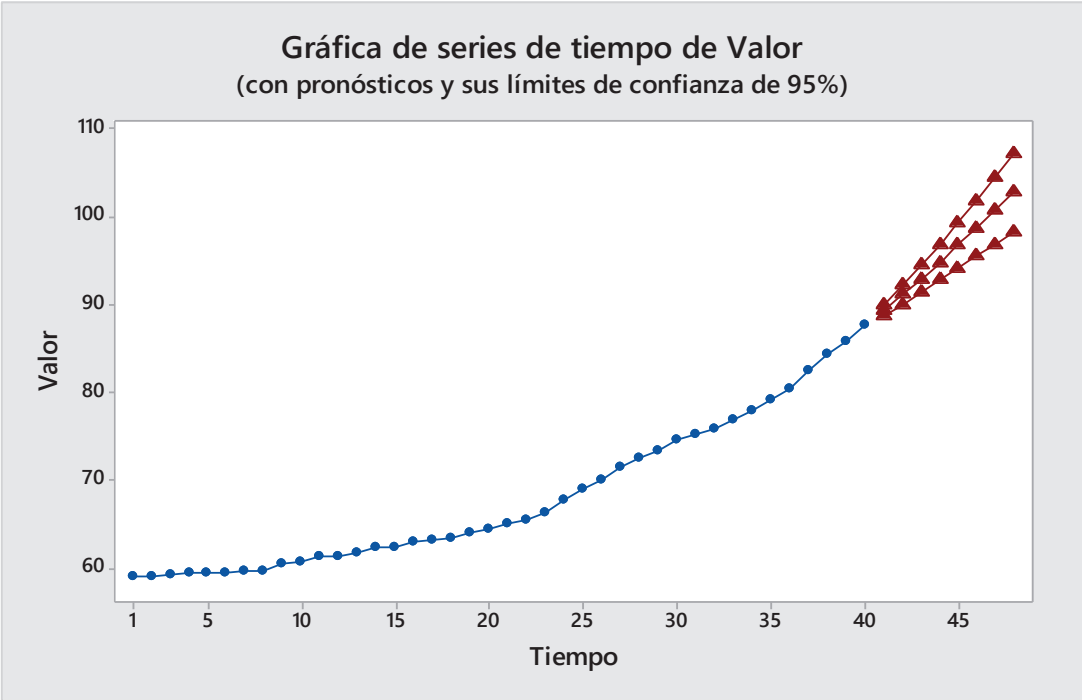


Figura 5.16 Pronóstico de un Modelo ARIMA(0,2,1)

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:249

Modelo en términos de los valores de los residuos (a_t)

Cualquier modelo ARIMA se puede expresar en función de sus residuos. Esto es se puede reemplazar los términos AR por una serie infinita de términos MA. Un modelo MA puro está ya en función de sus residuos.

Aunque esta forma no es muy conveniente para producir pronósticos, es muy útil para estimar la varianza del pronóstico y así determinar los intervalos de confianza alrededor de los pronósticos puntuales. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Los coeficientes en esta forma se representan por ψ_i donde i corresponde al retardo de tiempo asociado al residuo pasado.

$$Z_t = \mu + \psi_0 a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \psi_3 a_{t-3} + \cdots \quad (5.76)$$

Donde $\psi_0 = 1$. Si la secuencia $\psi_1, \psi_2, \psi_3, \ldots$ es finita, entonces (5.76) es un modelo MA puro. Si la secuencia es infinita (5.76) representa un modelo AR o mixto.

Para series estacionarias, μ es simplemente la media. Para series no estacionarias, μ representa el nivel cambiante de la serie determinado por la operación de diferencia.

Cualquier modelo MA puro de orden q, está ya en función de sus residuos. Por ejemplo sea un MA(2):

$$(Z - \mu) = (1 - \theta_1 B - \theta_2 B^2) a_t$$

o

$$Z_t = \mu + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} \quad (5.77)$$

Siendo $\psi_0 = 1$, $\psi_1 = -\theta_1$ y $\psi_2 = -\theta_2$, se puede escribir (5.77) en término de los residuos como:

$$Z_t = \mu + \psi_0 a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} \quad (5.78)$$

Que no es más que la versión truncada de (5.76).

Modelos AR pueden ser expresados en términos de los residuos invirtiendo y expandiendo el operador AR. Por ejemplo un modelo AR(1) se puede escribir de la forma MA como sigue:

$$(1 - \emptyset_1 B)(Z_t - \mu) = a_t$$

Dividiendo ambos lados para el operador AR queda:

$$(Z_t - \mu) = (1 - \emptyset_1 B)^{-1} a_t$$

Si $|\emptyset_1| < 1,\ (1-\emptyset_1 B)^{-1}$ es equivalente a la serie convergente

$(1-\emptyset_1 B)^{-1} = (1+\emptyset_1 B+\emptyset_1^2 B^2+\emptyset_1^3 B^3+\ldots)$

Así el modelo AR(1) se puede escribir como:

$(Z_t-\mu) = (1+\emptyset_1 B+{\emptyset_1}^2 B^2+{\emptyset_1}^3 B^3+\cdots)a_t$

o

$$Z_t = \mu + a_t + \emptyset_1 a_{t-1} + {\emptyset_1}^2 a_{t-2} + {\emptyset_1}^3 a_{t-3} \qquad (5.79)$$

Dando: $\psi_0 = 1, \psi_1 = \emptyset_1, \psi_2 = {\emptyset_1}^2, \psi_3 = {\emptyset_1}^3$ y así en adelante.

Los valores de los coeficientes para diferentes modelos ARIMA varían (excepto para ψ_0 que siempre es igual a 1) dependiendo del grado de diferencia y de los valores de los coeficientes AR y MA del modelo. Sería muy tedioso calcular a mano los valores de ψ usualmente son generados por programas estadísticos. Sin embargo se mostrará un método para calcular los valores de ψ.

Sea el modelo ARIMA :

$$\phi(B)\nabla^d \tilde{Z}_t = \theta(B)a_t \qquad (5.80)$$

Además Z_t se puede expresar como: $\tilde{Z}_t = \psi(B)a_t$ en la forma de los residuos

Reemplazando $\tilde{Z}_t = \psi(B)a_t$ en (5.80) se tiene:

$\phi(B)\nabla^d\psi(B)a_t = \theta(B)a_t$

Si se divide para a_t los dos lados quedaría:

$\phi(B)\nabla^d\psi(B) = \theta(B)$

Si se expande esta expresión se obtiene:

$(\psi_0+\psi_1 B+\psi_2 B^2+\psi_3 B^3+\cdots)\left(1-\emptyset_1 B-\emptyset_2 B^2-\cdots-\emptyset_p B^p\right)(1-B)^d.$

$= \left(1-\theta_1 B-\theta_2 B^2-\cdots-\theta_q B^q\right) \qquad (5.81)$

A partir de (5.81) se pueden obtener los valores ψ_i igualando coeficientes de potencias similares de B. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Se utilizan estos valores para construir intervalos de confianza alrededor de pronósticos puntuales.

5.2.1.8.4.1 Dispersión de Pronósticos ARIMA

La ecuación de diferencias de un modelo ARIMA puede producir pronósticos puntuales, donde "*puntual"* significa un solo valor en lugar de un rango de valores. Utilizando la forma de los residuos, se puede determinar la varianza del error de pronóstico. Esta nos permite construir intervalos de confianza aproximados alrededor del pronóstico, además provee información de que tan confiable puede ser el pronóstico.

Error de Pronóstico, varianza y desviación estándar

Primero se definirá el error de pronóstico para un origen t y tiempo de espera l, representado por $e_t(l)$, como el valor de Z observado en el período $t+l$ menos el pronóstico de Z en ese mismo período:

$$e_t(l) = Z_{t+l} - \hat{Z}_t(l) \tag{5.82}$$

Utilizando (5.76) para escribir Z_{t+l} en la forma de los residuos como:

$$Z_{t+l} = \mu + \psi_0 a_{t+l} + \psi_1 a_{t+l-1} + \psi_2 a_{t+l-2} + \psi_3 a_{t+l-3} + \cdots \tag{5.83}$$

El pronóstico correspondiente $\hat{Z}_t(l)$, que es la esperanza condicional $E(Z_{t+l}|I_t)$ se encuentra desde (5.83) y es:

$\hat{Z}_t(l) = E(Z_{t+l}|I_t)$

$$\hat{Z}_t(l) = \mu + \psi_l a_t + \psi_{l+1} a_{t-1} + \psi_{l+2} a_{t-2} + \psi_{l+3} a_{t-3} + \cdots \tag{5.84}$$

El conjunto de información I_t se define como la información de la serie Z, solamente hasta el período t.La expresión (5.84) contiene residuos solamente desde el período t o anteriores ya que cualquier residuo después del período t es desconocido al tiempo t.(se asume por simplicidad que los residuos al tiempo t o anteriores son observables. En la práctica ellos deben ser estimados en la etapa de estimación de los residuos. Residuos después del tiempo t son no solamente desconocidos al tiempo t sino que también no pueden ser estimados al tiempo t.)

Sustituyendo (5.83) y (5.84) en (5.82), se encuentra

$$e_t(l) = \psi_0 a_{t+l} + \psi_1 a_{t+l-1} + \cdots + \psi_{l-1} a_{t+1} \tag{5.85}$$

Esto es (5.83) contiene todos los residuos hasta el período $t + l$, mientras que (5.84) contiene solo aquellos hasta el período t. Restando (5.84) de (5.83) quedan los residuos desde el período $t + l$ hacia atrás hasta $t + 1$.

Ahora utilizando (5.85) se puede encontrar que la varianza (condicional) de $e_t(l)$ es:

$\sigma^2[e_t(l)] = E\{e_t(l) - E[e_t(l)]|I_t\}^2$

$\sigma^2[e_t(l)] = E[e_t(l)]^2.$

$$\sigma^2[e_t(l)] = {\sigma_a}^2(1 + \psi_1^2 + \psi_2^2 + \cdots + \psi_{l-1}^2) \quad (5.86)$$

Por lo tanto la desviación estándar de $e_t(l)$ es

$$\sigma[e_t(l)] = \sigma_a(1 + \psi_1^2 + \psi_2^2 + \cdots + \psi_{l-1}^2)^{1/2} \quad (5.87)$$

La varianza (5.86) se encuentra elevando al cuadrado (5.85). {Note en (5.86) que $E[e_t(l)] = 0$ y $\psi_0 = 1$}. Todos los términos con productos cruzados tienen un valor esperado de cero ya que se asumen son independientes.

En la práctica, $\sigma[e_t(l)]$ debe ser estimado, ya que σ_a es desconocida y es reemplazada por el RMSE ($\hat{\sigma}_a$) y ya que los coeficientes son desconocidos son reemplazados por sus estimados ($\hat{\psi}_i$) calculados a partir de los coeficientes estimados ($\phi_s \, y \, \theta_s$). Las varianzas de los errores de pronóstico resultantes (y los intervalos de confianza de los pronósticos) son por lo tanto solamente aproximados.

Intervalos de Confianza del Pronóstico

Si los residuos son normalmente distribuidos (como se asume) y se ha estimado un modelo ARIMA apropiado con una muestra suficientemente grande, los pronósticos para este modelo son aproximadamente *normalmente distribuidos*. Utilizando (5.87) se puede por lo tanto construir intervalos de confianza alrededor de cada pronóstico puntual utilizando la tabla de probabilidades para una distribución normal estándar. Así un intervalo con una confianza de aproximadamente el 95% está dado por:

$$\hat{Z}_t(l) \pm 1.96 \, \hat{\sigma}[e_t(l)] \quad (5.88)$$

Y un intervalo con una confianza de aproximadamente el 80% está dado por:

$$\hat{Z}_t(l) \pm 1.28 \, \hat{\sigma}[e_t(l)] \quad (5.89)$$

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Estos intervalos son interpretados de una manera estadística usual así para el 95% de confianza se puede decir: que se tendrá un 95% de confianza que el valor observado Z_{t+l} estará dentro del intervalo calculado.

5.2.1.8.4.2 Pronóstico de Datos en Forma Logarítmica

Cuando se analizaba la varianza no estacionaria se había dicho que si la *desviación estándar* de la serie es proporcional a su media, tomar el logaritmo natural produce una nueva serie con varianza constante. Sin embargo, lo que se necesita es el pronóstico de los datos originales en lugar de los datos logarítmicos. Es tentador aplicar el antilogaritmo al pronóstico logarítmico para obtener el pronóstico de la serie original. Pero si se hiciera esto se tendría el siguiente problema: Si los residuos de la serie logarítmica están normalmente distribuidos, entonces los residuos de la serie original (y el pronóstico de esta serie) seguirán una distribución Log – Normal.

Si representamos a la serie logarítmica por Z_t' y por Z_t a la serie original. Se puede demostrar que el pronóstico de Z_{t+l} depende del pronóstico y de la varianza del error de pronóstico de Z'_{t+l} de la siguiente forma:

$$\hat{Z}_t(l) = \exp\{\hat{Z}_t'(l) + \frac{1}{2}\sigma^2[e_t'(l)]\} \qquad (5.90)$$

Así, no se toma simplemente el antilogaritmo de $\hat{Z}_t'(l)$ para encontrar $\hat{Z}_t(l)$. En su lugar se debe tomar en cuenta la varianza del pronóstico logarítmico como se muestra en (5.90). Sin embargo los límites de confianza alrededor de $\hat{Z}_t(l)$ se encuentran tomando el antilogaritmo de los límites alrededor de $\hat{Z}_t{}'(l)$. Esto significa que el intervalo alrededor de $\hat{Z}_t(l)$ no es simétrico ya que el intervalo alrededor de $\hat{Z}_t{}'(l)$ es simétrico.

Finalmente, se debe notar que los pronósticos de Z_t' pueden ser interpretados en términos de Z_t sin aplicar el antilogaritmo ya que un valor logarítmico representa el cambio en porcentaje del correspondiente valor original. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.5 Modelos ARIMA Estacionales y Periódicos

Las series de tiempo a menudo muestran un comportamiento periódico. Una serie periódica tiene un patrón que se repite cada s períodos de tiempo, donde $s > 1$. La experiencia ha demostrado que modelos ARIMA producen buenos pronósticos de datos periódicos. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Uno de los más comunes tipos de comportamientos periódicos es la variación *estacional*.

Modelos ARIMA para series de tiempo estacionales se construyen utilizando el mismo procedimiento iterativo que el utilizado para datos no estacionales: identificación, estimación y diagnóstico – chequeo.

Con datos estacionales se debe hacer la diferencia de las observaciones por un período s es decir $Z_t - Z_{t-s}$. Además se debe prestar atención especial a los coeficientes de las funciones sacf y spacf para múltiplos del retardo s, $(s, 2s, 3s, \dots)$. Igualmente, en la etapa de estimación se obtiene un estimado de los coeficientes AR y MA que aparecen en los múltiplos del retardo s. La etapa de diagnóstico – chequeo se enfoca en los coeficientes de autocorrelación residual para los múltiples retardos de s.

Analizar series de tiempo estacionales es muy desafiante ya que la mayoría de series estacionales también tienen patrones no estacionales. Distinguir estos dos patrones para obtener una representación con parsimonia y estadísticamente adecuada de una realización puede ser difícil. Especialmente al inicio. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.5.1 Datos Periódicos

Como ejemplo de una serie periódica considere el gráfico de figura 5.17 representa datos mensuales de los kilovatios – hora utilizados en cierta región geográfica desde Enero de 1973 hasta diciembre de 1987. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

El patrón ondulado en la figura 5.17 sugiere que los datos tienen un componente estacional. La figura 5.18 muestra que los datos después de una diferencia regular $(d = 1)$ se mueven alrededor de un nivel constante, pero ciertas observaciones están regularmente hacia arriba y otras regularmente hacia abajo de esta media general. Esto es, el nivel de la serie cambia de una manera estacional. Después de una diferencia estacional $(D = 1, con\ S = 12)$, los datos ya no muestran ese obvio cambio estacional en el nivel como se muestra en la figura 5.19.

Para datos mensuales Z_t se relaciona con $Z_{t-12}, Z_{t-24}, Z_{t-36}$, y así en adelante, esto es, un mes dado es similar con el mismo mes un año antes, dos años antes, tres años antes, etc. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

Por convención, el orden de la estacionalidad es el número de estaciones en un período anual. Así picos estacionales cada cuatro meses indican un orden estacional de $S = 4$, si los datos presentan una estacionalidad mensual, entonces el orden de la estacionalidad es $S = 12$. (Yaffee, 2000).

Datos Estacionales

Los tipos más comunes de datos periódicos en economía y negocios son los datos con una variación estacional, lo que significa variación durante un año.

Se debe recalcar que los datos pueden ser periódicos pero no estacionales; esto es, los patrones repetitivos en estas series ocurren por ejemplo de semana a semana $(s = 7)$, en lugar de año a año $(s = 12)$. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

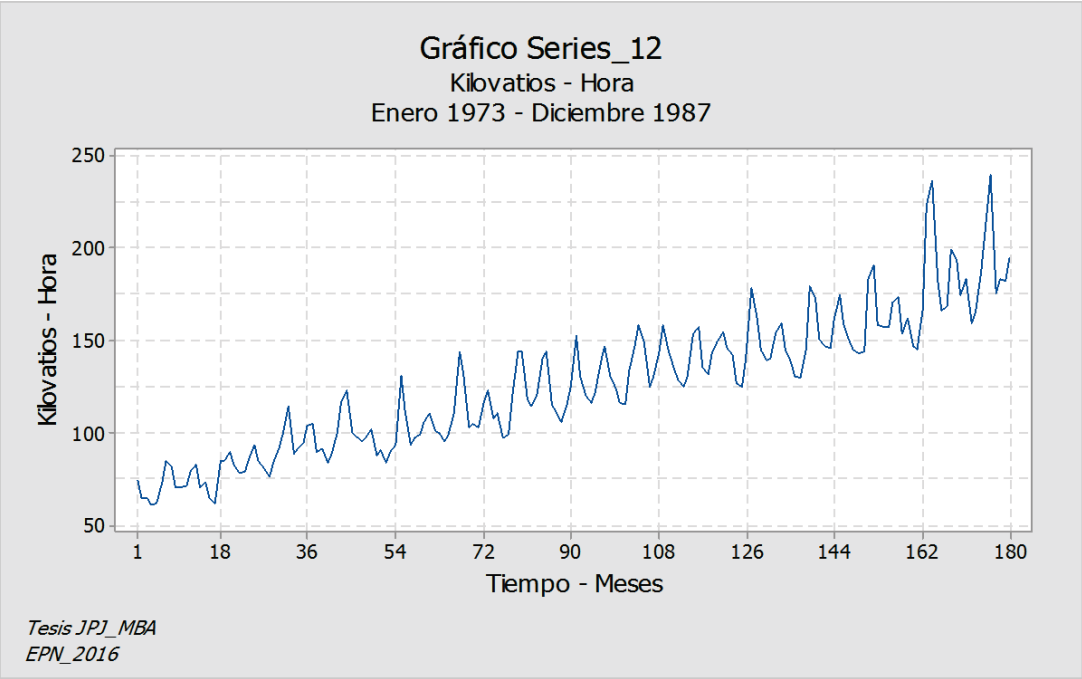


Figura 5.17 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984

(Pankratz, Forecasting with Dynamic Regression Models, 1991) pp:132

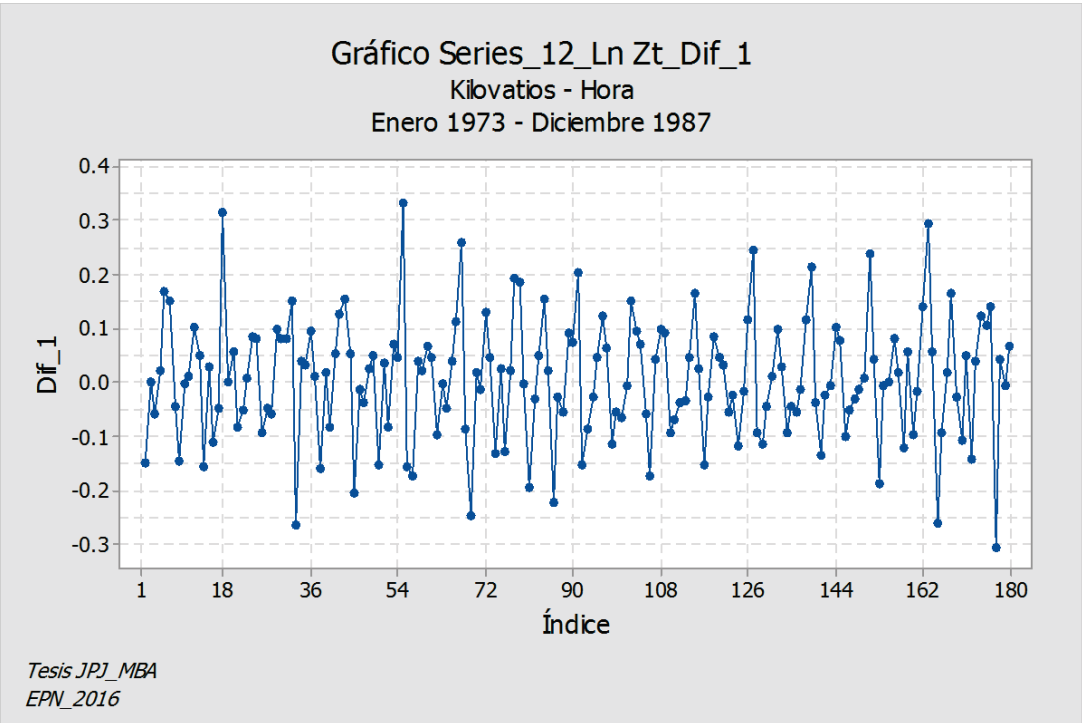


Figura 5.18 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984 d=1

(Pankratz, Forecasting with Dynamic Regression Models, 1991)

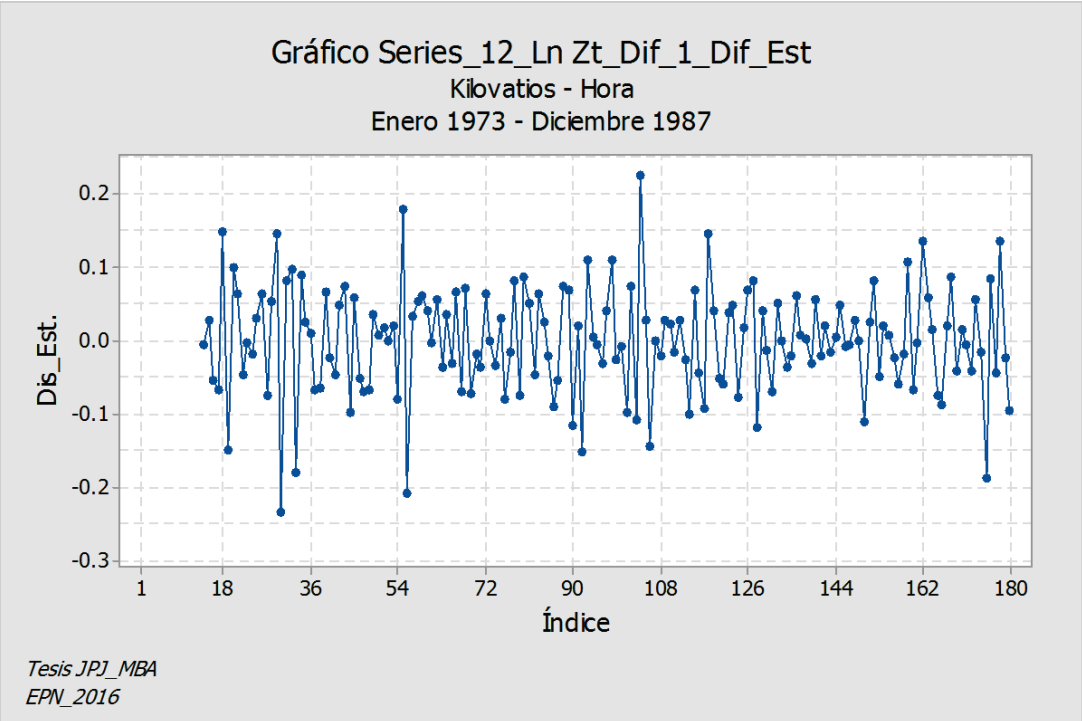


Figura 5.19 Kilovatios – Hora Usados: Enero 1973-Diceimbre 1984 d=D=1

(Pankratz, Forecasting with Dynamic Regression Models, 1991)

5.2.1.8.5.2 ACFs y PACFs Teóricas para Procesos Estacionales

En la práctica patrones estacionales y no estacionales ocurren al mismo tiempo en las series de tiempo y las funciones sacf y spacf. Pero es muy útil tratar de separar las dos partes en la mente cuando se examina las funciones sacf y spacf. Procesos estacionales puros de orden P y Q tienen funciones acfs y pacfs *idénticas* a la de los procesos no estacionales de orden p y q, con una excepción: Para procesos estacionales puros los patrones ocurren en los retardos $s, 2s, 3s, \ldots$, en luagr de los retardos $1,2,3, \ldots$.

Las funciones acfs y pacfs teóricas para procesos estacionales puros siguen los siguientes patrones:

1. Si la función acf decae lentamente en los múltiples retardos s $(s, 2s, 3s, \ldots)$ una diferencia estacional es necesaria.
2. Un proceso AR puramente estacional y estacionario de orden P, la función acf decae en los retardos múltiplos de s; la función pacf tiene picos en los

retardos múltiplos de s y luego desaparece, con el último pico al retardo P. Ver figura 5.20.

3. Un proceso MA puramente estacional y estacionario de orden Q, la función pacf decae en los retardos múltiplos de s; la función acf tiene picos en los retardos múltiplos de s y luego desaparece, con el último pico al retardo Q. Ver figura 5.21
4. Para procesos puramente estacionales mixtos y estacionarios de orden $P\ y\ Q$, las dos funciones acf y pacf decaen en los retardos múltiplos de s.

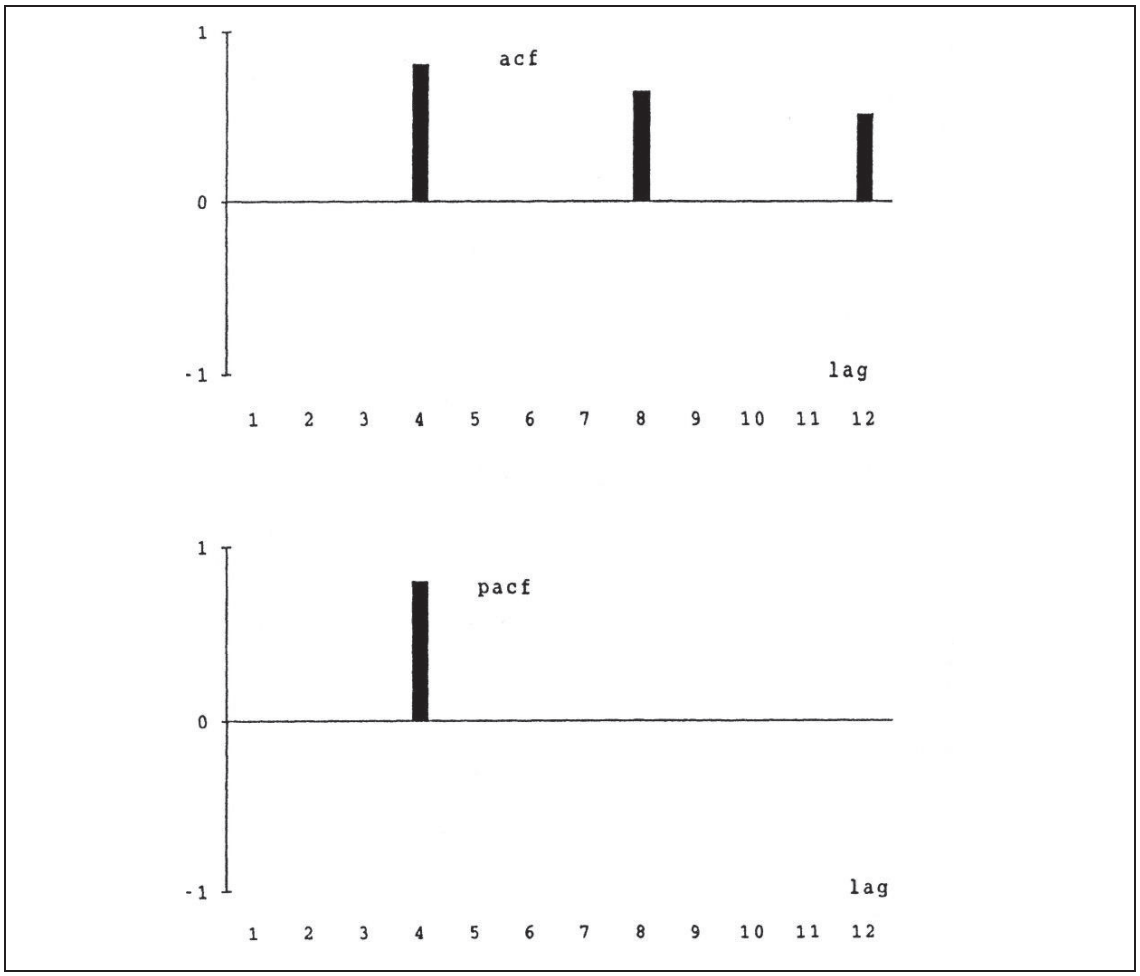


Figura 5.20 ACF y PACF teóricas para un proceso estacional $AR(1)_4$ con $\emptyset_4 = 0.8$

(Pankratz, Forecasting with Dynamic Regression Models, 1991) pp:56

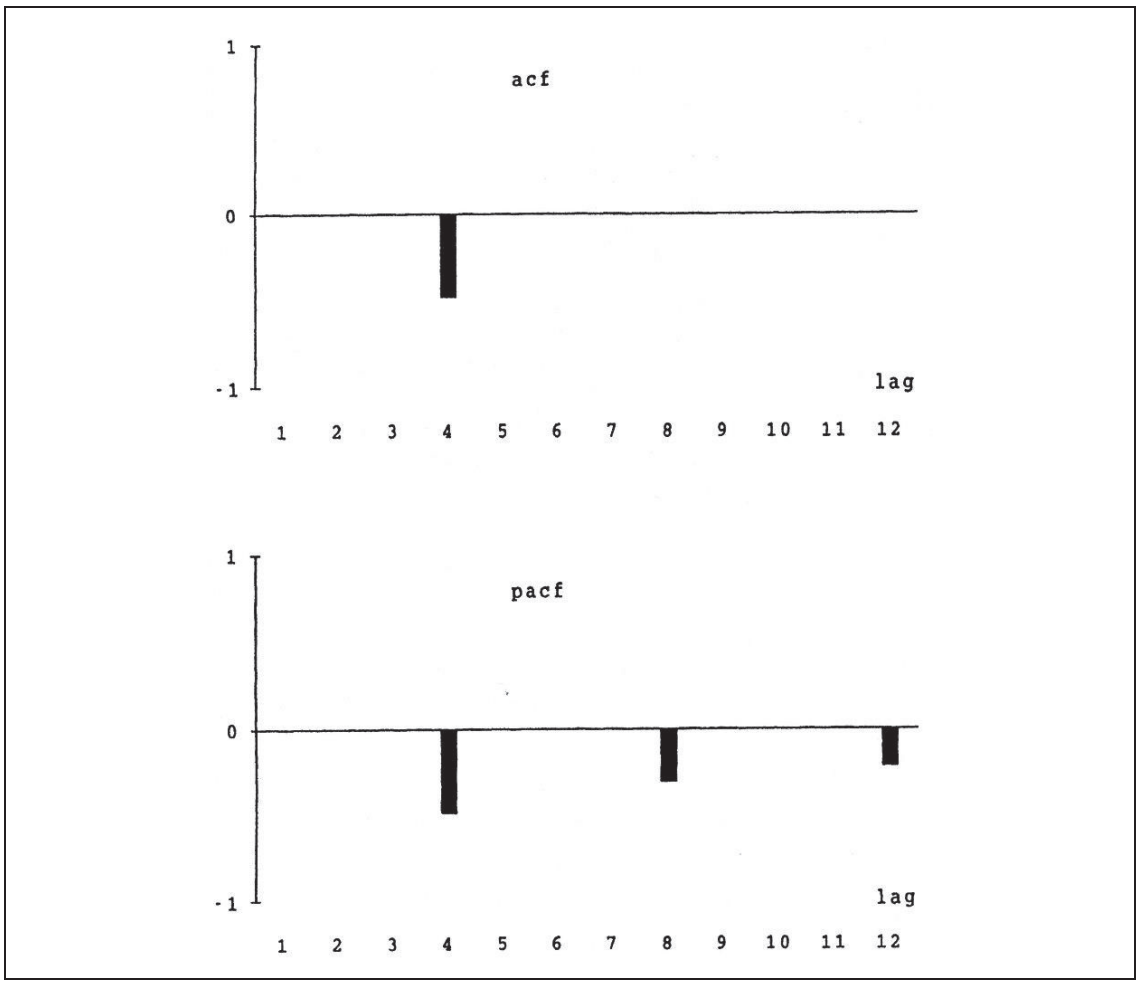


Figura 5.21 ACF y PACF teóricas para un proceso estacional $MA(1)_4$ con $\theta_4 = 0.8$

(Pankratz, Forecasting with Dynamic Regression Models, 1991) pp:57

Un proceso estacional con una media no estacionaria tiene una función acf similar a la función acf de un proceso no estacionario no estacional. Como se vió anteriormente la función acf de un proceso de media no estacionaria no se desvanece rápidamente a cero. Un proceso estacional con media no estacionaria tiene una acf con picos en los retardos $s, 2s, 3s$, ...que no decrece rápidamente hacia cero. No es necesario que la autocorrelación tenga un valor alto para indicar que la media es no estacionaria. El punto clave es que no se desvanece rápidamente a cero. Cuando una realización tiene una función acf estimada similar a la de la figura 5.22, una diferencia estacional es necesaria. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

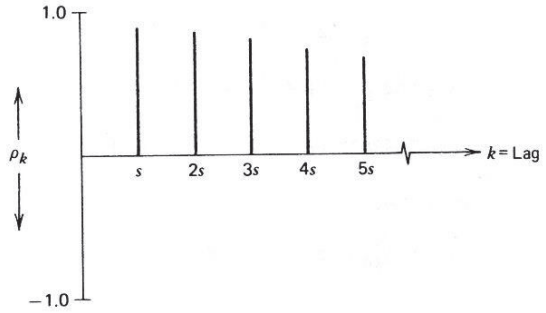


Figura 5.22 Función ACF teórica para un proceso estacional NO estacionario
(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp: 273

5.2.1.8.5.3 Diferencia Estacional

La media de una realización podría cambiar significativamente de un período a otro debido a una fuerte variación estacional. Sin embargo, las observaciones para una estación dada podrían fluctuar alrededor de una media constante.

La diferencia estacional es similar a la diferencia regular ya que las dos calculan los cambios en la serie de datos. En una diferencia regular se calculan cambios de período a período $Z_t - Z_{t-1}$. Pero para calcular una diferencia estacional, se calcula el cambio desde la última estación correspondiente $Z_t - Z_{t-s}$. Por lo tanto se pierden s observaciones debido a la diferencia estacional de longitud s.

La prioridad es eliminar la no-estacionariedad estocástica estacional mediante la diferencia estacional. Entonces la diferencia estacional significa una diferencia del orden de la periodicidad estacional. (Yaffee, 2000).

La acf estimada nos puede dar pistas de una no estacionariedad estacional si esta decae lentamente en los rezagos ($s, 2s, 3s, ...$). Cuando la media de una realización cambia acorde con el patrón estacional, una *diferencia estacional* a menudo induce a una media constante. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Cuando las dos diferencias regular y estacional son aplicadas, no hay diferencia en el resultado final si se aplica una primero que la otra. Sin embargo se recomienda que *la diferencia estacional se aplique primero* ya que a veces la serie resultante puede ser estacionaria y ya no necesita una diferencia regular. (Makridakis S.,

1998), (Capa, Un Primer Curso en Series Temporales, 2008), (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Si D representa el grado de la diferencia estacional, se puede calcular la diferencia estacional de la primera diferencia estacional $(w_t - w_{t-12}) - (w_{t-12} - w_{t-24})$. Esta se llama diferencia estacional de segundo grado $(D = 2)$. En la práctica, diferencias estacionales de segundo orden (o mayores) casi nunca se dan. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Se utilizará el operador de retardo B para escribir expresiones compactas que muestren el grado de la diferencia estacional. Recuerde que el operador de B está definido por: $Z_t B^k = Z_{t-k}$. Si $k = s$, la diferencia D (longitud s) de la serie Z_t puede ser escrita:

$$\nabla_s^D Z_t = (1 - B^S)^D Z_t \qquad (5.91)$$

Por ejemplo si $D = 1$ la primera diferencia estacional de Z_t es:

$$\nabla_s Z_t = (1 - B^S) Z_t$$

$$\nabla_s Z_t = Z_t - B^S Z_t$$

$$\nabla_s Z_t = Z_t - Z_{t-s} \qquad (5.92)$$

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.2.1.8.5.4 Procesos Estacionales y No Estacionales Combinados

Los modelos o procesos ARIMA pueden tener los dos elementos de estacionalidad y no estacionalidad incluidos. Por ejemplo sea el siguiente modelo:

$$(1 - \emptyset_1 B - \emptyset_2 B^2)(1 - B^{12})(1 - B) Z_t = (1 - \theta_{12} B^{12}) a_t \qquad (5.93)$$

Este es un modelo ARIMA $(2,1,0)(0,1,1)_{12}$ donde los ordenes no estacionales $(p, d, q) = (2,1,0)$ y los ordenes estacionales son $(P, D, Q)_{12} = (0,1,1)_{12}$. Este modelo dice que, después de las dos diferencias estacional y no estacional de orden uno $(d = D = 1)$, los registros de los datos tienen un patrón no estacional AR(2) $(p = 2)$ y un patrón estacional $MA\ (1)_{12}$ $(Q = 1)$.

Procesos ARIMA $(p, d, q)(P, D, Q)_s$ Generales

A veces es necesario escribir en una forma general procesos estacionales y no estacionales (E-NE) combinados.

Construir modelos ARIMA para realizaciones (E-NE) es desafiante ya que las funciones estimadas acf's y pacf's reflejan los dos elementos estacionales y no estacionales. Se trata de separar estas dos partes visualmente y mentalmente. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

Box, Jenkins y Reinsel (Box G., 2008 4th Edition) pp: 356-359 sugieren un cierto tipo de modelo para representar realizaciones E-NE. En este tipo de modelo los elementos estacionales y no estacionales se multiplican unos con otros.

Se supone que tenemos la parte no estacional de un modelo ARIMA expresada por:

$$\emptyset(B)\nabla^d Z_t = C + \theta(B)a_t \qquad (5.94)$$

Pero se supone que la parte estacional no está representada todavía en el proceso. Por lo tanto la serie de sus residuos está correlacionada por esa razón se utiliza la serie (representada por b_t) que contiene el patrón estacional.

$$\emptyset(B)\nabla^d Z_t = C^* + \theta(B)b_t \qquad (5.95)$$

Donde $C^* = (1 - \sum_{i=1}^{p} \emptyset_i)$; si $d = 0$, entonces $\mu = \mu_z$; si $d > 0$, entonces $\mu = \mu_w$.

Si se supone que el patrón estacional en la serie b_t puede ser representado por términos AR y MA en los retardos estacionales hasta un retardo estacional AR máximo (P_s) y un retardo estacional MA máximo (Q_s). Se definen los siguientes operadores:

$\nabla_s^D = (1 - B^S)^D$ (Operador de diferencia estacional de orden D)

$\emptyset(B^s) = (1 - \emptyset_s B^s - \emptyset_{2s} B^{2s} - \cdots - \emptyset_{Ps} B^{Ps}$ (Operador AR estacional de orden P)

$\theta(B^s) = (1 - \theta_s B^s - \theta_{2s} B^{2s} - \cdots - \theta_{Ps} B^{Ps}$ (Operador MA estacional de orden P)

Ahora se supone que el comportamiento estacional de b_t se describe por:

$$\emptyset(B^s)\nabla_s^D b_t = \theta(B^s)a_t \qquad (5.96)$$

Despejando b_t de (5.96) y reemplazando en (5.95) se obtiene el modelo *multiplicativo general* ARIMA $(p, d, q)(P, D, Q)$:

$$\emptyset(B^s)\emptyset(B)\nabla_s^D\nabla^d Z_t = C + \theta(B^s)\theta(B)a_t \qquad (5.97)$$

Donde: $C = \emptyset(B^s)C^* = \mu(1 - \sum_{i=1}^{p} \emptyset_i)(1 - \sum_{i=1}^{P} \emptyset_{is})$

Si $d = D = 0$ entonces $\mu = \mu_z$; caso contrario $\mu = \mu_w$, la media de la serie con una diferencia $w_t = \nabla_s^D \nabla^d Z_t$. (Box G., 2008 4th Edition), (Pankratz, Forecasting with Dynamic Regression Models, 1991).

En la práctica todos los valores de (p, q, d, P, Q, D) tienden a ser pequeños, a menudo no más de 1 o 2. Se puede notar en (5.97) que los operadores AR estacionales y no estacionales se multiplican entre sí y los operadores MA de igual manera. Estos elementos podrían ser tratados de forma aditiva; ver (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:288-292.

Estacionariedad e Invertibilidad

Una de las ventajas del modelo multiplicativo general es que simplifica el chequeo de las condiciones de estacionariedad e invertibilidad. Con el multiplicativo estas condiciones se aplican separadamente a los coeficientes estacionales y no estacionales.

Aquí se tratará a B como una variable algebraica ordinaria. Y se considerarán a $\emptyset(B) y\, \theta(B)$ como polinomios en B. Formalmente, estacionariedad requiere que todas las raíces de $\emptyset(B) * \emptyset(B^s) = 0$ estén fuera del círculo unitario, que es equivalente a que las raíces de las ecuaciones características $\emptyset(B) = 0$ y $\emptyset(B^s) = 0$ estén fuera del círculo unitario en el plano complejo. De igual forma, invertibilidad requiere que todas las raíces de las ecuaciones características $\theta(B) = 0$ y $\theta(B^s) = 0$ estén fuera del círculo unitario. Para los casos AR(1), AR(2), MA(1) y MA(2), se mostró anteriormente que se deben cumplir las condiciones (5.37), (5.38), (5.43) y (5.44). Para $p > 2, P > 2, q > 2\ o\ Q > 2$, es fácil usar métodos numéricos para resolver las raíces de la ecuación característica y luego examinar las distancias al origen; todas estas distancias deben exceder a uno en valor absoluto para que el modelo sea estacionario e invertible. (Pankratz, Forecasting with Dynamic Regression Models, 1991).

5.2.1.8.5.5 Equivalencias con Modelos de Suavizamiento Exponencial

Existe un mito muy común que los modelos ARIMA son más generales que los modelos de suavizamiento exponencial. Mientras los modelos de suavizamiento

exponencial lineales son todos casos especiales de los modelos ARIMA, los modelos de suavizamiento exponencial no lineales no tienen ninguna equivalencia con su contraparte ARIMA. Existen también modelos ARIMA que no tienen equivalencia con ningún método de suavizamiento exponencial. En particular, cada modelo de suavizamiento exponencial es no estacionario, mientras los modelos ARIMA pueden ser estacionarios.

Los modelos de suavizamiento exponencial con estacionalidad o tendencia no amortiguada o ambos tienen dos raíces unitarias (es decir se necesitan dos diferencias para volverlos estacionarios). Todos los otros modelos de suavizamiento exponencial tienen una raíz unitaria (solo necesitan una diferencia para volverse estacionarios). (Hyndman R., 2014).

A continuación las equivalencias entre las dos clases de modelos.

El método de ***suavizamiento exponencial simple*** es óptimo para producir pronósticos si la realización Z_t sigue un proceso ***ARIMA(0,1,1)*** y la constante de suavizamiento exponencial es $\alpha = (1 - \theta)$. Además Ledolter y Abraham (1984) muestran que si se utiliza retropredicción (backcasting) para obtener el estimado del nivel medio inicial μ_0, el suavizamiento exponencial simple da el mismo pronóstico que el modelo ARIMA (0,1,1). (Terence, 1990).

El ***método lineal de Holt*** es equivalente a un proceso ***ARIMA(0,2,2)*** (Harrison (1967), Harvey (1984)).

Los parámetros de suavizamiento están dados por $\alpha = 1 + \theta_2$ y $\beta = \frac{(1-\theta_1-\theta_2)}{(1+\theta_2)}$. (Terence, 1990).

El ***método aditivo de Holt – Winters*** produce pronósticos equivalentes a un modelo ***ARIMA***$(\mathbf{0}, \mathbf{1}, \boldsymbol{s}+\mathbf{1})(\mathbf{0}, \mathbf{1}, \mathbf{0})_{\boldsymbol{s}}$. Hay varias restricciones en los parámetros ya que un modelo ARIMA estacional tiene (s+1) parámetros mientras que el método de Holt – Winters usa solamente tres parámetros. (Makridakis S., 1998)

El ***método de Holt – Winters multiplicativo*** no tiene equivalencia con ningún modelo ARIMA.

5.2.1.8.5.6 Aclaración de Pronósticos ARIMA Óptimos

Se ha dicho que los modelos ARIMA producen un pronóstico *óptimo*. Esto significa que ningún otro pronóstico de una sola variable tiene un error medio cuadrático (MSE) más pequeño.

Algunos puntos deben ser aclarados:

- Al decir óptimo se refiere a la esperanza matemática del error de pronóstico *al cuadrado*, no otro error en particular. Esto es, algún otro pronóstico proveniente de otro modelo NO ARIMA podría tener un menor error al cuadrado que el pronóstico de un modelo ARIMA en un caso particular, pero no en promedio.
- Optimo aplica si el modelo ARIMA considerado es el correcto. Así, un pronóstico ARIMA tiene un mínimo MSE en la práctica solamente si la estrategia de identificación, estimación y diagnóstico – chequeo es adecuada para el problema y solo si la estrategia ha sido empleada apropiadamente.
- Se deben comparar pronósticos ARIMA con pronósticos de otros modelos de una sola variable. Ya que modelos de múltiples variables podrían producir pronósticos con un MSE menor que los modelos ARIMA.
- Se consideran solamente modelos de una sola variable con combinaciones *lineales* de su pasado con coeficientes fijos. Es posible que combinaciones NO lineales de la variable podrían producir pronósticos con un menor MSE que el modelo ARIMA lineal.
- Finalmente los coeficientes del modelo ARIMA no dependen del tiempo, es decir se asume son fijos. Modelos de una sola variable con coeficientes variables en el tiempo podrían producir pronósticos con errores MSE más pequeños que el pronóstico ARIMA.

Todas estas condiciones para un pronóstico ARIMA óptimo parecerían ser muy restrictivas. Pero modelos de una sola variable lineales con coeficientes fijos son muy utilizado en la práctica. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983).

5.3 EJEMPLOS PRÁCTICOS CON LA METODOLOGÍA DE BOX-JENKINS

5.3.1 EJEMPLO #. 1

En la tabla 5.3 se muestran datos de permisos para la construcción de viviendas desde el año 1947 hasta 1967 en una región de los Estados Unidos, los datos fueron tomados de la referencia (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp:370, para aplicar la metodología ARIMA.

Tabla 5.3 Ejemplo 1 Permisos de Construcción de Viviendas desde 1947 a 1967 USA

Año	Permisos	Año	Permisos	Año	Permisos	Año	Permisos	Año	Permisos
47	83.3	52	106.8	57	89.7	62	108.2	67	83.5
	83.2		102.2		89.9		110.7		95.8
	105.3		110.3		90.2		113.2		107.7
	117.7		114.1		89.6		114.6		113.7
48	104.6	53	109.1	58	85.8	63	112.2		
	108.8		105.4		96.9		120.2		
	93.9		97.6		112.7		122.1		
	86.1		100.7		122.7		126.6		
49	83	54	102.7	59	119.8	64	122.3		
	102.4		110.9		117.4		115.9		
	119.6		120.2		111.9		116.9		
	141.4		131.3		104.7		110.1		
50	158.6	55	138.9	60	98.3	65	110.4		
	161.3		130.9		94.9		108.9		
	158.2		123.1		93.3		112.1		
	136.1		110.8		90.9		117.6		
51	121.9	56	108.8	61	91.9	66	112.2		
	97.7		103.8		97.2		96		
	103.3		97		104.7		78		
	92.7		93.2		107.7		66.9		

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp: 370

En la figura 5.23 se muestra un gráfico de la serie de tiempo. Los datos se han tomado cada cuatro meses.

Se puede observar en la figura 5.23 que la varianza de la serie es aproximadamente constante, por lo tanto no se realizará ninguna transformación.

Ya que la metodología ARIMA sólo se puede aplicar a series de tiempo estacionarias, se verificará su estacionariedad mediante la prueba de Dickey – Fuller Aumentada (DFA), el programa EViews 9 nos permite correr esta prueba, los resultados se muestran en la tabla 5.4.

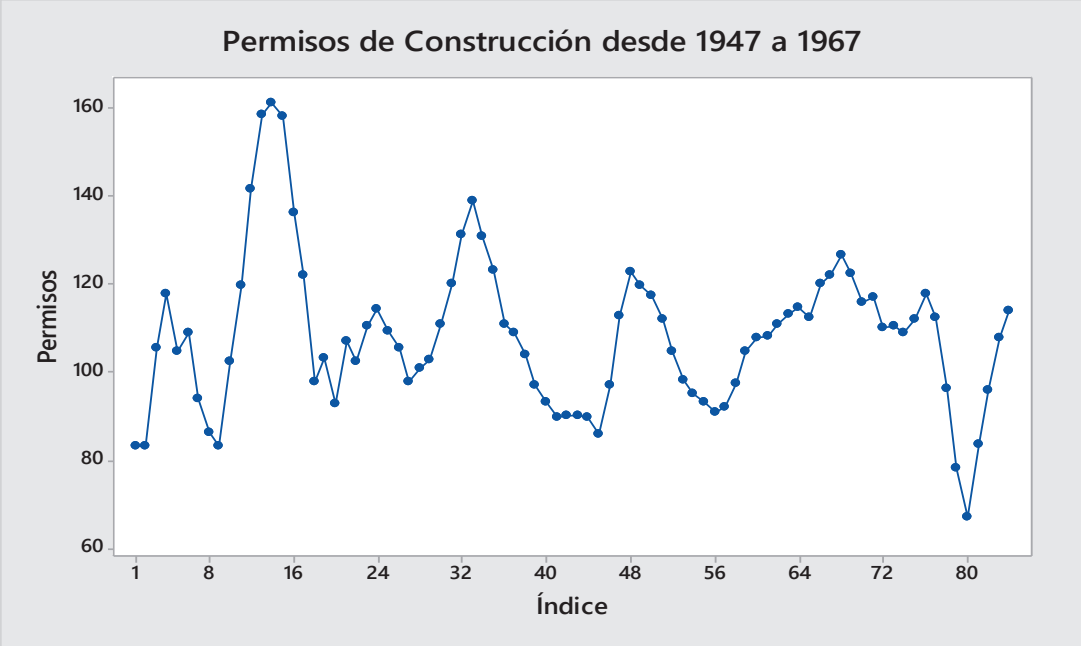


Figura 5.23 Ejemplo # 1 Permisos de Construcción desde 1947 a 1967

(Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983) pp: 370

Tabla 5.4 Prueba de Dickey Fuller Aumentada Ejemplo # 1

Null Hypothesis: SERIES01 has a unit root
Exogenous: Constant
Lag Length: 3 (Automatic - based on SIC, maxlag=11)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-3.668699	0.0064
Test critical values:	1% level	-3.514426	
	5% level	-2.898145	
	10% level	-2.586351	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(SERIES01)
Method: Least Squares
Date: 05/21/16 Time: 12:56
Sample (adjusted): 5 84
Included observations: 80 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
SERIES01(-1)	-0.233880	0.063750	-3.668699	0.0005
D(SERIES01(-1))	0.438027	0.094011	4.659318	0.0000
D(SERIES01(-2))	0.364766	0.103020	3.540747	0.0007
D(SERIES01(-3))	-0.229592	0.109670	-2.093473	0.0397
C	25.30678	6.969393	3.631132	0.0005

R-squared	0.465146	Mean dependent var	-0.050000
Adjusted R-squared	0.436620	S.D. dependent var	9.296358
S.E. of regression	6.977720	Akaike info criterion	6.783783
Sum squared resid	3651.643	Schwarz criterion	6.932660
Log likelihood	-266.3513	Hannan-Quinn criter.	6.843472
F-statistic	16.30627	Durbin-Watson stat	1.832055
Prob(F-statistic)	0.000000		

Después de observar los resultados en la tabla 5.4 se puede concluir que la serie es *estacionaria*, ya que el estadístico DFA (-3.6686) es mayor que los valores críticos de los niveles de significación usuales (1%, 5% y 10%), es decir está en la zona de rechazo de la hipótesis nula, por lo tanto rechazamos la hipótesis nula que existe una raíz unitaria, entonces la serie es estacionaria.

Otra forma de verificar la estacionariedad de la serie (no formal) es analizando los gráficos de sacf y spacf, los mismos se muestran en las figuras 5.24 y 5.25.

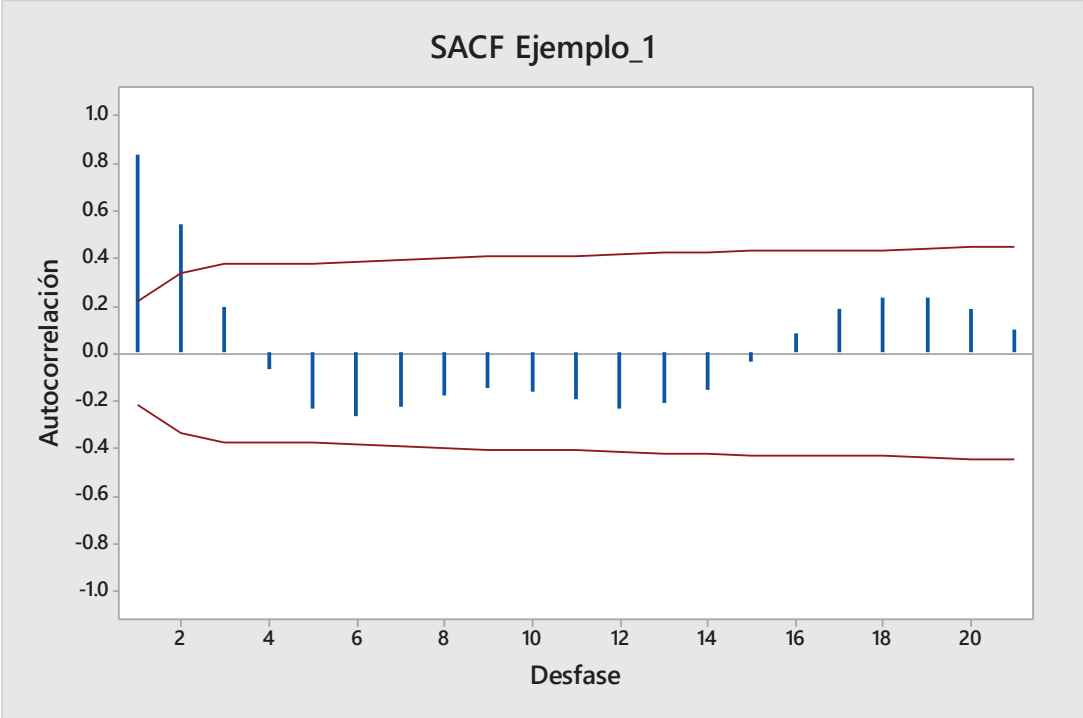


Figura 5.24 Ejemplo # 1 Función de Autocorrelación Muestral SACF

Si la función SACF no decrece rápidamente a cero se podría sospechar de una media no estacionaria, en este caso la SACF decrece muy rápido a cero. Solamente dos autocorrelaciones sobrepasan las líneas que representan el nivel de significancia del 5%. En otras palabras no hay evidencia de que se requiera una diferencia regular para obtener una media estacionaria.

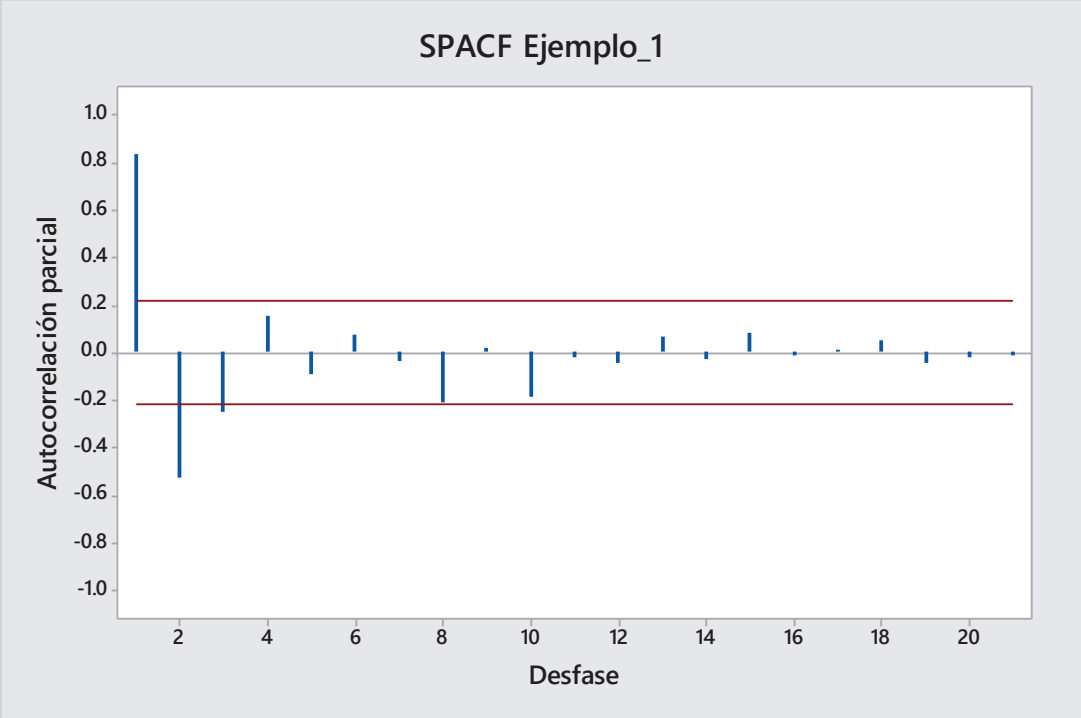


Figura 5.25 Ejemplo # 1 Función de Autocorrelación Parcial Muestral SPACF

Los datos de esta serie se han tomado cada cuatro meses, se debe verificar las autocorrelaciones en los rezagos 4,8,12,…, para descartar algún patrón estacional. A veces los patrones estacionales se muestran de una manera más clara en la función de autocorrelación muestral SACF de las primeras diferencias, en la figura 5.26 se muestra la SACF de las primeras diferencias. En esta figura se puede observar un pico que sobresale en el rezago 4, pero es difícil atribuirle a un patrón estacional, ya que en los rezagos 8 y 12 desaparece, más bien parece una onda seno amortiguada que inicia en el rezago 1.

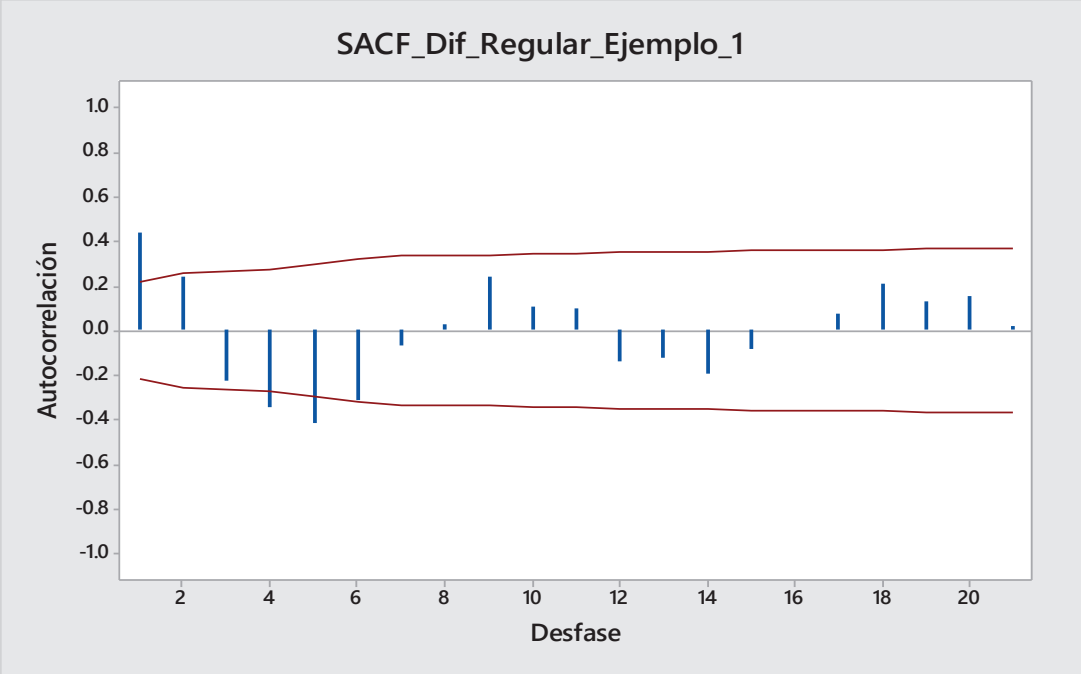


Figura 5.26 Ejemplo # 1 Función de Autocorrelación Primeras Diferencias

A continuación se tratará de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas.

Las figuras 5.24 y 5.25 sugieren un modelo MA(2) puro, ya que la función sacf tiene dos picos significativos y el resto de valores ya no son significativos, mientras la función spacf decrece cambiando el signo en los primeros rezagos.

En las figuras 5.27 y 5.28 se muestran las funciones sacf y spacf de un modelo MA(2) puro. La función sacf residual no es buena ya que existen dos picos en los primeros rezagos, sugiriendo que un patrón AR está presente en la serie de tiempo, y actualmente ya se tiene dos coeficientes MA en los rezagos 1 y 2.

La función spacf residual además sugiere coeficientes AR en el modelo.

Se debe pensar entonces en que un modelo ARMA(p, q).podría ser apropiado, pero es difícil encontrar los valore de $p\ y\ q$. Se va a cambiar de estrategia.

En este caso se iniciará con un modelo AR puro, para que la función sacf residual nos sugiera los coeficientes MA a utilizar. (Pankratz, Forecasting With Univariate Box-Jenkins Models, 1983)

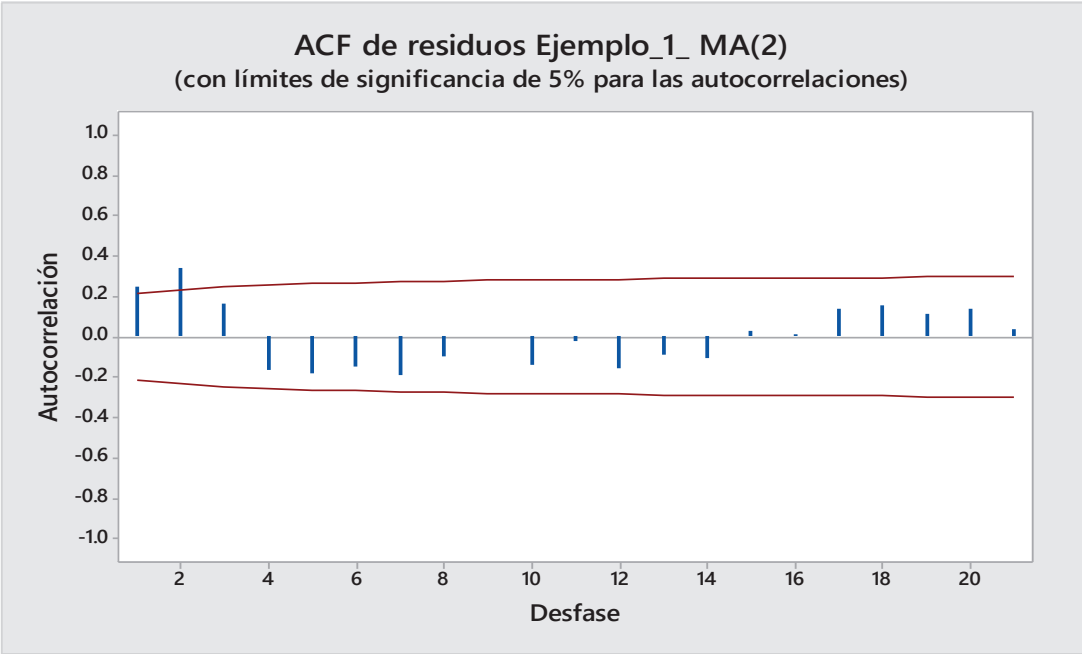


Figura 5.27 Ejemplo # 1 Función de Autocorrelación Modelo MA(2)

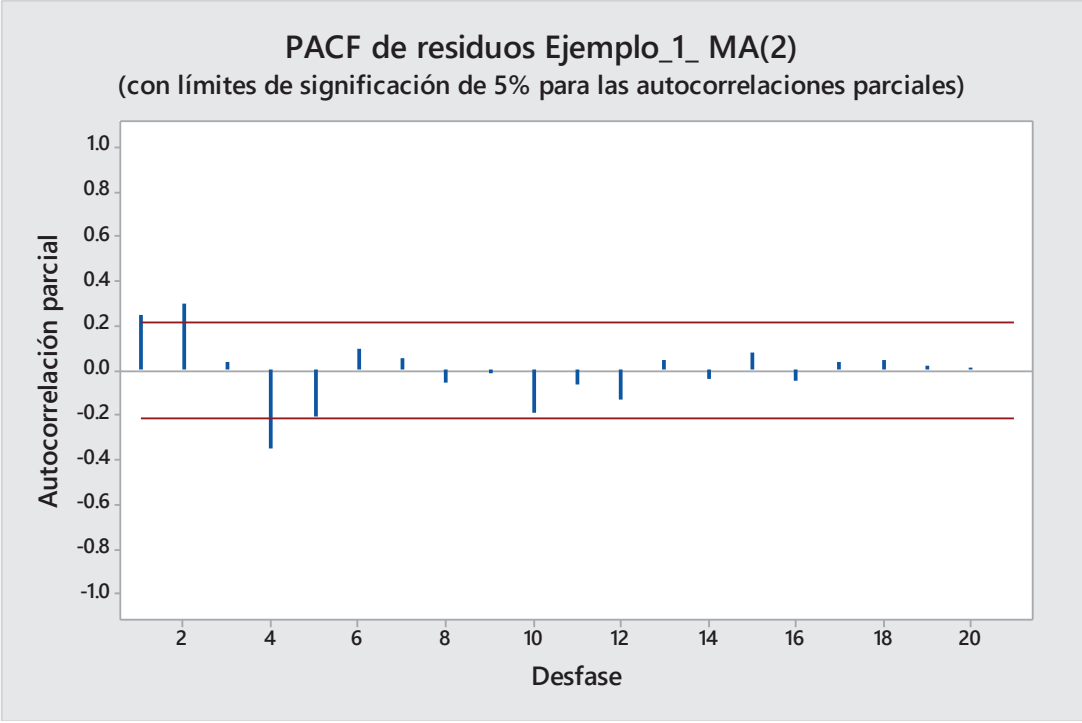


Figura 5.28 Ejemplo # 1 Función de Autocorrelación Parcial Modelo MA(2)

A continuación se muestra las función sacf residual de un modelo AR(2) puro.

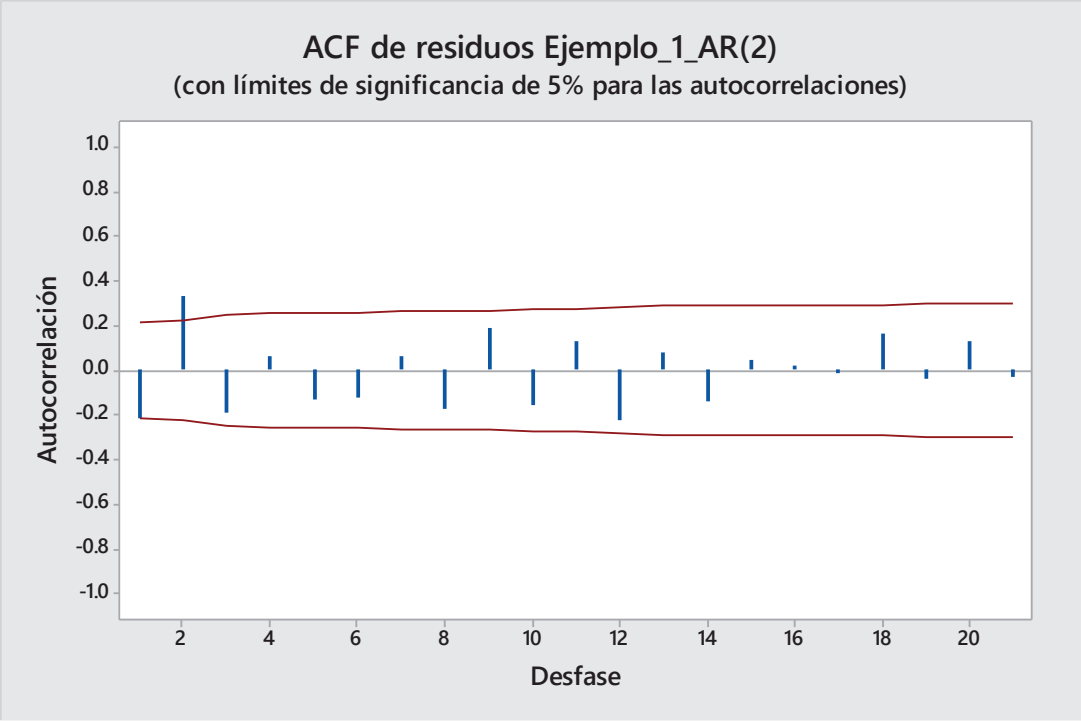


Figura 5.29 Ejemplo # 1 Función de Autocorrelación del Modelo AR(2)

La figura 5.29 sugiere un coeficiente θ_2 para un modelo MA(2) y θ_1 igual a cero ya que el pico en el rezago 1 no es significativo. Se postulará el siguiente modelo ARMA(2,2):

$$(1 - \emptyset_1 B - \emptyset_2 B^2)\tilde{Z}_t = (1 - \theta_2 B^2)a_t \qquad (5.99)$$

Los resultados de la estimación mediante el programa EViews 9 se muestran en la Tabla 5.4.

La tabla 5.5 muestra que el modelo es satisfactorio ya que todos los coeficientes son estadísticamente significativos (estadístico t mayor a 2), los coeficientes del componente AR cumplen con las condiciones de estacionariedad:

$|\emptyset_2| < 1 \ (0.54 < 1)$;

$(\emptyset_2 + \emptyset_1) < 1 \ (-0.5454 + 1.228) = 0.6826 < 1$;

$(\emptyset_2 - \emptyset_1) < 1 \ (-0.5454 - 1.228) = -1.77 < 1.$

Tabla 5.5 Estimación Modelo ARMA(2,2) Ejemplo # 1

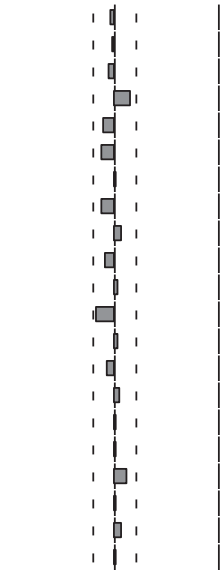
Dependent Variable: SERIES01
Method: ARMA Maximum Likelihood (BFGS)
Date: 05/21/16 Time: 21:55
Sample: 1 84
Included observations: 84
Convergence achieved after 7 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	107.8837	3.412018	31.61875	0.0000
AR(1)	1.228092	0.091993	13.34982	0.0000
AR(2)	-0.545423	0.092111	-4.921345	0.0000
MA(2)	0.431210	0.118992	3.623850	0.0005
SIGMASQ	48.64525	7.586545	6.412043	0.0000

R-squared	0.833974	Mean dependent var	108.1298
Adjusted R-squared	0.825568	S.D. dependent var	17.21999
S.E. of regression	7.191945	Akaike info criterion	6.871699
Sum squared resid	4086.201	Schwarz criterion	7.016391
Log likelihood	-283.6114	Hannan-Quinn criter.	6.929864
F-statistic	99.20727	Durbin-Watson stat	2.038532
Prob(F-statistic)	0.000000		

El estadístico de Durbin Watson es cercano a 2 por lo tanto no hay autocorrelación residual, el correlograma de los residuos sacf se muestra en la figura 5.30, la misma ratifica que los residuos no están autocorrelacionados, es decir se aproximan a ruido blanco. Por lo tanto el modelo es estadísticamente adecuado.

Date: 11/04/16 Time: 16:41
Sample: 1 84
Included observations: 84
Q-statistic probabilities adjusted for 3 ARMA terms



	AC	PAC	Q-Stat	Prob
1	-0.032	-0.032	0.0895	
2	-0.013	-0.014	0.1052	
3	-0.057	-0.058	0.3925	
4	0.165	0.162	2.8634	0.091
5	-0.104	-0.100	3.8559	0.145
6	-0.132	-0.140	5.4814	0.140
7	0.017	0.029	5.5084	0.239
8	-0.117	-0.166	6.8160	0.235
9	0.074	0.095	7.3394	0.291
10	-0.095	-0.066	8.2207	0.314
11	0.025	-0.031	8.2828	0.406
12	-0.183	-0.152	11.627	0.235
13	0.031	-0.041	11.722	0.304
14	-0.079	-0.089	12.371	0.336
15	0.041	0.030	12.547	0.403
16	0.004	0.012	12.549	0.483
17	0.016	-0.009	12.578	0.560
18	0.127	0.096	14.335	0.500
19	0.009	-0.006	14.344	0.573
20	0.068	0.013	14.869	0.605
21	0.002	0.060	14.870	0.671

Figura 5.30 Ejemplo # 1 Correlograma residuos modelo ARMA(2,2)

Tabla 5.6 Inverso de la raíces modelo ARMA(2,2) Ejemplo # 1

Inverse Roots of AR/MA Polynomial(s)
Specification: SERIES01 C AR(1) AR(2) MA(2)
Date: 11/04/16 Time: 16:51
Sample: 1 84
Included observations: 84

AR Root(s)	Modulus	Cycle
0.614046 ± 0.410330i	0.738528	10.66590

No root lies outside the unit circle.
ARMA model is stationary.

MA Root(s)	Modulus	Cycle
0.000000 ± 0.656665i	0.656665	4.000000

No root lies outside the unit circle.
ARMA model is invertible.

El inverso de las raíces del polinomio característico del modelo ARMA (2,2) no caen fuera del círculo unitario, por lo tanto se puede considerar al modelo estable.

El modelo con los coeficientes estimados es:

$(1 - 1.228B + 0.5454B^2)\widetilde{Z}_t = (1 - 0.4312B^2)a_t$ (5.100)

Después de este análisis se puede concluir que le modelo (5.100) es estadísticamente adecuado.

Ya que el modelo ARMA(2,2) como se muestra en (5.100) es adecuado, se procede a realizar el pronóstico mediante el programa EViews 9. Los resultados se presentan en la tabla 5.7.

Tabla 5.7 Pronóstico Modelo ARMA(2,2) Ejemplo # 1 año 1968

Año	Inferior	Pronóstico	Superior
68	98.26	113.12	127.98
	87.93	111.36	136.79
	78.16	109.30	140.43
	72.96	107.72	142.49

En la figura 5.31 se muestra el gráfico del pronóstico con el modelo ARMA(2,2) y sus intervalos de confianza.

En la referencia (Box G., 2008 4th Edition)pp:62-63 los autores advierten que un caso especial se da en modelos de segundo orden, cuando el discriminante de la ecuación característica es menor a cero ($\Delta = \emptyset_1^2 + 4\emptyset_2 < 0$), en un modelo AR(2) se dará un comportamiento *pseudoperiódico* y se generará un *ciclo estocástico* cuyo *período promedio* (p^*) se calcula mediante (5.36):

$$p^* = \frac{360^o}{cos^{-1}\left(\frac{\emptyset_1}{2(-\emptyset_2)^{\frac{1}{2}}}\right)}$$

En este ejemplo el discriminante $\Delta = -0.67$, entonces se tendrá un ciclo estocástico cuyo período será igual a $p^* = 10.66$ cuartos, es decir hay un comportamiento cíclico cuyo ciclo se repite cada 42 meses o 3.5 años.

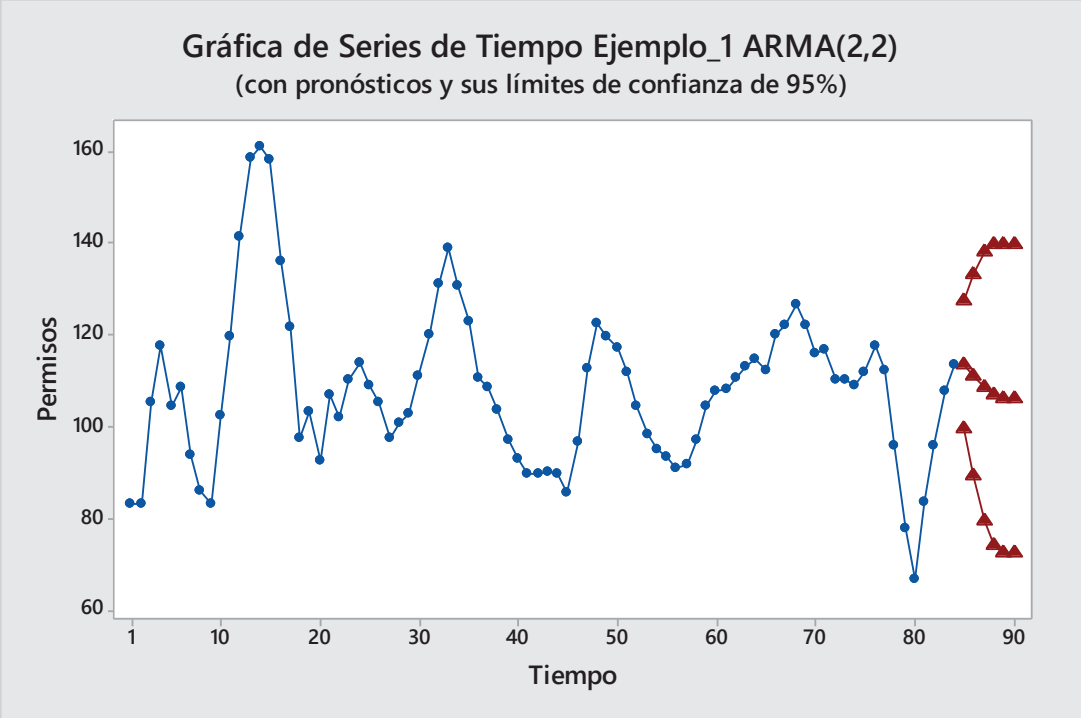


Figura 5.31 Ejemplo # 1 Pronósticos con Modelo ARMA(2,2)

5.3.2 EJEMPLO #. 2

En la tabla 5.8 se presentan los datos del ejemplo # 2, los datos representan las ventas de un producto desde el año 1997 hasta el 2006, el mismo fue tomado de la referencia (Capa, Un Primer Curso en Series Temporales, 2008) pp: 163-177.

Tabla 5.8 Datos Ejemplo # 2 Ventas de un producto desde 1997 hasta 2006

Mes	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006
1	1490	1543	2184	2279	2583	2911	2972	2579	3681	3899
2	1597	1529	2221	2256	2303	2786	2841	3213	3365	3217
3	908	1153	1167	1289	1388	1571	1674	1978	2265	2148
4	437	503	522	649	772	770	947	951	1043	1134
5	150	205	184	241	228	275	296	332	336	348
6	112	132	141	170	169	188	187	212	236	256
7	87	89	106	115	131	152	160	162	177	180
8	161	189	201	207	234	257	288	314	322	321
9	498	636	652	815	853	938	1015	993	1044	1152
10	1433	1185	1543	1520	1653	1694	2348	2255	2670	2681
11	1793	2358	2434	2825	2841	3269	3790	3659	4405	4511
12	3195	3258	3652	4323	4422	4597	5816	5909	6437	6327

En la figura 5.32 se muestra el gráfico de las ventas de un producto desde el año 1997 hasta el año 2006.

Se modelará la serie con los datos hasta el año 2005 y se reservarán los datos del año 2006 para comparar con los pronósticos obtenidos de la modelación ARIMA.

Para la modelación de esta serie se utilizarán dos paquetes estadísticos Minitab ver. 17 y Eviews ver. 9, Minitab ver. 17 básicamente para gráficos en este ejemplo.

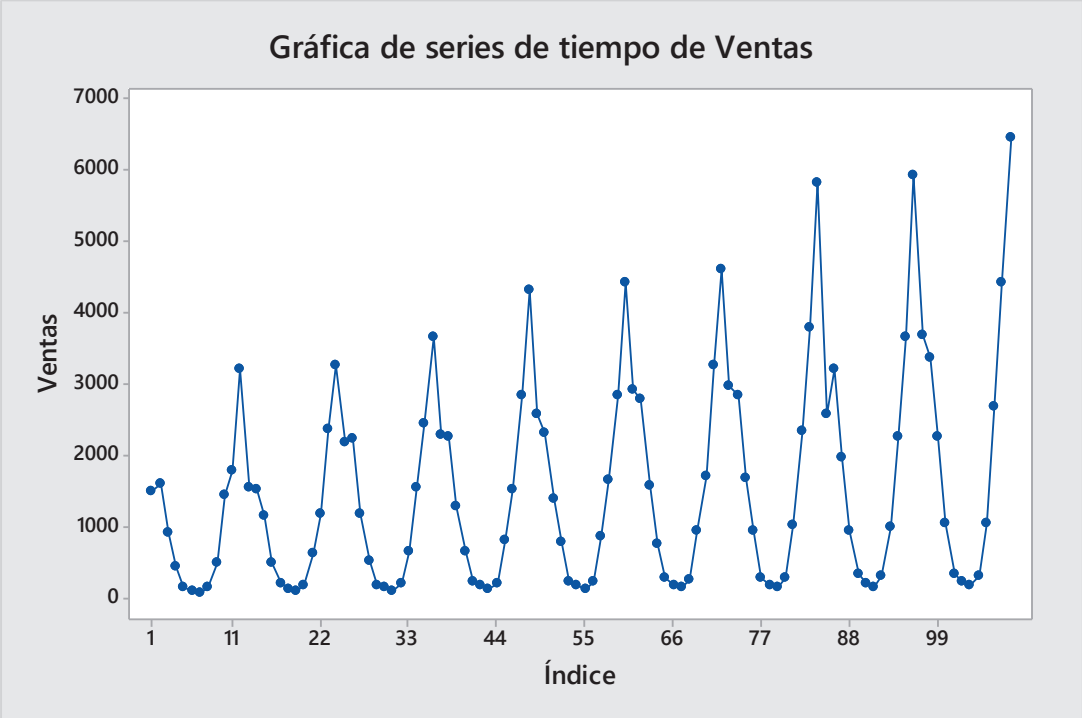


Figura 5.32 Ejemplo # 2 Ventas de un Producto desde 1997 hasta 2005

(Capa, Un Primer Curso en Series Temporales, 2008) pp:163-177

El primer paso en la identificación es mirar el gráfico de la serie de tiempo, se puede observar claramente un marcado patrón estacional razón por la cual se aplicará una diferencia estacional $(D = 1)$, además se puede observar que no hay una tendencia marcada en la serie, por lo que a primera vista no se aplicara una diferencia regular.

A continuación se muestran los gráficos de las funciones de autocorrelación y autocorrelación parcial de la realización, figuras 5.33 y 5.34.

En la figura 5.33 se puede observar que la función de autocorrelación muestral decae rápidamente a cero, ya que en el rezago tres prácticamente se alcanza ese

valor. Esto corrobora lo observado en el gráfico de la serie, no es necesaria una diferencia regular $(d = 0)$.

En la figura 5.35 se muestra la serie con una primera diferencia estacional $(D = 1)$. El gráfico indica que ya no es necesario otra diferencia estacional, ya que los datos varían alrededor de la media y se ha eliminado el patrón estacional. El correlograma de la serie con $D = 1$ ratifica lo dicho, ver figuras 5.36 y 5.37.

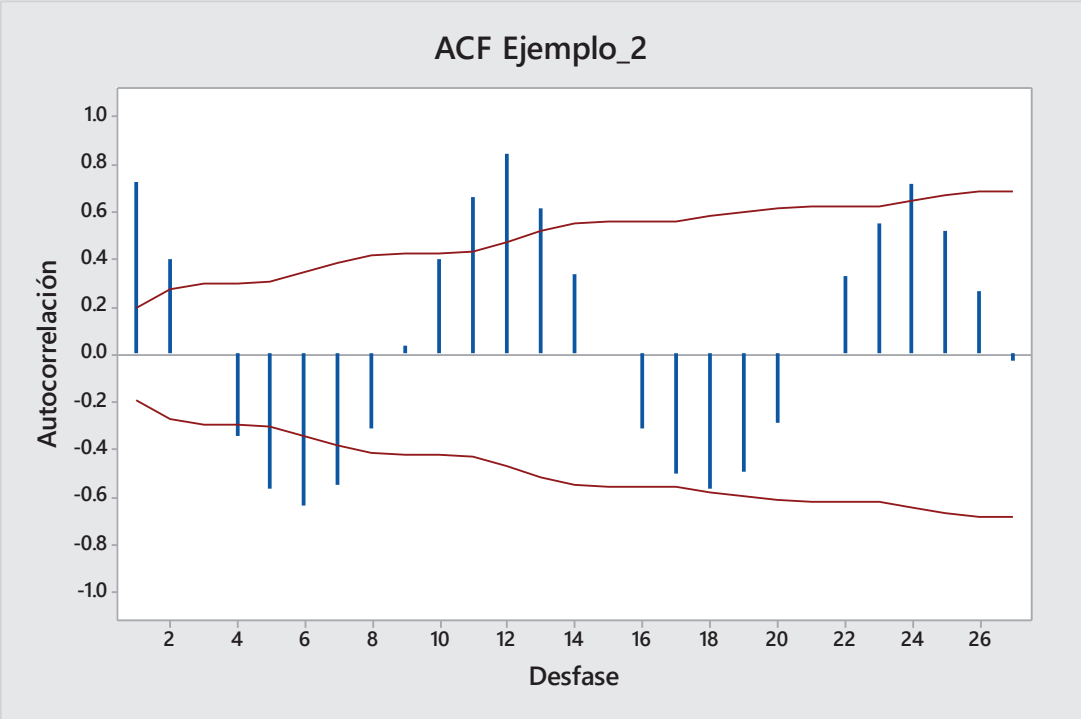


Figura 5.33 Ejemplo # 2 Función de Autocorrelación Muestral SACF

En la figura 5.36 se puede observar que la autocorrelación muestral en el rezago 13 se sale de la banda de confianza (95%), por lo que tentativamente se elige un coeficiente MA(13), en la figura 5.37 se observa que las autocorrelaciones parciales de orden 12 y 13 también son significativas, razón por la cual se eligen coeficientes AR(12) y AR(13).

El modelo 1 quedaría:

$$(1 - \emptyset_{12}B^{12} - \emptyset_{13}B^{13})\tilde{Z}_t = (1 - \theta_{13}B^{13})a_t \qquad (5.101)$$

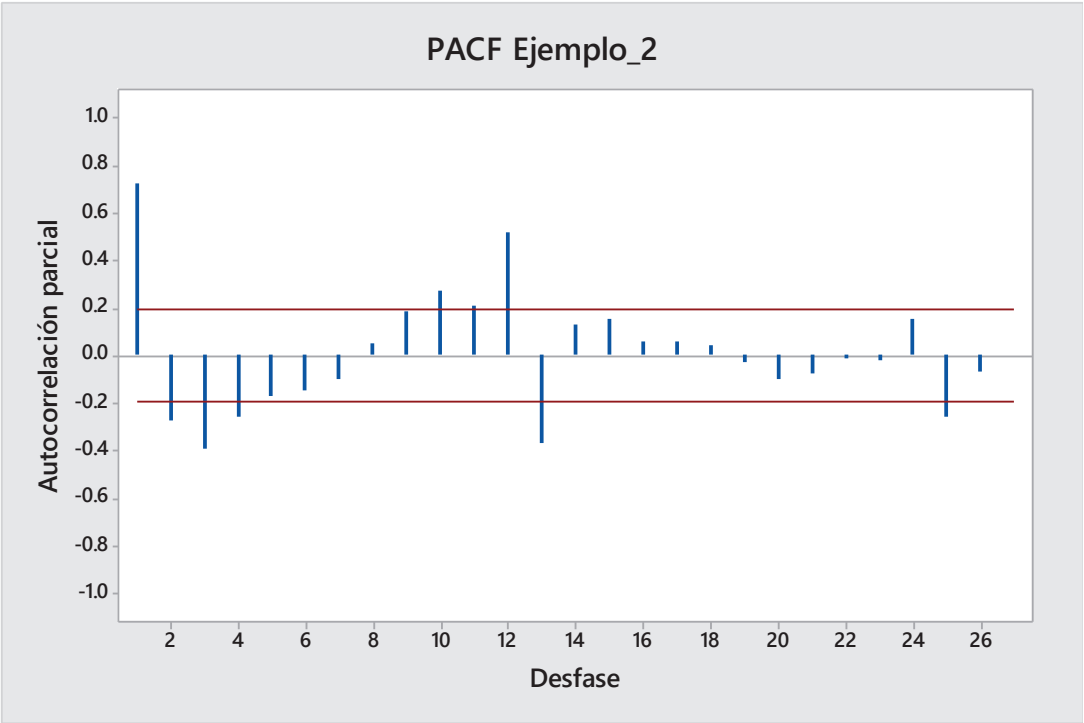


Figura 5.34 Ejemplo # 2 Función de Autocorrelación Parcial SPACF

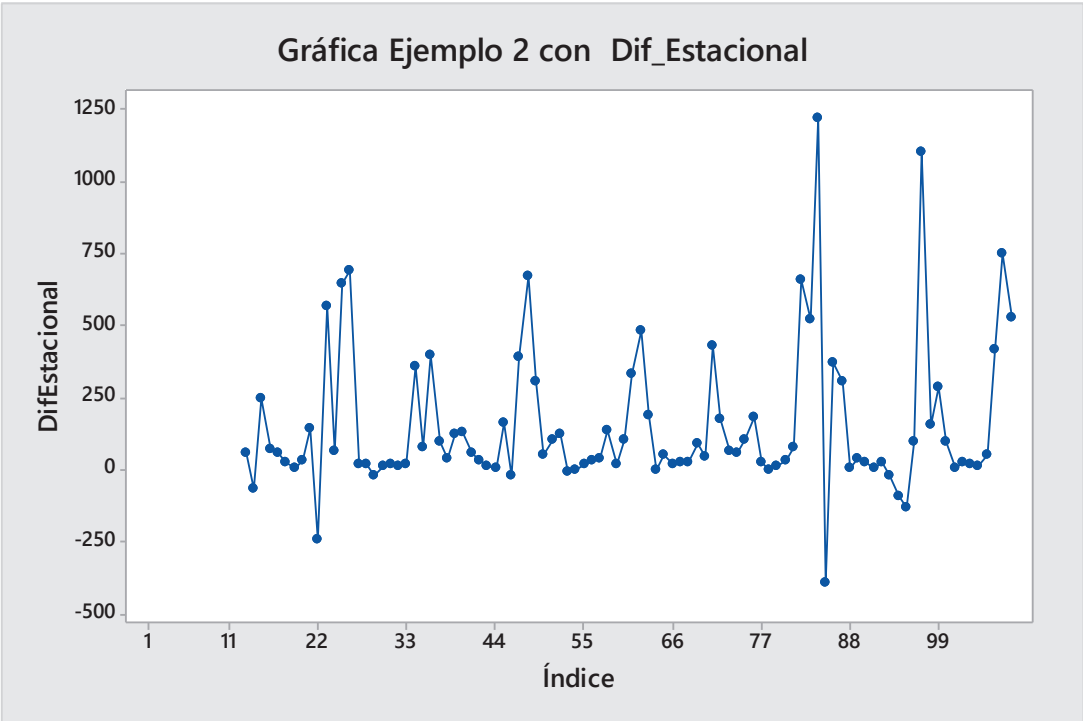


Figura 5.35 Ejemplo # 2 Gráfico de la serie con una diferencia estacional

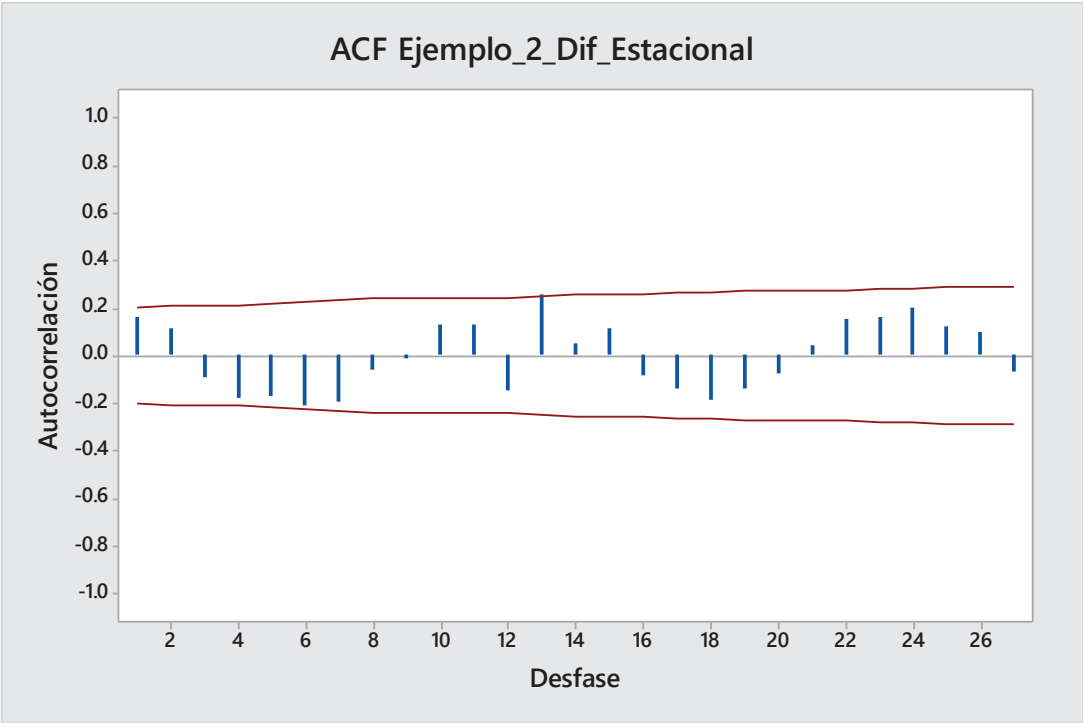


Figura 5.36 Ejemplo # 2 SACFde la serie con una diferencia estacional

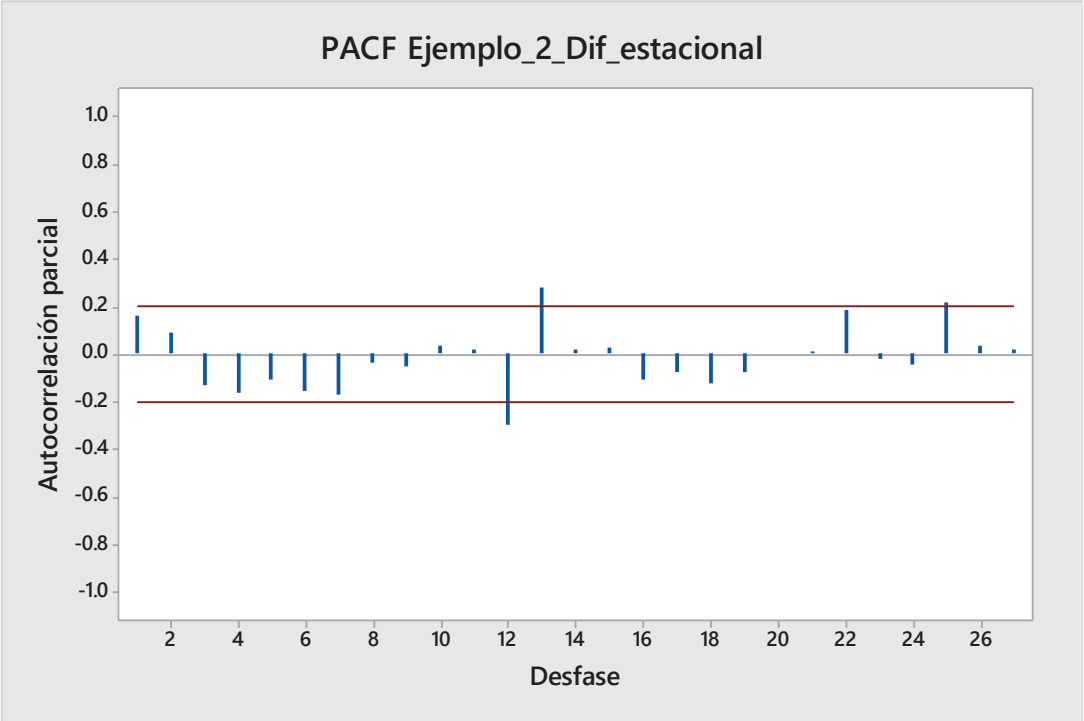


Figura 5.37 Ejemplo # 2 SPACF de la serie con una diferencia estacional

Siguiendo la metodología iterativa de Box – Jenkins, se inicia el proceso de identificación, estimación, diagnóstico – chequeo y modificación si fuera necesario. En la tabla 5.9 se muestran los resultados de la estimación del modelo 1 mediante el programa EViews 9.

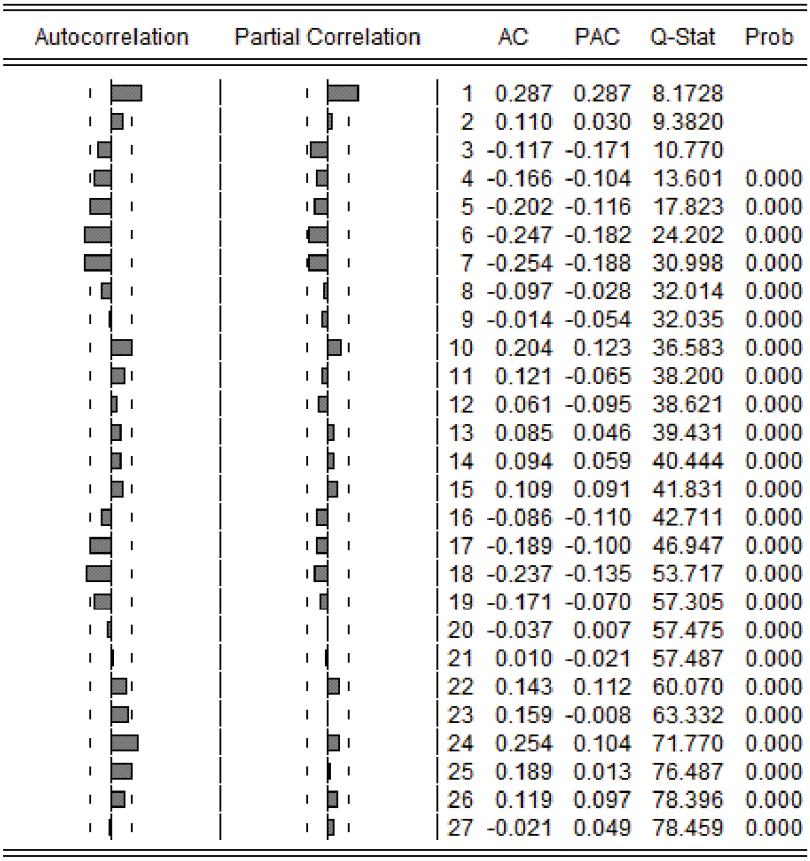
Se puede observar que el coeficiente AR(13) no es significativo (t=-0.63 es menor a 2) o (p>0.05), por esta razón se descarta este coeficiente.

En la figura 5.38 se muestra el correlograma del modelo 1, el cual nos indica que existe autocorrelación y autocorrelación parcial en el rezago 1, lo que sugiere aumentar un coeficiente AR(1) o MA(1).

Tabla 5.9 Resultados Estimación Modelo 1

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	148.3624	29.96524	6.951151	0.0000
AR(12)	-0.355102	0.103143	-3.442813	0.0009
AR(13)	-0.195047	0.310038	-0.629106	0.5309
MA(13)	0.580960	0.324745	1.788974	0.0769
SIGMASQ	50596.38	6279.728	8.057097	0.0000

R-squared	0.192236	Mean dependent var	147.0833
Adjusted R-squared	0.156730	S.D. dependent var	251.5888
S.E. of regression	231.0333	Akaike info criterion	13.82060
Sum squared resid	4857252.	Schwarz criterion	13.95416
Log likelihood	-658.3887	Hannan-Quinn criter.	13.87459
F-statistic	6.414168	Durbin-Watson stat	1.372526
Prob(F-statistic)	0.000589		



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.287	0.287	8.1728	
		2	0.110	0.030	9.3820	
		3	-0.117	-0.171	10.770	
		4	-0.166	-0.104	13.601	0.000
		5	-0.202	-0.116	17.823	0.000
		6	-0.247	-0.182	24.202	0.000
		7	-0.254	-0.188	30.998	0.000
		8	-0.097	-0.028	32.014	0.000
		9	-0.014	-0.054	32.035	0.000
		10	0.204	0.123	36.583	0.000
		11	0.121	-0.065	38.200	0.000
		12	0.061	-0.095	38.621	0.000
		13	0.085	0.046	39.431	0.000
		14	0.094	0.059	40.444	0.000
		15	0.109	0.091	41.831	0.000
		16	-0.086	-0.110	42.711	0.000
		17	-0.189	-0.100	46.947	0.000
		18	-0.237	-0.135	53.717	0.000
		19	-0.171	-0.070	57.305	0.000
		20	-0.037	0.007	57.475	0.000
		21	0.010	-0.021	57.487	0.000
		22	0.143	0.112	60.070	0.000
		23	0.159	-0.008	63.332	0.000
		24	0.254	0.104	71.770	0.000
		25	0.189	0.013	76.487	0.000
		26	0.119	0.097	78.396	0.000
		27	-0.021	0.049	78.459	0.000

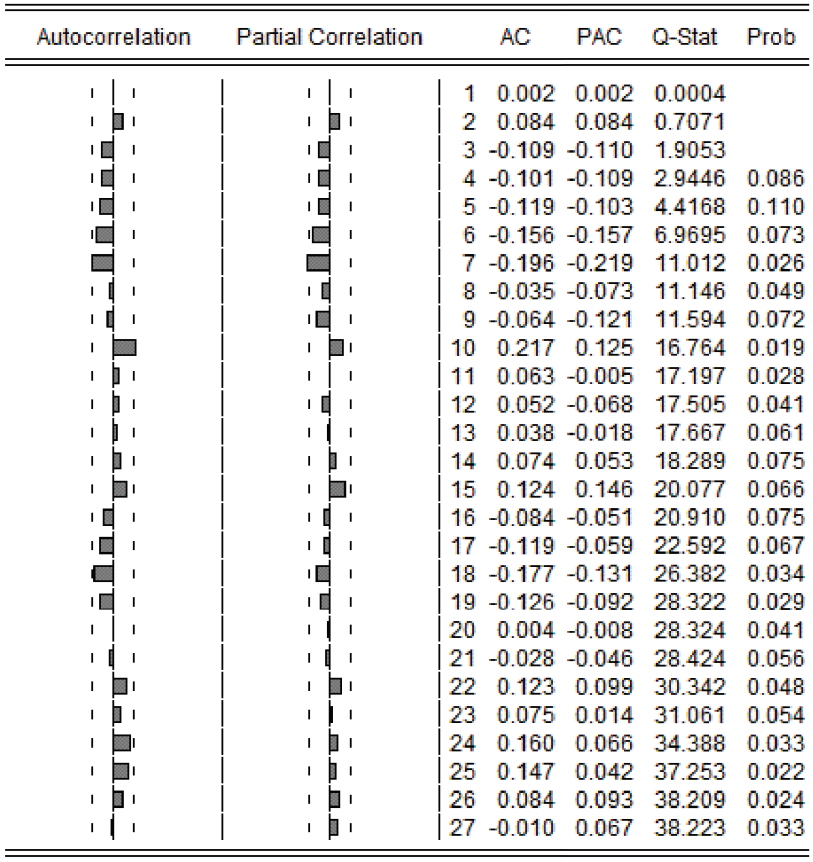
Figura 5.38 Correlograma Modelo 1 Ejemplo2

Se aumentó un coeficiente AR(1) al modelo 1 y al modelo resultante se denominará modelo 2.

Tabla 5.10 Resultados Estimación Modelo 2

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	149.4970	38.97028	3.836179	0.0002
AR(12)	-0.383738	0.099935	-3.839874	0.0002
AR(1)	0.296579	0.059164	6.012786	0.0000
MA(13)	0.508548	0.100090	6.080885	0.0000
SIGMASQ	46467.29	4953.818	9.380096	0.0000

R-squared	0.258156	Mean dependent var	147.0833
Adjusted R-squared	0.225548	S.D. dependent var	251.5888
S.E. of regression	221.4056	Akaike info criterion	13.73546
Sum squared resid	4460860.	Schwarz criterion	13.86902
Log likelihood	-656.3022	Hannan-Quinn criter.	13.78945
F-statistic	7.916836	Durbin-Watson stat	1.967828
Prob(F-statistic)	0.000016		



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.002	0.002	0.0004	
		2	0.084	0.084	0.7071	
		3	-0.109	-0.110	1.9053	
		4	-0.101	-0.109	2.9446	0.086
		5	-0.119	-0.103	4.4168	0.110
		6	-0.156	-0.157	6.9695	0.073
		7	-0.196	-0.219	11.012	0.026
		8	-0.035	-0.073	11.146	0.049
		9	-0.064	-0.121	11.594	0.072
		10	0.217	0.125	16.764	0.019
		11	0.063	-0.005	17.197	0.028
		12	0.052	-0.068	17.505	0.041
		13	0.038	-0.018	17.667	0.061
		14	0.074	0.053	18.289	0.075
		15	0.124	0.146	20.077	0.066
		16	-0.084	-0.051	20.910	0.075
		17	-0.119	-0.059	22.592	0.067
		18	-0.177	-0.131	26.382	0.034
		19	-0.126	-0.092	28.322	0.029
		20	0.004	-0.008	28.324	0.041
		21	-0.028	-0.046	28.424	0.056
		22	0.123	0.099	30.342	0.048
		23	0.075	0.014	31.061	0.054
		24	0.160	0.066	34.388	0.033
		25	0.147	0.042	37.253	0.022
		26	0.084	0.093	38.209	0.024
		27	-0.010	0.067	38.223	0.033

Figura 5.39 Correlograma Modelo 2 Ejemplo 2

En la tabla 5.10 se puede observar que todos los coeficientes son significativos y el correlograma de los residuos (Figura 5.39) prácticamente es ruido blanco.

En base al modelo 2 se podría tratar de mejorarlo aumentando un coeficiente MA(7) o AR(7), ya que si se observa la Figura 5.39 la autocorrelación muestral en el rezago 7 casi esta fuera de la banda de confianza. Además se debería aumentar un coeficiente MA(10), ya que la autocorrelación muestral en el rezago 10 también está al borde de la banda de confianza.

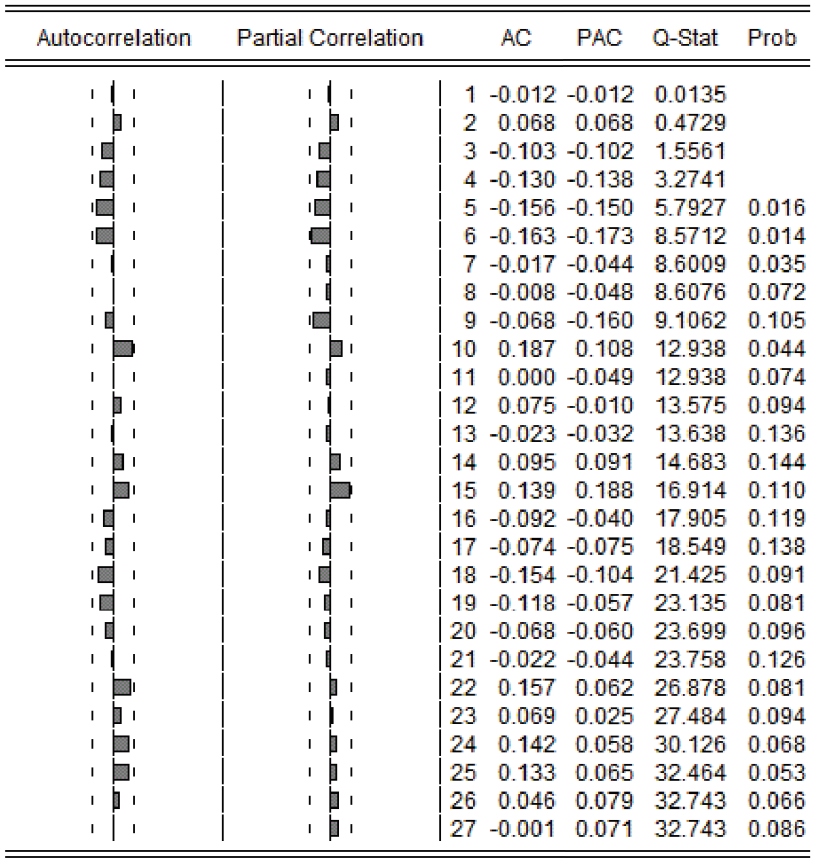
Iniciaremos estos cambios creando el modelo 3, es decir aumentando un coeficiente MA(7). Los resultados de la estimación se muestran en la Tabla 5.11 y su correlograma en la figura 5.40.

Todos los coeficientes son significativos, el coeficiente MA(7) es muy cercano a 2 por lo tanto se lo mantendrá, ya que ayuda a mejorar el correlograma.

Tabla 5.11 Resultados Estimación Modelo 3

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	148.5280	33.43670	6.442066	0.0000
AR(12)	-0.471069	0.104065	-6.526691	0.0000
AR(1)	0.306030	0.104610	2.925432	0.0044
MA(13)	0.657894	0.220937	2.977739	0.0037
MA(7)	-0.252134	0.142378	-1.770879	0.0800
SIGMASQ	41252.37	5217.735	7.906184	0.0000

R-squared	0.341412	Mean dependent var	147.0833
Adjusted R-squared	0.304823	S.D. dependent var	251.5888
S.E. of regression	209.7678	Akaike info criterion	13.69646
Sum squared resid	3960228.	Schwarz criterion	13.85673
Log likelihood	-651.4302	Hannan-Quinn criter.	13.76125
F-statistic	9.331187	Durbin-Watson stat	1.993769
Prob(F-statistic)	0.000000		



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.012	-0.012	0.0135	
		2	0.068	0.068	0.4729	
		3	-0.103	-0.102	1.5561	
		4	-0.130	-0.138	3.2741	
		5	-0.156	-0.150	5.7927	0.016
		6	-0.163	-0.173	8.5712	0.014
		7	-0.017	-0.044	8.6009	0.035
		8	-0.008	-0.048	8.6076	0.072
		9	-0.068	-0.160	9.1062	0.105
		10	0.187	0.108	12.938	0.044
		11	0.000	-0.049	12.938	0.074
		12	0.075	-0.010	13.575	0.094
		13	-0.023	-0.032	13.638	0.136
		14	0.095	0.091	14.683	0.144
		15	0.139	0.188	16.914	0.110
		16	-0.092	-0.040	17.905	0.119
		17	-0.074	-0.075	18.549	0.138
		18	-0.154	-0.104	21.425	0.091
		19	-0.118	-0.057	23.135	0.081
		20	-0.068	-0.060	23.699	0.096
		21	-0.022	-0.044	23.758	0.126
		22	0.157	0.062	26.878	0.081
		23	0.069	0.025	27.484	0.094
		24	0.142	0.058	30.126	0.068
		25	0.133	0.065	32.464	0.053
		26	0.046	0.079	32.743	0.066
		27	-0.001	0.071	32.743	0.086

Figura 5.40 Correlograma Modelo 3 Ejemplo 2

El modelo 4 se obtiene aumentando un coeficiente MA(10), los resultados de la estimación se muestran en la tabla 5.12 y su correlograma en la figura 5.41.

Tabla 5.12 Resultados Estimación Modelo 4

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	148.9089	46.01470	3.236116	0.0017
AR(12)	-0.400189	0.110955	-3.606774	0.0005
AR(1)	0.305079	0.065177	6.680794	0.0000
MA(13)	0.549029	0.103397	6.309899	0.0000
MA(10)	0.171311	0.119166	1.437583	0.1540
SIGMASQ	44213.01	5566.521	7.945520	0.0000

R-squared	0.294146	Mean dependent var	147.0833
Adjusted R-squared	0.254931	S.D. dependent var	251.5888
S.E. of regression	217.1648	Akaike info criterion	13.72006
Sum squared resid	4244449.	Schwarz criterion	13.88033
Log likelihood	-652.5627	Hannan-Quinn criter.	13.78484
F-statistic	7.501010	Durbin-Watson stat	1.976332
Prob(F-statistic)	0.000006		

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.004	0.004	0.0020	
		2	0.083	0.083	0.6867	
		3	-0.096	-0.097	1.6142	
		4	-0.087	-0.094	2.3908	
		5	-0.137	-0.123	4.3390	0.037
		6	-0.144	-0.144	6.5117	0.039
		7	-0.168	-0.179	9.4876	0.023
		8	-0.003	-0.029	9.4884	0.050
		9	-0.042	-0.083	9.6799	0.085
		10	0.076	-0.009	10.319	0.112
		11	0.065	-0.000	10.786	0.148
		12	0.050	-0.036	11.061	0.198
		13	0.023	-0.042	11.122	0.267
		14	0.059	0.024	11.517	0.319
		15	0.128	0.141	13.433	0.266
		16	-0.082	-0.080	14.218	0.287
		17	-0.109	-0.107	15.637	0.269
		18	-0.170	-0.141	19.114	0.161
		19	-0.102	-0.100	20.384	0.158
		20	0.009	0.017	20.394	0.203
		21	-0.031	-0.051	20.511	0.249
		22	0.118	0.064	22.284	0.220
		23	0.147	0.075	25.087	0.158
		24	0.142	0.073	27.729	0.116
		25	0.089	0.024	28.780	0.119
		26	0.063	0.067	29.306	0.136
		27	-0.009	0.066	29.317	0.170
		28	-0.071	0.012	30.010	0.184
		29	-0.096	0.009	31.301	0.179
		30	-0.098	-0.038	32.675	0.172
		31	-0.059	0.025	33.183	0.191
		32	-0.064	-0.003	33.781	0.208
		33	-0.112	-0.103	35.659	0.184
		34	0.051	0.002	36.057	0.206
		35	0.166	0.125	40.294	0.123
		36	0.199	0.180	46.520	0.047

Figura 5.41 Correlograma Modelo 4 Ejemplo 2

Se puede ver en la tabla 5.12 que el coeficiente MA(10) no es significativo, razón por la cual se lo debe eliminar.

Después de varias estimaciones, ya que tenemos estacionalidad en los datos, se ensayó el modelo 5. Su estimación se muestra en la tabla 5.13 y su correlograma en la figura 5.42.

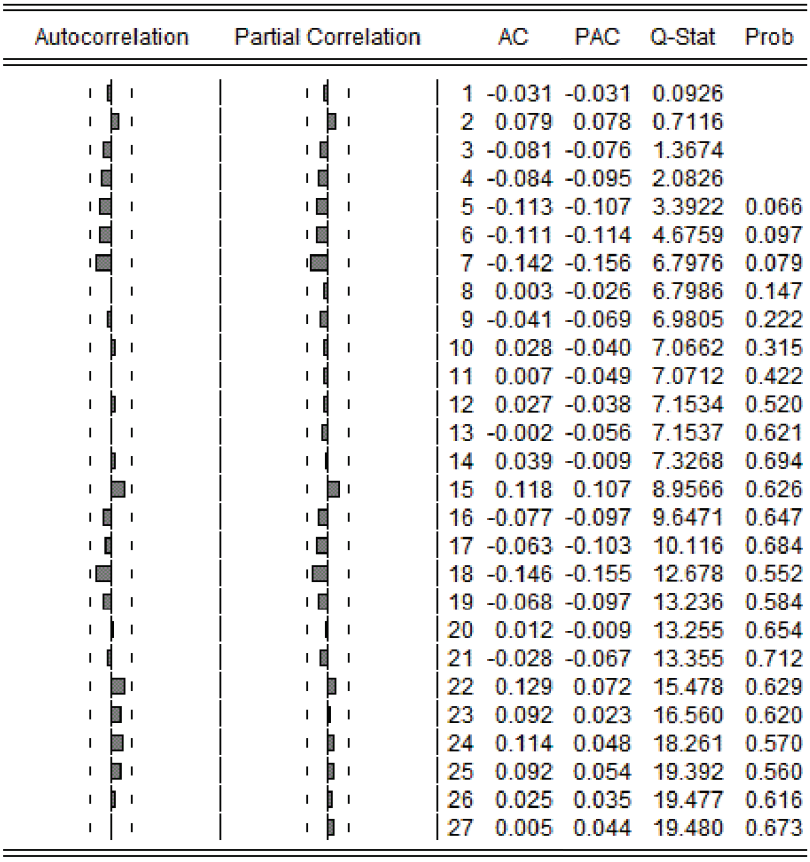
El modelo 5 matemáticamente se expresaría de la siguiente forma:

$$(1 - \emptyset_1 B - \emptyset_{10} B^{10})(1 - \Phi_{12} B^{12})(1 - B^{12})\tilde{Z}_t = (1 - \theta_{13} B^{13}) a_t \qquad (5.102)$$

Después de observar la tabla 5.13, se puede concluir que el modelo es adecuado en cuanto a coeficientes se refiere, además la figura 5.42 muestra un correlograma muy aproximado al ruido blanco.

Tabla 5.13 Resultados Estimación Modelo 5

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	149.0403	58.55185	2.545441	0.0126
AR(1)	0.314813	0.056089	6.612688	0.0000
AR(10)	0.234102	0.120803	1.937886	0.0558
SAR(12)	-0.420867	0.106145	-3.965005	0.0001
MA(13)	0.410549	0.112101	3.662331	0.0004
SIGMASQ	43608.53	5161.847	8.448241	0.0000
R-squared	0.303796	Mean dependent var		147.0833
Adjusted R-squared	0.265118	S.D. dependent var		251.5888
S.E. of regression	216.6751	Akaike info criterion		13.70142
Sum squared resid	4186418.	Schwarz criterion		13.86169
Log likelihood	-651.6680	Hannan-Quinn criter.		13.76620
F-statistic	7.854495	Durbin-Watson stat		2.043940
Prob(F-statistic)	0.000004			



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.031	-0.031	0.0926	
		2	0.079	0.078	0.7116	
		3	-0.081	-0.076	1.3674	
		4	-0.084	-0.095	2.0826	
		5	-0.113	-0.107	3.3922	0.066
		6	-0.111	-0.114	4.6759	0.097
		7	-0.142	-0.156	6.7976	0.079
		8	0.003	-0.026	6.7986	0.147
		9	-0.041	-0.069	6.9805	0.222
		10	0.028	-0.040	7.0662	0.315
		11	0.007	-0.049	7.0712	0.422
		12	0.027	-0.038	7.1534	0.520
		13	-0.002	-0.056	7.1537	0.621
		14	0.039	-0.009	7.3268	0.694
		15	0.118	0.107	8.9566	0.626
		16	-0.077	-0.097	9.6471	0.647
		17	-0.063	-0.103	10.116	0.684
		18	-0.146	-0.155	12.678	0.552
		19	-0.068	-0.097	13.236	0.584
		20	0.012	-0.009	13.255	0.654
		21	-0.028	-0.067	13.355	0.712
		22	0.129	0.072	15.478	0.629
		23	0.092	0.023	16.560	0.620
		24	0.114	0.048	18.261	0.570
		25	0.092	0.054	19.392	0.560
		26	0.025	0.035	19.477	0.616
		27	0.005	0.044	19.480	0.673

Figura 5.42 Correlograma Modelo 5 Ejemplo 2

Para realizar el pronóstico se debe escoger el mejor modelo, se hará una comparación entre los modelos 3,4, y 5, ver tabla 5.14.

Tabla 5.14 Comparación de Modelos Ejemplo 2

	Suma Residuos al Cuadrado	Log. Verosimilitud	AIC	BIC	HQC	DW
Modelo 3	3960228.00	-651.43	13.70	13.86	13.76	1.99
Modelo 4	4460860.00	-654.30	13.73	13.87	13.79	1.97
Modelo 5	4186418.00	-651.67	13.70	13.86	13.77	2.04

De acuerdo a la tabla 5.14 el modelo 3 tiene los mejores valores, ya que minimiza la suma de residuos al cuadrado, AIC, BIC y HQC y maximiza el logaritmo de

verosimilitud. El coeficeinte de Durbin Watson ronda el valor de 2, lo que garantiza que no exista correlación entre los residuos.

En cuanto a condiciones de estacionariedad e invertibilidad la figura 5.43 muestra el inverso de las raíces del modelo 3.

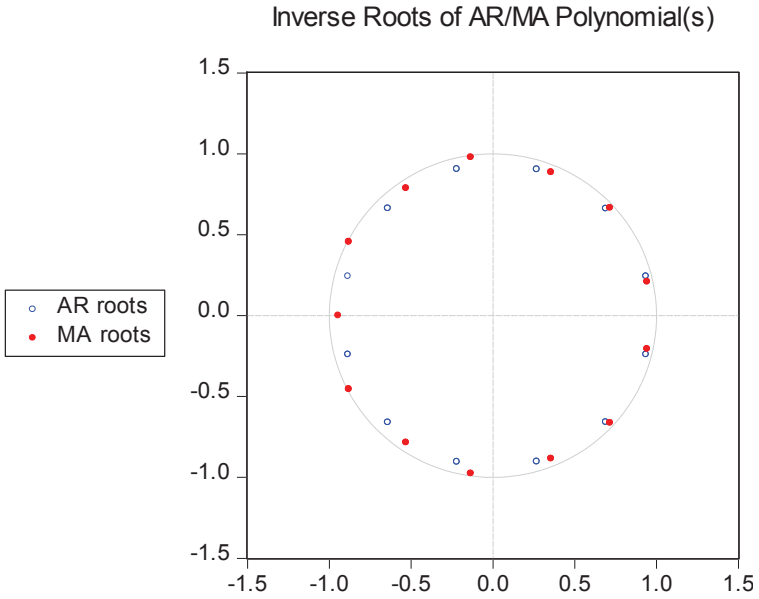


Figura 5.43 Inverso de las Raices Modelo 3 Ejemplo 2

Todos los valores se encuentran dentro del círculo unitario, por esta razón se puede concluir que el modelo es estacionario e invertible.

Se realizará el pronóstico con el modelo 3, que es el siguiente:

$$(1 - 0.306B + 0.471B^{12})\tilde{Z}_t = (1 - 0.252B^{7} + 0.658B^{13})a_t \qquad (5.103)$$

Los valores del pronóstico obtenido con este modelo se podrán comparar con los valores retenidos del año 2006, en la tabla 5.15 se muestran dichos valores.

En la figura 5.44 se muestra gráficamente los valores reales, el pronóstico realizado con este modelo y sus intervalos de confianza correspondientes (nivel de confianza 95%).

Tabla 5.15 Pronóstico, Valores Reales, Intervalo de Confianza, obtenido con el Modelo 3

Valores Reales	L_Inferior	Pronóstico	L_Superior
3899	3270.42	3739.74	3976.39
3217	3177.53	3637.22	3867.07
2148	2136.55	2620.97	2863.68
1134	792.39	1242.66	1467.80
348	-58.67	407.36	640.37
256	-293.01	237.53	502.80
180	-436.76	120.29	397.81
321	-88.49	416.11	668.41
1152	682.44	1153.01	1388.30
2681	2256.17	2722.65	2956.89
4511	3886.02	4376.60	4618.90
6327	6066.53	6569.39	6821.81

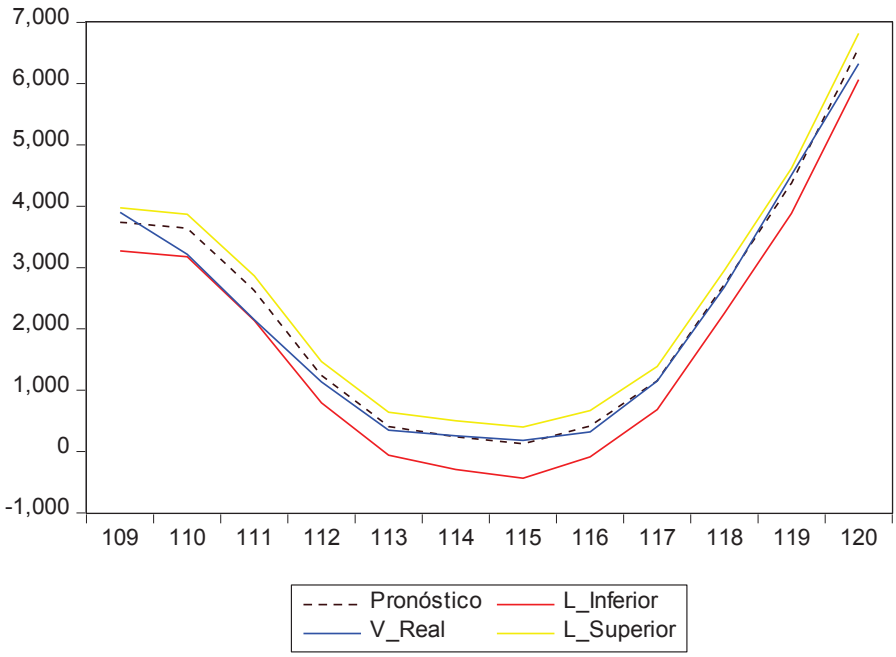


Figura 5.44 Pronóstico, Intervalo de Confianza y Valores Reales Modelo 3 Ejemplo 2

5.4 PRONÓSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRÁFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, CON LA METODOLOGÍA DE BOX - JENKINS

En la tabla 5.16 se muestra la información del consumo de placas digitales formato 510x400x0.15 (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo GTO_52, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utiliza los paquetes estadísticos EViews 9 y Minitab ver 17, que ayudarán en la evaluación de los modelos y pronóstico mediante la metodología ARIMA, para luego calcular el error de pronóstico MAPE y comparar con el resto de métodos al final de la investigación.

Tabla 5.16 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	16600	16750	15300	17400	32390	45300	33700
2	16600	18000	18750	20350	30400	39672	30660
3	26850	26500	17750	24950	36300	34000	37900
4	16250	22800	14400	23800	37200	36862	35885
5	18150	23000	16300	37600	37956	42900	36400
6	18550	22150	19100	25323	32044	39233	36190
7	22200	20650	16100	29050	37700	38025	33900
8	21250	22265	16400	35600	34526	36300	32400
9	19300	25230	17700	30350	38880	37680	38300
10	21400	23100	18200	37100	41982	41911	36000
11	19950	18700	15150	42260	47700	47457	39300
12	28850	22800	19000	40900	48464	47563	39300

En la figura 5.45 se muestran el gráfico de la demanda de placas digitales desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Se modelará la serie con los datos hasta el primer semestre del año 2015 (78 observaciones) y se reservarán los datos del segundo semestre del año 2015 (6 observaciones) para poder comparar con los pronósticos obtenidos por medio de la metodología ARIMA..

Para la modelación de esta serie de tiempo se utilizará el paquete estadístico Minitab ver. 17, tal como se hizo con el método de Holt – Winters en el capítulo anterior.

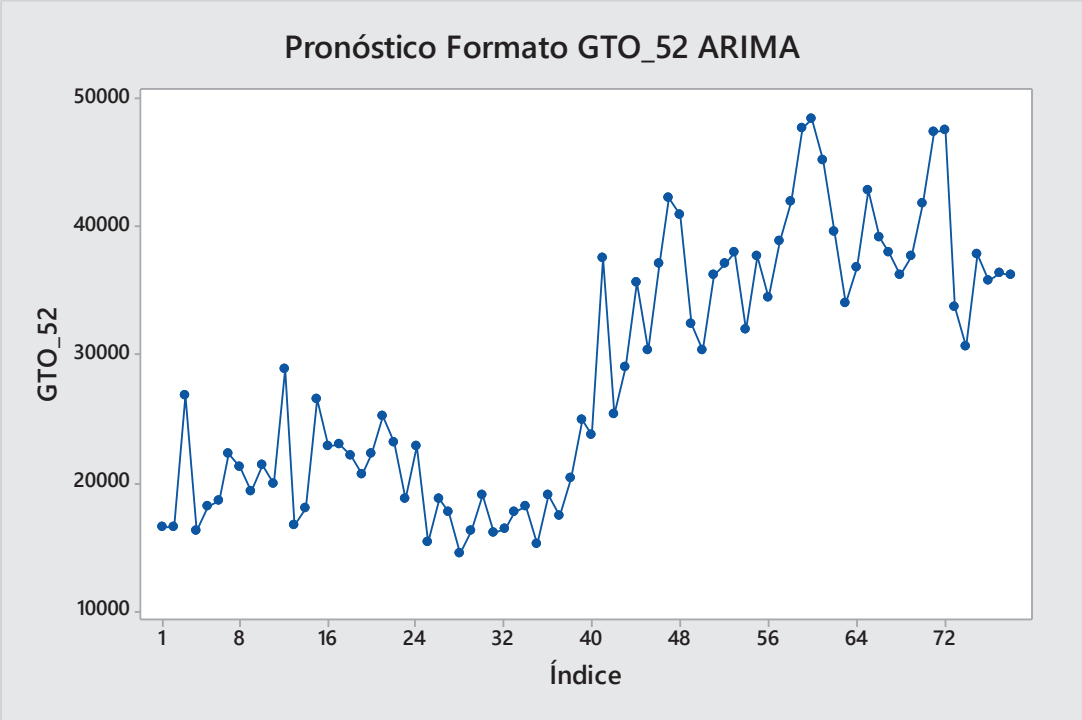


Figura 5.45 Consumo de placas digitales formato 510x400x0.15 (GTO_52) (2009 – 2015)

A la vista se puede saber que esta serie es NO estacionaria en su media, ya que se observan componentes de tendencia y estacionalidad. También se puede observar que la varianza no cambia de manera significativa.

Se pueden ver ciertos picos en los períodos 48, 60 y 72 que se verificarán más adelante con el correlograma.

Para comprobar esta sospecha se va a correr la prueba de Dickey Fuller Aumentada (DFA), mediante el programa EViews 9. Los resultados se muestran en la tabla 5.17.

El estadístico de DFA se encuentra en la zona de no rechazo de la hipótesis nula, por lo tanto aceptamos la hipótesis nula que existe una raíz unitaria, esto comprueba a su vez que la serie es no estacionaria, como se sospechaba.

Las figuras 5.46 y 5.47 son gráficos de las funciones de autocorrelación y autocorrelación parcial (sacf y spacf).

Tabla 5.17 Prueba DFA del Consumo de Placas Digitales Formato 510x400x0.15

Null Hypothesis: CONSUMO_GTO_52 has a unit root
Exogenous: Constant
Lag Length: 0 (Automatic - based on SIC, maxlag=11)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-2.384161	0.1495
Test critical values:	1% level	-3.517847	
	5% level	-2.899619	
	10% level	-2.587134	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(CONSUMO_GTO_52)
Method: Least Squares
Date: 06/26/16 Time: 19:21
Sample (adjusted): 2 78
Included observations: 77 after adjustments

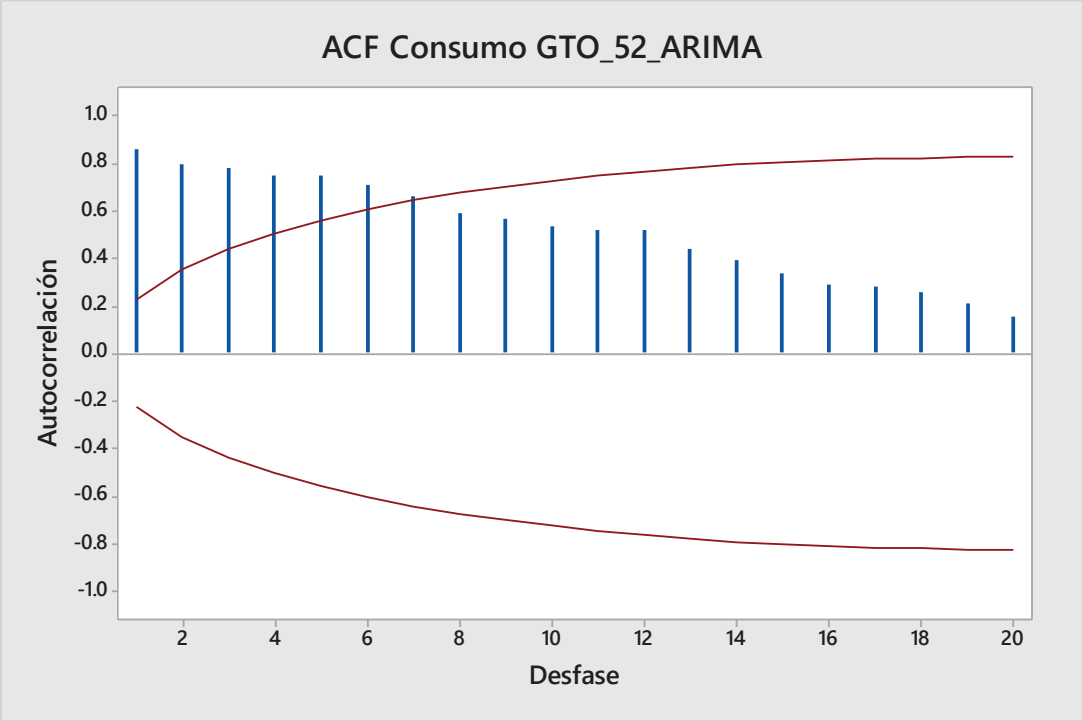


Figura 5.46 Autocorrelación Muestral del Consumo de placas digitales formato GTO_52

La función sacf corrobora la presencia de No estacionariedad, ya que decrece lentamente a cero.

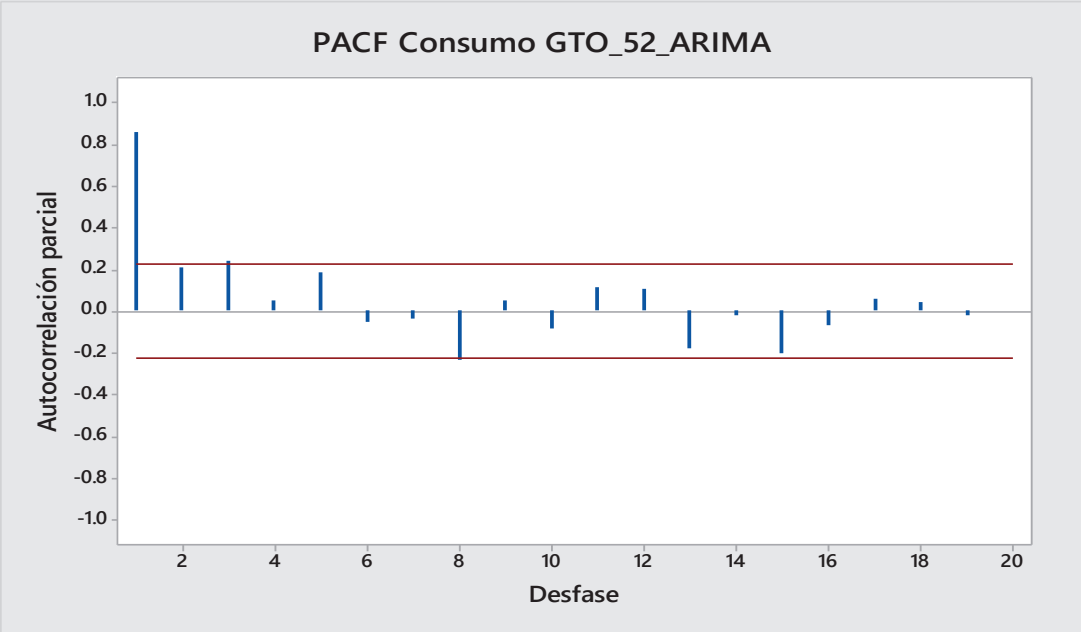


Figura 5.47 Autocorrelación Parcial del Consumo de placas digitales formato GTO_52

Se puede garantizar ahora sí que la serie es no estacionaria en la media, razón por la cual se aplicará una primera diferencia regular.

La figura 5.48 muestra el gráfico de la serie después de la primera diferencia regular.

Al observar la figura 5.48 se puede concluir que una diferencia regular es suficiente para volver a la serie de media estacionaria. Además se puede observar que los datos oscilan alrededor de una media aproximadamente cero, por esta razón no se utilizará la constante C en las diferentes simulaciones y estimaciones de parámetros.

La figura 5.49 muestra la función SACF de la serie de las primeras diferencias, esta es muy útil: primero para comprobar que esta serie ya es estacionaria en su media, ya que decrece rápidamente a cero, segundo los datos de esta serie son tomados cada mes por lo tanto se debe verificar las autocorrelaciones en los rezagos 4,8,12,24,…, para descartar algún patrón estacional.

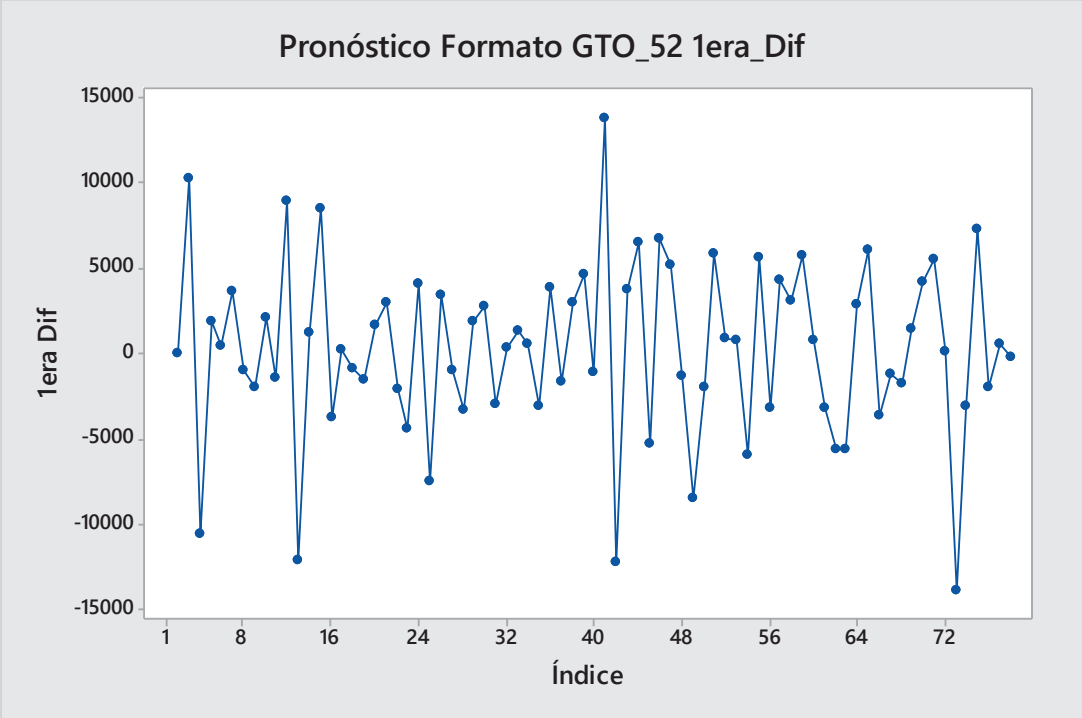


Figura 5.48 Primera diferencia de la serie: consumo de placas formato GTO_52

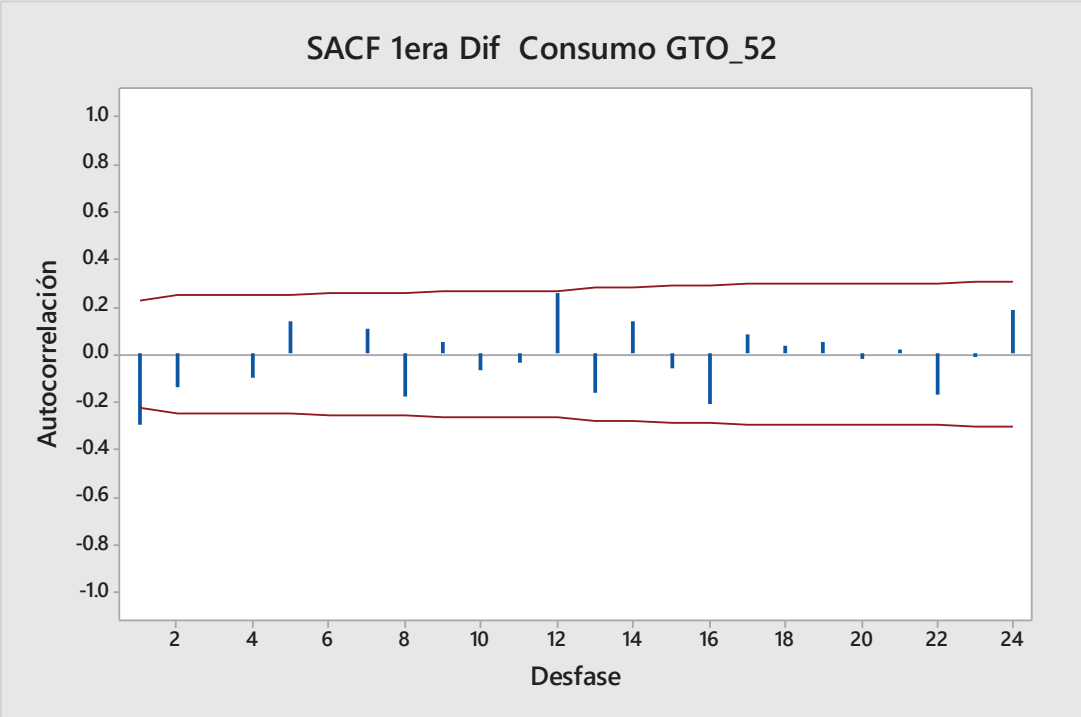


Figura 5.49 SACF Primera diferencia de la serie: consumo de placas formato GTO_52

Se puede observar un patrón estacional ya que existen picos en los rezagos 8,12,16 y 24 y tercero como los picos en los rezagos 12 y 24 se atenúan rápidamente también se puede garantizar que no es necesaria una diferencia estacional.

La figura 5.50 muestra la función SPACF de la serie de primeras diferencias del consumo de placas digitales formato GTO_52.

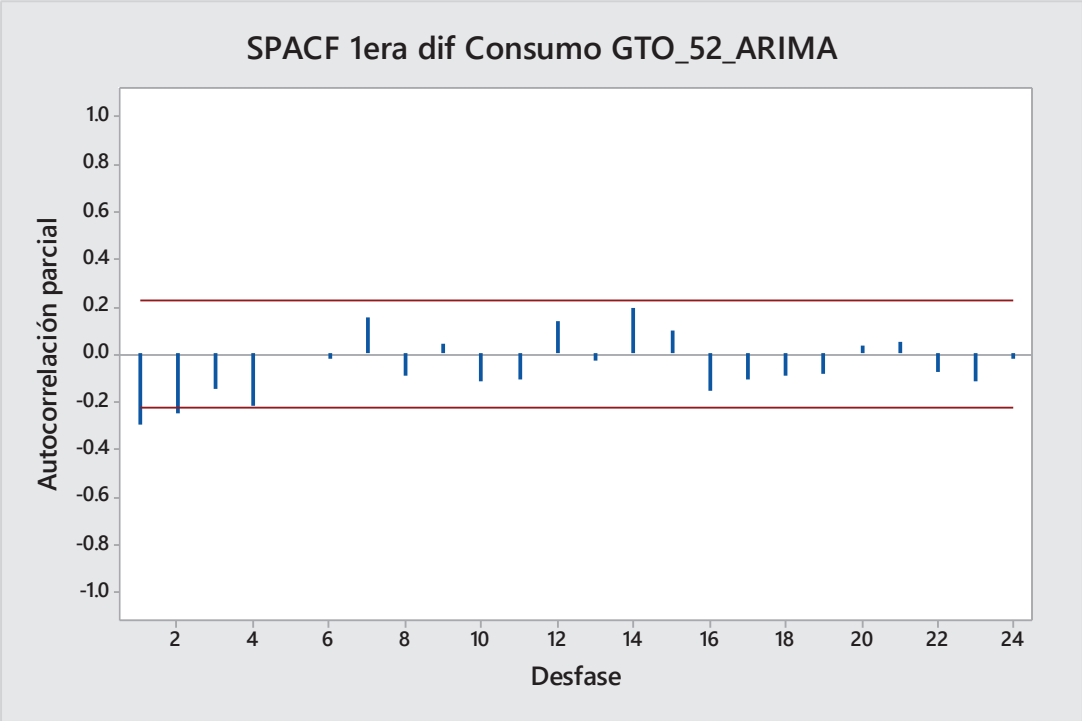


Figura 5.50 SPACF Primera diferencia de la serie: consumo de placas formato GTO_52

A continuación se tratará de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas sacf y spacf.

En base a las figuras 5.49 y 5.50 se puede sugerir un modelo 1 $ARIMA(2,1,0)(0,0,1)_{12}$ ya que la función spacf tiene dos picos significativos y el resto de valores ya no son significativos, mientras la función acf decrece cambiando el signo en los primeros rezagos. Además la función sacf sugiere un término MA(12) estacional puro ya que solamente tiene un pico significativo en el rezago 12 y no tiene valores significativos en la spacf.

Las figuras 5.51 y 5.52 muestran las funciones sacf y spacf de los residuos del modelo 1 $ARIMA(2,1,0)(0,0,1)_{12}$.

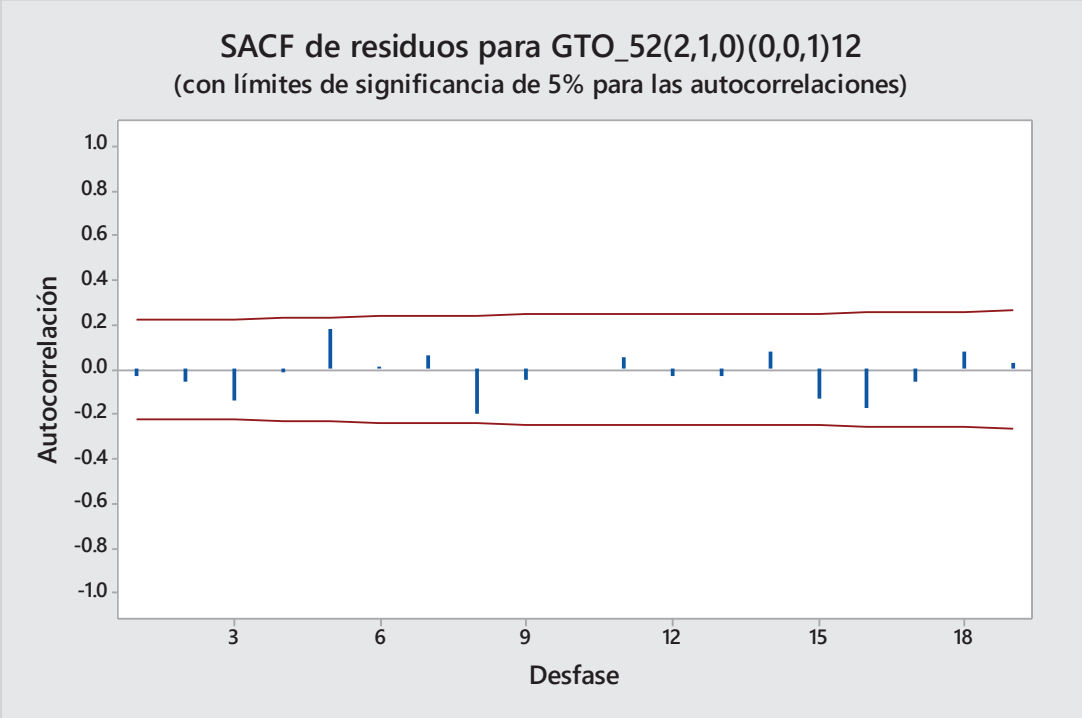


Figura 5.51 SACF de los residuos del Modelo 1 para el consumo de placas GTO_52

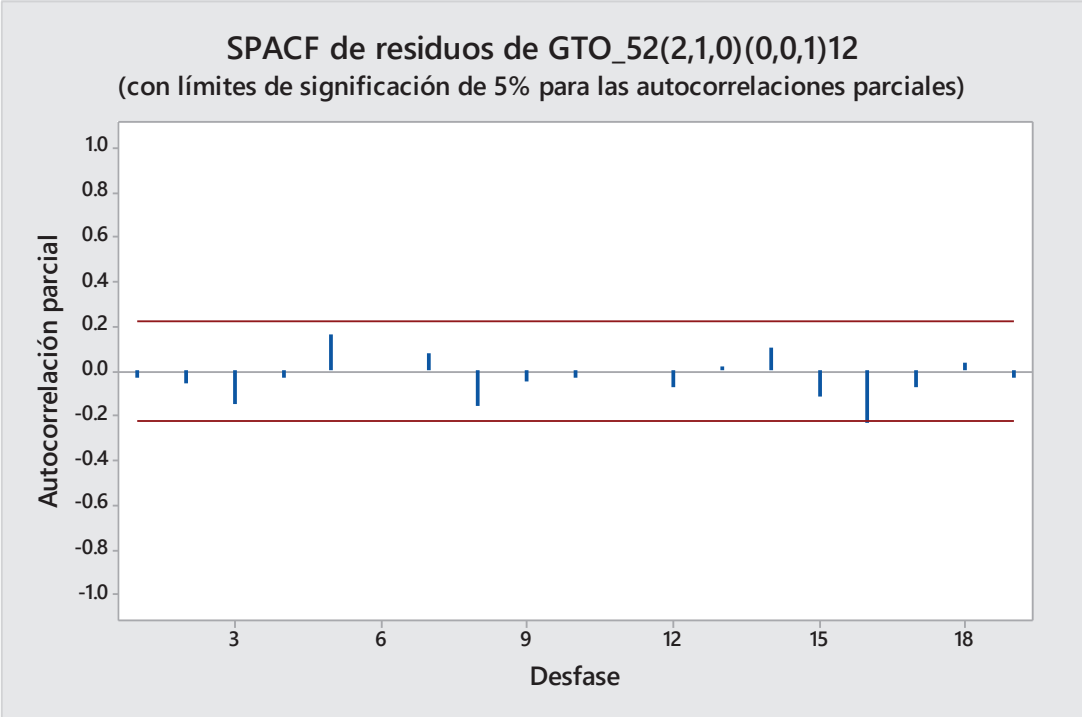


Figura 5.52 SPACF de los residuos del Modelo 1 para el consumo de placas GTO_52

La función sacf residual figura 5.52 es buena ya que prácticamente tiene un patrón de ruido blanco.

La que no satisface del todo es la función spacf residual figura 5.52 ya que tiene un pico en el rezago 16 que sale fuera de los límites de confianza.

Se debe pensar entonces en que un modelo mixto podría ser apropiado, entonces se necesita encontrar el valor de q adecuado.

Si regresamos a la figura 5.49, la función SACF de las primeras diferencias de la serie sugiere un valor de $q = 1$, ya que tiene un pico en el rezago 1, es decir se puede sugerir un modelo 2 $ARIMA(2,1,1)(0,0,1)_{12}$

Las figuras 5.53 y 5.54 muestran las funciones SACF y SPACF del modelo 2.

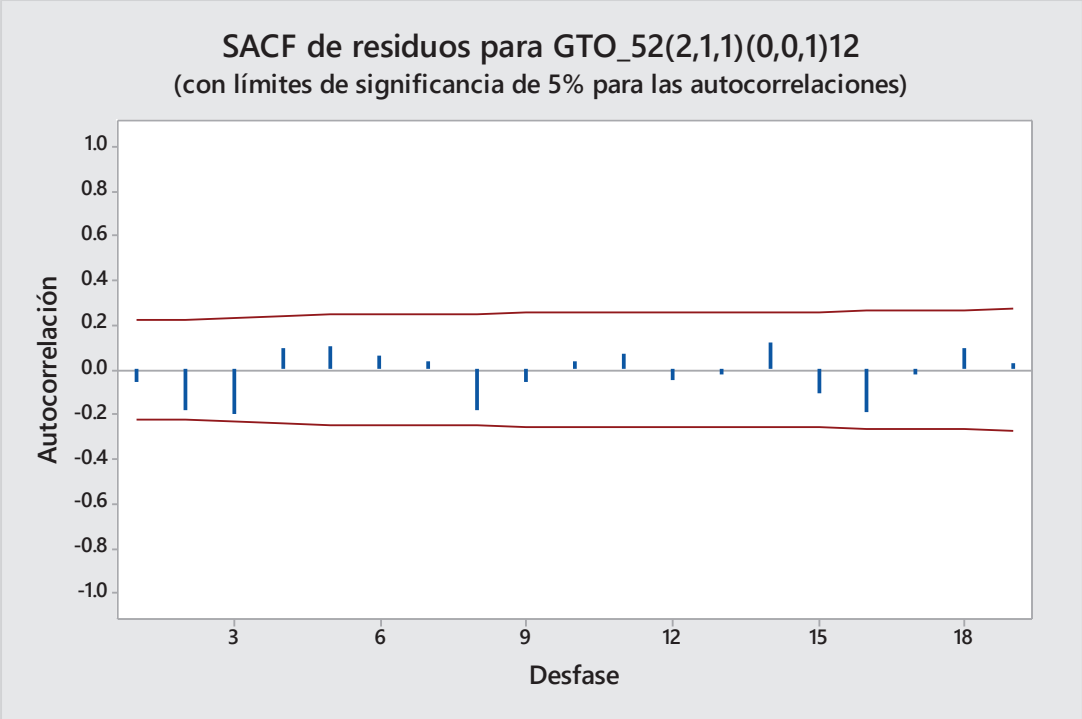


Figura 5.53 SACF de los residuos del Modelo 2 para el consumo de Placas GTO_52

La función sacf residual figura 5.54 es buena ya que prácticamente tiene un patrón de ruido blanco, pero los valores de las autocorrelaciones de los rezagos 2 y 3 son mayores que en el Modelo 1.

La función spacf residual figura 5.54 sigue sin ser satisfactoria, si bien el pico en el rezago 16 ya no sale fuera de los límites de confianza, se creó un pico en el rezago 3 que si sale de los límites de confianza.

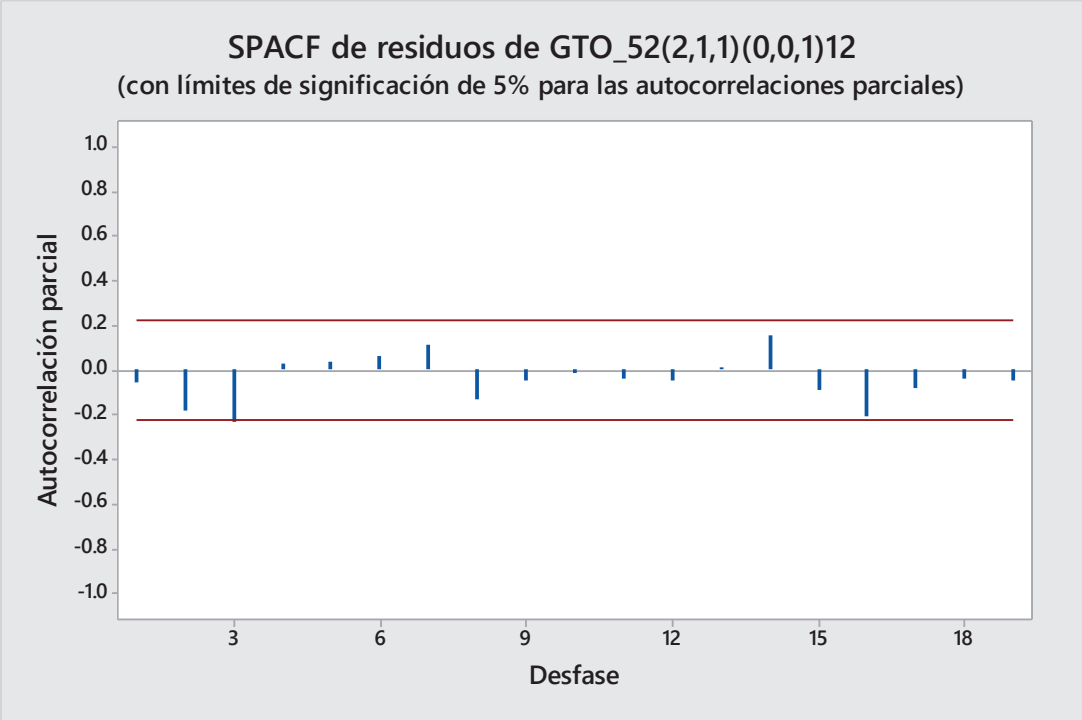


Figura 5.54 SPACF de los residuos del Modelo 2 para el consumo de placas GTO_52

La figura 5.54 da una pista de que debemos controlar los dos picos en los rezagos 2 y 3 que crecieron con el Modelo 2, razón por la cual ensayaremos el modelo 3 $ARIMA(2,1,2)(0,0,1)_{12}$.

Las figuras 5.55 y 5.56 muestran las funciones SACF y SPACF de los residuos del Modelo 3. Ambas funciones son satisfactorias.

Al igual que con el método de Holt – Winters, con cada uno de los modelos arriba indicados se realizará el pronóstico, se calcularán los errores y se elegirá el que presente el menor error.

En las tablas 5.18, 5.19 y 5.20 se muestran los pronósticos y sus errores de los modelos 1, 2, y,3.

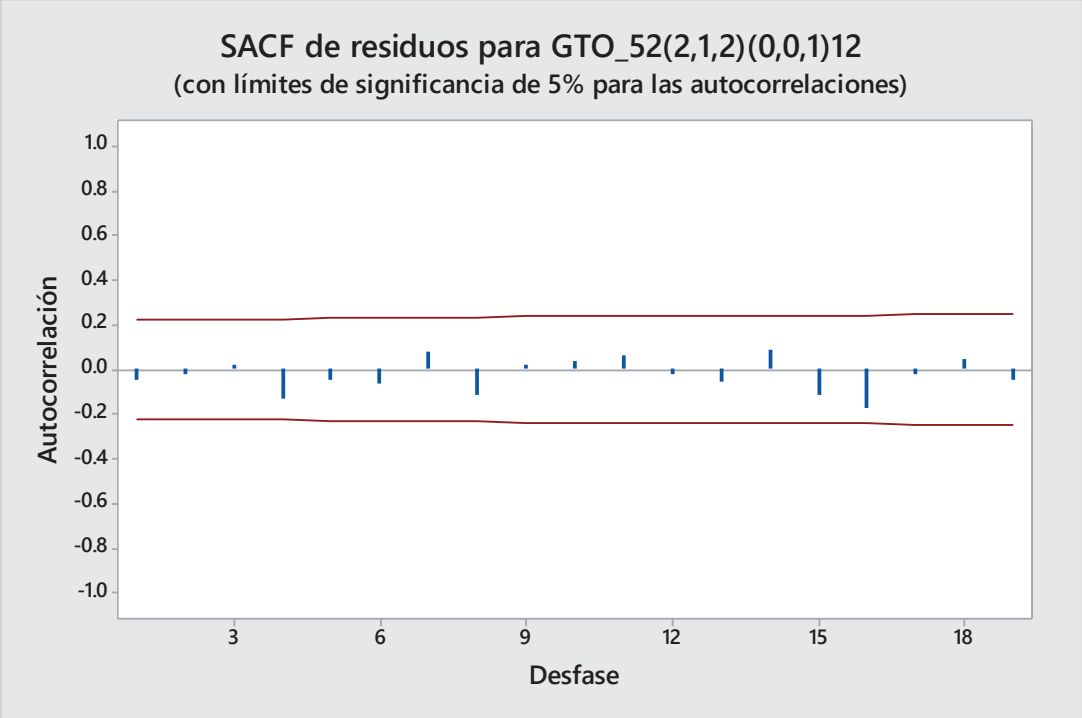


Figura 5.55 SACF de los residuos del Modelo 3 para el consumo de placas GTO_52

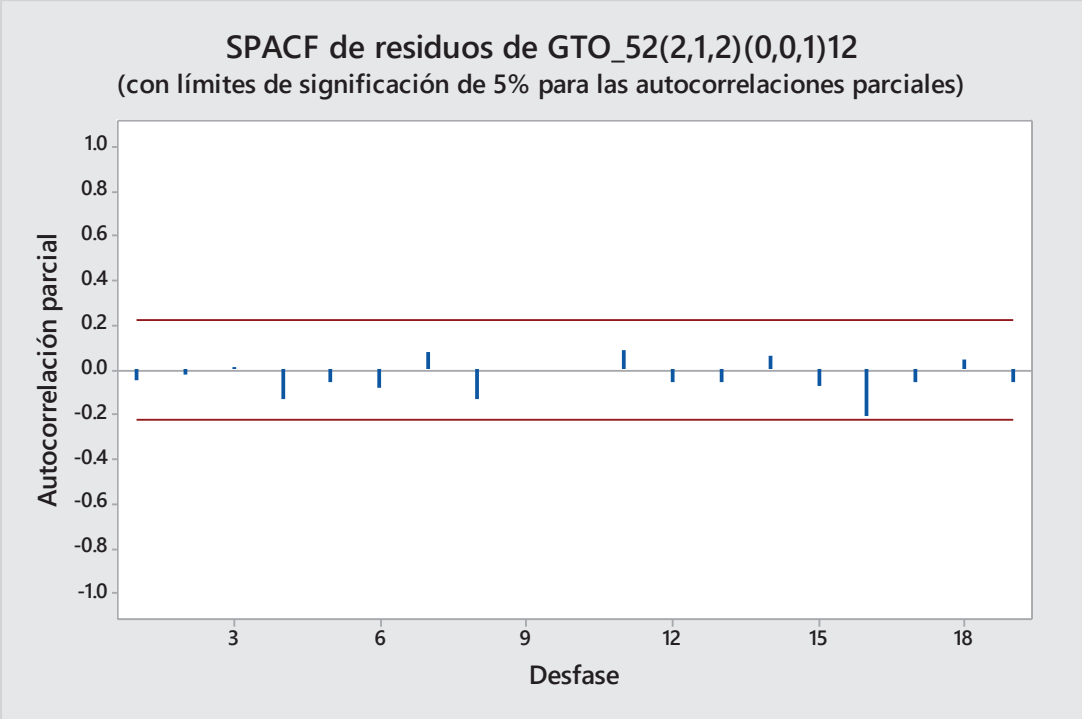


Figura 5.56 SPACF de los residuos del Modelo 3 para el consumo de placas GTO_52

Tabla 5.18 Pronósticos y errores Modelo 1

Mes	**Pronóstico Modelo 1**	Valores Reales	Error	(Error)^2	PEt	ABS PEt
79	35599	33900	-1699	2886601	-5.01%	5.01%
80	35241	32400	-2841	8071281	-8.77%	8.77%
81	34927	38300	3373	11377129	8.81%	8.81%
82	36540	36000	-540	291600	-1.50%	1.50%
83	38035	39300	1265	1600225	3.22%	3.22%
84	37809	39300	1491	2223081	3.79%	3.79%
			MSE	**4408319.5**	**MAPE**	**5.18%**
			RMSE	**2099.60**		

Tabla 5.19 Pronósticos y errores Modelo 2

Mes	**Pronóstico Modelo 2**	Valores Reales	Error	(Error)^2	PEt	ABS PEt
79	34682	33900	-782	611524	-2.31%	2.31%
80	35025	32400	-2625	6890625	-8.10%	8.10%
81	34766	38300	3534	12489156	9.23%	9.23%
82	35957	36000	43	1849	0.12%	0.12%
83	37435	39300	1865	3478225	4.75%	4.75%
84	37245	39300	2055	4223025	5.23%	5.23%
			MSE	**4615734**	**MAPE**	**4.95%**
			RMSE	**2148.43**		

Tabla 5.20 Pronósticos y errores Modelo 3

Mes	**Pronóstico Modelo 3**	Valores Reales	Error	(Error)^2	PEt	ABS Pet
79	33835	33900	65	4225	0.19%	0.19%
80	34525	32400	-2125	4515625	-6.56%	6.56%
81	35761	38300	2539	6446521	6.63%	6.63%
82	37905	36000	-1905	3629025	-5.29%	5.29%
83	39094	39300	206	42436	0.52%	0.52%
84	38200	39300	1100	1210000	2.80%	2.80%
			MSE	**2641305.33**	**MAPE**	**3.67%**
			RMSE	**1625.21**		

Como se esperaba el modelo 3 presenta los menores errores.

A continuación se chequeará el resto de criterios para ver si el modelo 3 es estadísticamente adecuado.

Todos los coeficientes del Modelo 3 son significativos ya que tienen valores del estadístico t mayores a 2 ($t > 2$) o los p valores de la probabilidad son menores a 0.05 ($p < 0.05$), como se muestra en la tabla 5.21.

En la tabla 5.22 se pueden ver los valores del estadístico Q^* de Ljung – Box, todos ellos se han chequeado con la tabla 2 (Distribución χ^2) del Anexo C y son adecuados (menor que los valores al 0.05).

Tabla 5.21 Coeficientes del Modelo 3

Tipo	Coeficiente	Error Std. Coef	Estadístico t	p-valor
AR 1	0.8343	0.1997	6.18	0.000
AR 2	-0.6395	0.1429	-6.48	0.000
MA 1	1.2333	0.1867	6.61	0.000
MA 2	-0.7796	0.1465	-6.32	0.000
SMA 12	-0.34	0.1311	-2.59	0.011

Tabla 5.22 Estadístico de Ljung – Box Q^* para el Modelo 3

Estadística Chi-cuadrada modificada de Box-Pierce (Ljung-Box)				
Desfase	12	24	36	48
Chi-cuadrada	6.4	13.5	32.1	38.8
GL	7	19	31	43
Valor p	0.731	0.810	0.411	0.653

Tabla 5.23 Matriz de Correlación de los parámetros estimados para el Modelo 3

Matriz de correlación de los parámetros estimados			
	1	2	3
2	-0.306		
3	0.856	0.033	
4	-0.834	0.423	-0.809

Además la matriz de correlación absoluta de los coeficientes estimados (Tabla 5.23) no muestra ningún valor superior a 0.9, es decir no son altamente correlacionados por lo tanto garantiza que los coeficientes estimados serán estables (no dependerán de esta realización en particular).

Los coeficientes del componente AR cumplen con las condiciones de estacionariedad, estas son:

$|\emptyset_2| < 1 \ (0.6396 < 1);$

$(\emptyset_2 + \emptyset_1) < 1 \ (-0.6396 + 0.8343) = 0.1947 < 1;$

$(\emptyset_2 - \emptyset_1) < 1 \ (-0.6396 - 0.8343) = -1.4739 < 1.$

Los coeficientes del componente MA cumplen con las condiciones de invertibilidad:

$|\theta_2| < 1 \ (0.7796 < 1);$

$(\theta_2 + \theta_1) < 1 \ (-0.7796 + 1.2233) = 0.4437 < 1;$

$(\theta_2 - \theta_1) < 1 \ (-0.7796 - 1.2233) = -2.0029 < 1.$

Por último el coeficiente estacional SMA(12) cumple con la condición de invertibilidad:

$|\Theta_{12}| < 1 \ (0.34 < 1).$

Después de chequear todos los criterios anteriores se puede concluir que el Modelo 3 es estadísticamente adecuado.

El Modelo 3 quedaría:

$$(1 - 0.8343B + 0.6395B^2)(1 - B)\tilde{Z}_t = (1 + 0.34B^{12})(1 - 1.2233B + 0.7796B^2)a_t \qquad (5.104)$$

Su pronóstico se presenta nuevamente en la tabla 5.24, este modelo será el representante de esta metodología.

Tabla 5.24 Pronósticos y errores Modelo 3

Mes	**Pronóstico Modelo 3**	Valores Reales	Error	(Error)^2	PEt	ABS Pet
79	33835	33900	65	4225	0.19%	0.19%
80	34525	32400	-2125	4515625	-6.56%	6.56%
81	35761	38300	2539	6446521	6.63%	6.63%
82	37905	36000	-1905	3629025	-5.29%	5.29%
83	39094	39300	206	42436	0.52%	0.52%
84	38200	39300	1100	1210000	2.80%	2.80%
			MSE	**2641305.33**	**MAPE**	**3.67%**
			RMSE	**1625.21**		

6 REDES NEURONALES

6.1 INTRODUCCION

En el capítulo anterior se había aclarado que la tecnología de Box – Jenkins produce un *pronóstico óptimo*, es decir ningún otro pronóstico de una sola variable tiene un error medio cuadrático (MSE) más pequeño, siempre y cuando se consideran solamente modelos de una sola variable con combinaciones *lineales* de *su pasado* y con coeficientes fijos. Es posible que combinaciones *no lineales* de la variable podrían producir pronósticos con un menor MSE que el modelo ARIMA lineal. Por esta razón se analizará en este capítulo un método muy popular en el análisis moderno de datos, el pronóstico de series de tiempo mediante *redes neuronales*.

Una red neuronal puede ser considerada como una técnica de procesamiento de datos que relaciona (o mapea) algún tipo de flujo de información de entrada con un flujo de datos de salida.

En aplicaciones de series de tiempo de interés aquí, la entrada podría ser una serie de tiempo unidimensional y la salida podría ser el mejor estimado de los siguientes valores en la serie. En general las tareas de las redes neuronales pueden ser divididas en cuatro tipos de aplicaciones distintas: *Clasificación, Asociación, Codificación y Simulación*.

Una tarea particular podría necesitar de una o varias funciones como las arriba mencionadas. Por ejemplo el pronóstico de series de tiempo mediante un perceptrón multicapa podría necesitar de un elemento de compresión de los datos de entrada (codificación), un elemento de reducción de ruido (asociación), la capacidad de detectar patrones recurrentes que conduzcan a un comportamiento predictivo (clasificación) y el pronóstico de un evento único (simulación).

Algunas capacidades únicas de las redes neuronales se podrían resumir como: *Generalización, Flexibilidad y Modelamiento no lineal*. (Azoff, 1994).

Los pronósticos han sido dominados por modelos lineales durante décadas. Estos modelos son relativamente fáciles de desarrollar e implementar y son simples de

entender e interpretar. Sin embargo, los modelos lineales tienen serias limitaciones ya que ellos no pueden capturar relaciones no lineales en los datos. La aproximación de modelos lineales a relaciones no lineales complicadas, comunes en el mundo real, no siempre es satisfactoria.

Las redes neuronales son una alternativa prometedora para los pronósticos. Su estructura inherentemente no lineal es muy útil para capturar las relaciones complejas en muchos problemas del mundo real.

Las redes neuronales tiene varias características que son muy valiosas para las tares de pronósticos. En primer lugar, las redes neuronales al ser métodos no paramétricos en el manejo de datos no requieren muchas asunciones restrictivas de cómo los datos fueron generados. Al aprender de los datos o la experiencia, las redes neuronales son muy valoradas en varias situaciones de pronóstico donde los datos son fáciles de recolectar pero el mecanismo de generación es desconocido o no es pre-especificado. En segundo lugar, ha sido demostrado matemáticamente que las redes neuronales son aproximadores universales de funciones, esta es una poderosa característica ya que el objetivo de cualquier método de pronóstico es capturar de una manera precisa la relación funcional entre la variable a ser predicha y otras variables relevantes. Esta combinación de características arriba mencionadas hace que las redes neuronales sean una herramienta muy general y flexible en el área de pronósticos. Este campo de pronósticos mediante redes neuronales ha tenido un gran progreso durante la última década, no será de sorprenderse que seguirá este gran avance y éxito en la siguiente década. (Zhang, 2004).

6.1.1 HISTORIA

Antes de empezar con la historia, cabe destacar que cuando se utiliza el término neurona no nos referiremos a las neuronas biológicas sino a neuronas artificiales consideradas como elementos de un programa o quizás de un circuito integrado. Redes de estas neuronas artificiales no tienen ni una fracción del poder del cerebro humano, pero pueden ser entrenadas para ejecutar importantes funciones. (Hagan M., 2015).

El término *nodo* se utilizará para describir una celda de entrada a la red neuronal, y el término *neurona* describirá una celda de procesamiento (o cálculo).

Algunos de los trabajos de investigación en el campo de redes neuronales ocurrieron a finales de siglo 19 y a inicios del siglo 20.

El punto de vista moderno de las redes neuronales se inicia en 1940 con el trabajo de Warren McCulloch and Walter Pitts, quienes mostraron que redes neuronales podrían en principio, calcular cualquier función aritmética o lógica. Este trabajo es conocido como el origen en el campo de las redes neuronales.

Después de McCulloch y Pitts, Donald Hebb propuso que el condicionamiento clásico (descubierto por Pavlov) está presente debido a las propiedades de las neuronas individuales. El propuso un mecanismo de aprendizaje para neuronas biológicas.

La primera aplicación práctica de las redes neuronales se dio a finales de 1950, con la invención del perceptrón simple y su regla de aprendizaje asociada por Frank Rosenblatt. Rosenblatt y sus colegas que construyeron el perceptrón simple demostraron su habilidad para ejecutar el reconocimiento de caracteres. Este acontecimiento generó un gran interés en el campo de las redes neuronales, desafortunadamente más tarde se demostró que el perceptrón simple podía resolver solamente una limitada clase de problemas.

Casi al mismo tiempo, Bernard Widrow y Ted Hoff introdujeron un nuevo algoritmo de aprendizaje que fue utilizado para entrenar redes neuronales adaptivas, las cuales eran de estructura y capacidad similar a las del perceptrón simple. Desafortunadamente estas redes neuronales adaptivas sufrían de la misma limitación que el perceptrón simple, las mismas que fueron ampliamente comentadas en el libro de Marvin Minsky y Seymour Papert . Rosenblat y Widrow conscientes de esta limitación propusieron nuevas redes que superaban esta limitación, pero no fueron capaces de modificar el algoritmo de aprendizaje para entrenar estas redes complejas.

Mucha gente influenciada por Minsky y Papert, creyeron que la investigación en el campo de las redes neuronales había muerto. Esto combinado con el hecho de que no existían computadores digitales poderosos para experimentar, causó que muchos investigadores se apartaran del tema por aproximadamente una década.

El interés en el campo de las redes neuronales a finales de 1960 fue reducido debido a la carencia de nuevas ideas y el débil poder de los computadores para experimentar.

Algunos trabajos importantes continuaron en 1970. En 1972 Teuvo Kohonen y James Anderson independientemente y separadamente desarrollaron algunas redes neuronales que podían actuar como memorias. Stephen Grossberg estuvo muy activo durante este período en la investigación de redes auto-organizadas.

Durante 1980 estos impedimentos antes mencionados fueron superados y la investigación en el campo de las redes neuronales se incrementó drásticamente. Nuevos computadores personales que crecieron rápidamente en capacidad se volvieron ampliamente disponibles, adicionalmente nuevos conceptos fueron introducidos.

Dos nuevos conceptos fueron responsables del renacimiento de las redes neuronales. El primero fue el uso de la estadística para explicar la operación de cierta clase de redes recurrentes, las cuales podrían ser usadas como memorias asociativas descritas por John Hopfield.

El segundo fue el desarrollo clave del algoritmo de retropropagación (Backpropagation) para entrenar al perceptrón multicapa (MLP), el cual fue descubierto independientemente por varios investigadores. El trabajo más influyente fue el realizado por David Rumelhard y James McClelland. Este algoritmo fue la respuesta a la crítica de Minsky y Papert hecha en 1960.

Estos nuevos desarrollos revigorizaron el campo de las redes neuronales. Ya que en 1980 miles de investigaciones fueron desarrolladas y las redes neuronales encontraron un sinfín de aplicaciones.

Así Barto, Sutton y Anderson en 1983 presentaron su trabajo de *aprendizaje reforzado*. Aunque no fueron los primeros en utilizar aprendizaje reforzado (Minsky realizó su tesis doctoral Ph.D acerca de este tema en 1954), esta investigación generó mucho interés en el aprendizaje reforzado y sus aplicaciones en control.

En 1984 el libro de Braitenberg *Vehículos: Experimentos en Psicología sintética* fue publicado. En este libro Braitenberg aboga por el principio de *objetivo–dirigido y ejecución auto-organizada.*

En 1986 el desarrollo del *algoritmo de retro-propagación* fue reportado por Rumelhart, Hinton y Williams. En el mismo año se publicaban dos volúmenes del libro *Procesamiento distribuido y paralelo* editado por Rumelhart y McClelland. Este libro ha sido la mayor influencia en el uso del aprendizaje por retropropagación que se convirtió en el algoritmo de aprendizaje más popular para el entrenamiento del perceptrón multicapa.

En 1988 Linsker describe un nuevo principio de autoorganización en redes perceptivas, entre otras publicaciones.

Las redes neuronales serán claramente importantes herramientas en el campo de la ingeniería y las matemáticas. Adicionalmente, recuerde que conocemos muy poco del funcionamiento de nuestro cerebro. Los avances más importantes en el campo de las redes neuronales ciertamente se darán en el futuro. (Hagan M., 2015), (Haykin, 1999).

6.1.2 ARQUITECTURA

La manera en la cual las neuronas de una red neuronal están estructuradas está íntimamente ligado con el algoritmo de aprendizaje utilizado para el entrenamiento de la red. En general se pueden identificar tres clases fundamentales de arquitecturas de las redes neuronales:

6.1.2.1 Redes Unidireccionales de una Sola Capa

En una red neuronal las neuronas están organizadas en forma de capas. En la forma más simple de una red, se tiene una capa de entrada de nodos de origen que se proyectan a una capa de salida de neuronas (celdas de cálculo o procesamiento), pero no viceversa. En otras palabras la red es estrictamente *unidireccional*. En la figura 6.1 se muestra el caso de una red de cuatro nodos en la capa de entrada y cuatro neuronas en la de salida. Esta red se denomina *red de una sola capa*, con la denominación de "*una sola capa*" refiriéndose a la capa de salida de celdas de cálculo (neuronas). No se toma en cuenta la capa de entrada con los nodos de entrada ya que ningún cálculo es ejecutado allí.

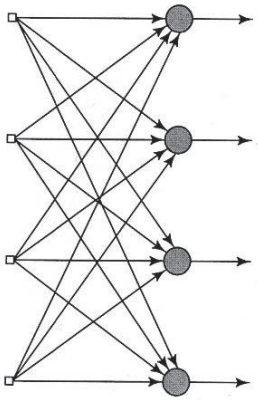


Figura 6.1 Red Unidireccional de una sola capa

((Haykin, 1999) Pág. 21)

6.1.2.2 Redes Unidireccionales Multicapa

En la segunda clase de redes neuronales unidireccionales existe la presencia de una o varias capas ocultas, cuyas celdas de cálculo se denominan neuronas ocultas o unidades ocultas. La función de las neuronas ocultas es intervenir entre la entrada externa y la salida de la red de alguna manera útil. Con la adición de una o más capas, la red es habilitada para poder extraer estadísticas de mayor orden.

Los nodos de origen en la capa de entrada de la red constituyen las señales de entrada aplicadas a las neuronas en la segunda capa (es decir la primera capa oculta). Las señales de salida de la segunda capa son usadas como entradas de la tercera capa y así hasta el final de la red. El conjunto de señales de salida de las neuronas en la capa de salida (final) constituyen la respuesta general de la red a las señales de activación suministradas por los nodos de origen en la capa de entrada (primera capa).

La figura 6.2 muestra una red unidireccional multicapa con una capa escondida. A una red como la de la figura 6.2 se le denominará como red 10-4-2 ya que tiene 10 nodos de origen, 4 neuronas ocultas y 2 neuronas de salida.

Además se dice que la red de la figura 6.2 está *totalmente conectada* en el sentido que cada celda en cada capa de la red está conectado a cada uno de las celdas de la capa adyacente hacia adelante. Cuando no existen algunas conexiones en la red se dice que la red es *parcialmente conectada*.

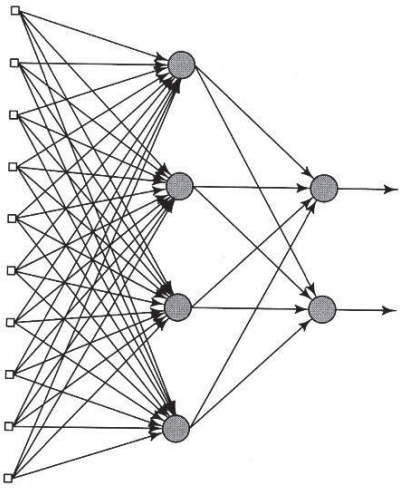


Figura 6.2 Red Unidireccional Multicapa

((Haykin, 1999) Pág. 22)

6.1.2.3 Redes Recurrentes

Las redes neuronales recurrentes se diferencian de las redes neuronales unidireccionales en que tienen al menos un lazo de realimentación.

En la figura 6.3 se muestra una red recurrente que consiste en una sola capa de neuronas con cada neurona realimentando su señal hacia las entradas de otras neuronas. En la figura 6.3 no existe lazos de *autorealimentación*, donde autorealimentación significa que la salida de una neurona es realimentada hacia su propia entrada.

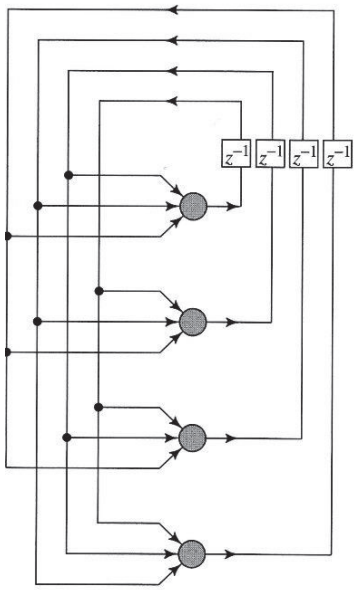


Figura 6.3 Red Recurrente sin autorealimentación y sin neuronas ocultas

((Haykin, 1999) Pág. 23)

En la figura 6.4 se muestra otro tipo de red recurrente con neuronas ocultas. La realimentación se origina desde neuronas ocultas como también desde las neuronas de salida de la red.

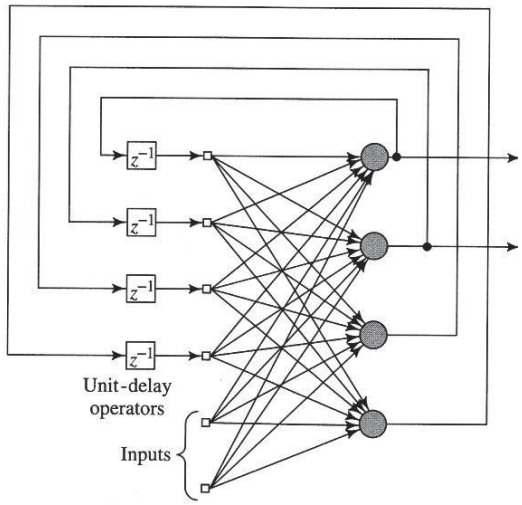


Figura 6.4 Red Recurrente con neuronas ocultas

((Haykin, 1999) Pág. 24)

La presencia de lazos de realimentación ya sea en la red recurrente de la figura 6.3 o de la figura 6.4, tiene un profundo impacto en las capacidades de aprendizaje de la red y en su ejecución. Sin embargo, los lazos de realimentación están vinculados al uso de *unidades de retardo* (representados por Z^{-1}). (Haykin, 1999).

6.1.3 APLICACIONES

Existe un amplio rango de aplicaciones de las redes neuronales actualmente. Las aplicaciones se han expandido ya que las redes neuronales son buenas resolviendo problemas, no solamente en el campo de la ingeniería, ciencias y matemáticas, sino en medicina, negocios, finanzas y literatura también.

Su aplicación a una amplia variedad de problemas en muchos campos las hace muy atractivas. Además, computadores más rápidos y algoritmos más eficientes han hecho posible el uso de redes neuronales en resolución de problemas industriales complejos que requieren mucha más velocidad de cálculo.

Aquí algunos ejemplos: Un instituto de investigación Italiano las utiliza para probar la pureza del aceite de oliva, Google las utiliza para etiquetado de imágenes (automáticamente identifica una imagen y le asigna una tecla), Microsoft ha desarrollado redes neuronales para convertir palabras habladas en inglés a palabras habladas en chino, Investigadores de la universidad de Lund y del Hospital Universitario Skane en Suiza han utilizado redes neuronales para mejorar la compatibilidad a largo plazo en los trasplantes de corazón, identificando coincidencias óptimas entre donadores y receptores. Otros campos en los que se están utilizando redes neuronales son: Aeroespacial, Automovilístico, Banca, Defensa, Electrónica, Seguros, Seguridad, Robótica, Pronósticos, Finanzas, etc.

Una lista completa de aplicaciones se la puede encontrar en la referencia (Hagan M., 2015) pp:1-5,1-6.

Se puede concluir que el número de aplicaciones de las redes neuronales, el dinero que ha sido invertido y el interés en este tema son enormes. (Hagan M., 2015).

En el presente capítulo se estudiará el fundamento teórico de las redes neuronales que incluirá: el perceptrón multicapa MLP (del Inglés Multi Layer Perceptron), el método de entrenamiento denominado retropropagación (Backpropagation) y sus

variaciones como el algoritmo de Levenberg – Marquardt y regularización Bayesiana, luego generalización de las redes neuronales, redes recurrentes o dinámicas aplicadas a los pronósticos. A continuación se realizará un ejemplo académico del uso de redes neuronales en la predicción de series de tiempo y finalmente la predicción de la demanda de placas digitales en el mercado gráfico quiteño desde el año 2009 hasta el 2015 mediante redes neuronales, utilizando el módulo de redes neuronales y programas desarrollados por el autor bajo la plataforma de MatLab ver. 15.

6.2 FUNDAMENTO TEÓRICO DE LAS REDES NEURONALES

6.2.1 MODELO DE NEURONA

6.2.1.1 Notación

Desafortunadamente no existe una notación universalmente aceptada para las redes neuronales, esto ha impedido la difusión de nuevas ideas en este campo. Se utilizará la siguiente notación para las ecuaciones matemáticas y figuras del presente trabajo, la misma que es utilizada por la referencia (Hagan M., 2015) que son los autores del módulo de redes neuronales del programa MatLab.:

Escalares.- letras *itálicas* minúsculas: *a,b,c*

Vectores.- Letras minúsculas no itálicas **negritas**: **a,b,c**

Matrices.- Letras Mayúsculas no itálicas **NEGRITAS**: **A,B,C**

6.2.1.2 Neurona con una sola entrada

En la figura 6.5 se muestra una neurona con una sola entrada.

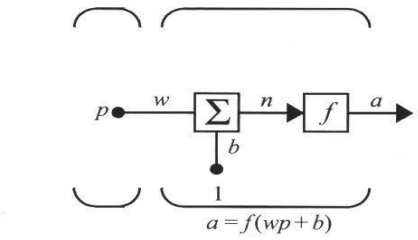


Figura 6.5 Neurona con una sola entrada

((Hagan M., 2015) Pág. 2-3)

El escalar p de entrada es multiplicado por su $peso\ w$ y forma wp , que es uno de los términos del sumador. La otra entrada 1 es multiplicada por su $bías\ b$ que es el otro término del sumador. La salida n del sumador pasa a través de la función de transferencia f , la cual produce la salida escalar de la neurona a.

Algunos autores utilizan los términos “función de activación” en lugar de *función de transferencia* y “offset” en lugar de $bías$. En este trabajo se utilizarán indistintamente cualquiera de los dos términos.

Si comparamos con una neurona biológica como la presentada en la figura 2.2 del marco teórico, el peso w corresponde a la fortaleza de la sinapsis, el cuerpo de la neurona o Soma es representado por el sumador y la función de transferencia f junto con la salida a representa la señal en el axón.

Se puede observar en la figura 6.5 que la salida a depende de la función de transferencia elegida.

El $bías$ es muy parecido a un peso, excepto que este tiene una constante 1 como entrada. Sin embargo si no se desea tener $bías$ en una neurona en particular, este puede ser omitido. Como se verá más adelante.

El $peso\ w$ y el $bías\ b$ ambos son parámetros ajustables de la neurona. Típicamente la función de transferencia es elegida por el diseñador y entonces los parámetros w y b serán ajustados por alguna regla de aprendizaje para que la relación entrada - salida de la neurona cumpla con cierto objetivo. (Hagan M., 2015).

6.2.1.3 Funciones de Transferencia

La función de transferencia de la figura 6.5 puede ser una función lineal o no lineal de n. Una función de transferencia particular es elegida para satisfacer alguna especificación del problema que la neurona resolverá.

En la *función de transferencia* $hard\ lim$ mostrada a la izquierda de la figura 6.6, la salida de la neurona será 0 si el argumento de la función es menor a 0 o igual a 1 si el argumento es mayor o igual a cero. Esta función es útil para clasificar las entradas en dos categorías discretas.

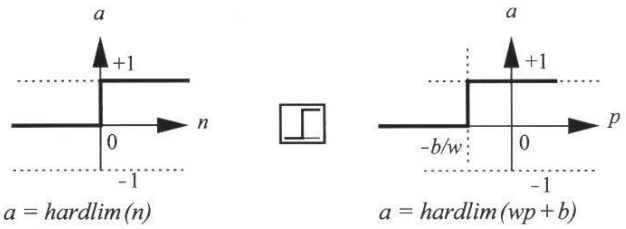


Figura 6.6 Función de transferencia *hardlim*

((Hagan M., 2015) Pág. 2-4)

En la parte derecha de la figura 6.6 se puede ver el efecto del peso y del bías.

La figura 6.7 muestra una función de *transferencia lineal* en la cual la salida es igual a su entrada.

$$a = n \tag{6.1}$$

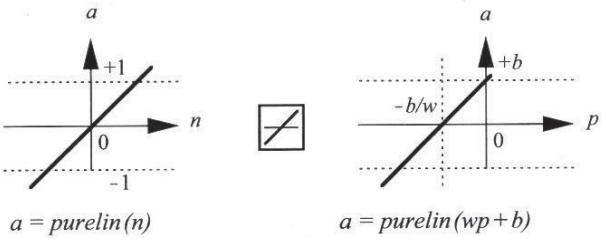


Figura 6.7 Función de transferencia *lineal*

((Hagan M., 2015) Pág. 2-4)

Este tipo de función de transferencia se utiliza en redes ADALINE.

La salida (a) versus (p) se muestra a la derecha en la figura 6.7.

La función de transferencia $log - sigmoid$ se muestra en la figura 6.8, esta función de transferencia toma la entrada (que tiene valores entre $+\infty\ y\ -\infty$) y los limita a un rango entre $(0,1)$.de acuerdo a la expresión:

$$a = \frac{1}{1+e^{-n}} \tag{6.2}$$

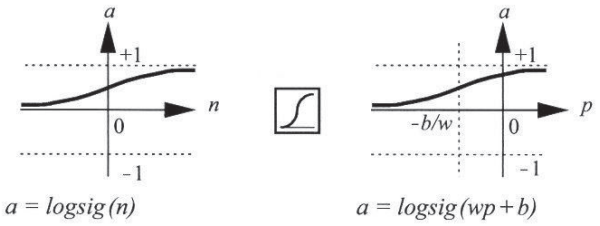


Figura 6.8 Función de transferencia *log-sigmoid*

((Hagan M., 2015) Pág. 2-5)

Esta función de transferencia $log - sigmoid$ es comúnmente utilizada en redes multicapa que son entrenadas con el algoritmo de retropropagación, en parte porque esta función es diferenciable. (Hagan M., 2015).

La tabla 6.1 muestra un listado de las funciones de transferencia presentes en el Programa MatLab. ver 15.

Tabla 6.1 Funciones de transferencia en Mat Lab

Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons		compet

((Hagan M., 2015) Pág. 2-6)

6.2.1.4 Neurona con Múltiples Entradas

Normalmente una neurona tiene más de una entrada. Una neurona con R entradas se muestra en la figura 6.9. Las entradas individuales $p_1, p_2, \dots, p_R$ con sus correspondientes pesos $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ de la matriz **W**.

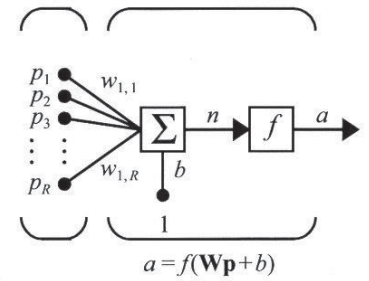


Figura 6.9 Neurona con múltiples entradas

((Hagan M., 2015) Pág. 2-7)

La neurona tiene un bías b, el cual es sumado a las entradas ponderadas para dar como resultado la salida n:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \cdots + w_{1,R}p_R \quad (6.3)$$

Esta expresión se puede escribir en forma matricial:

$$n = \mathbf{W}\mathbf{p} + b \quad (6.4)$$

Donde la matriz **W** para una sola neurona tiene una sola fila.

Entonces la salida de la neurona se puede escribir como:

$$a = f(\mathbf{W}\mathbf{p} + b) \quad (6.5)$$

Afortunadamente, las redes neuronales a menudo pueden se descritas con matrices. En el apéndice B se hace una revisión de los conceptos básicos del algebra matricial, que se utilizarán en el presente trabajo.

Se adoptará una convención particular para asignar los índices de los elementos de la matriz de pesos **W**. El primer índice nos indica la neurona de destino para ese peso. El segundo índice indica el origen de la señal que alimenta a la neurona. Por ejemplo los índices $w_{1,2}$ nos dicen que este peso representa la conexión a la primera neurona desde la segunda fuente de entrada. (Hagan M., 2015).

Se utilizará una notación abreviada para evitar la complejidad de redes con varias neuronas. En la figura 6.10 se muestra esta notación abreviada:

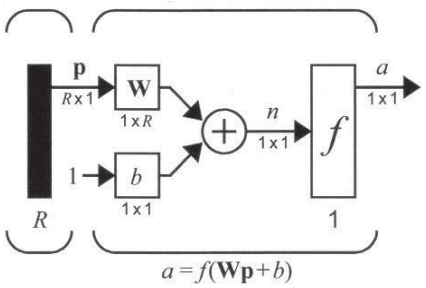


Figura 6.10 Notación abreviada de una neurona con R entradas

((Hagan M., 2015) Pág. 2-8)

En este caso la salida a de la red es un escalar, si se tuvieran más de una neurona, la salida de la red sería un vector.

Observe que en esta notación abreviada las dimensiones de las variables siempre están incluidas, así se puede interpretar inmediatamente si se trata de un escalar, vector o matriz.

Note que el número de entradas de la red se configuran de acuerdo a una especificación externa del problema. (Hagan M., 2015).

6.2.2 ARQUITECTURA DE LA RED

Normalmente una neurona, aunque con muchas entradas, puede no ser suficiente. Se necesitan cinco o diez, operando en paralelo, que se denominará una “capa”.

6.2.2.1 Capa de Neuronas

Una sola capa con S neuronas se muestra en la figura 6.11. Se puede notar que cada una de las R entradas está conectada a cada una de las neuronas y que la matriz de pesos tiene S filas.

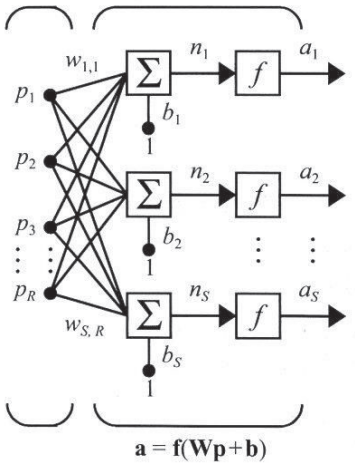


Figura 6.11 Capa con S neuronas

((Hagan M., 2015) Pág. 2-9)

La capa incluye la matriz de pesos, el sumador, el vector bías **b**, las funciones de transferencia y el vector de salida $\boldsymbol{a}$. Algunos autores se refieren a la entrada como otra capa, eso no se hará aquí.

Es común que el número de entradas a una capa sea diferente del número de neuronas. $(R \neq S)$.

Además las funciones de transferencia pueden ser diferentes en cada neurona.

Los elementos del vector de entrada ingresan a la red a través de la matriz de pesos **W**:

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,R} \\ \vdots & \ddots & \vdots \\ w_{S,1} & \cdots & w_{S,R} \end{bmatrix}$$

Como se indicó anteriormente, los índices de las filas de la matriz **W** indican la neurona de destino asociada a ese peso, mientras que los índices de las columnas indican el origen de la entrada para ese peso. Así $w_{3,2}$ nos dice que este peso representa la conexión a la tercera neurona desde la segunda fuente de entrada.

Afortunadamente una capa de S neuronas con R entradas se puede graficar en notación abreviada, como se muestra en la figura 6.12.

Los símbolos de las variables nos dicen que: $\boldsymbol{p}$ es vector de dimensión R, **W** es una matriz de SxR, **a** y **b** son vectores de dimensión S. (Hagan M., 2015).

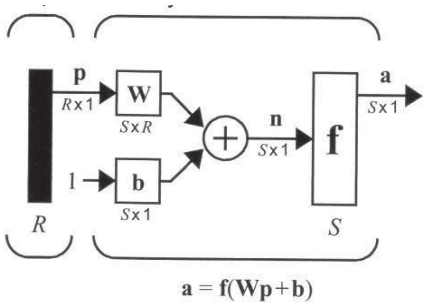


Figura 6.12 Notación Abreviada de una capa con *S* neuronas y *R* entradas

((Hagan M., 2015) Pág. 2-9)

6.2.2.2 Múltiples Capas de Neuronas

Ahora se considerará una red con varias capas. Cada capa tiene su propia matriz de pesos **W**, su propio vector bías **b**, su vector de entrada neto **n** y su vector de salida **a**. Se necesita introducir alguna notación adicional para distinguir entre las diferentes capas. Se utilizarán superíndices para identificar las capas. Se aumentará el número de la capa como *superíndice* al nombre de cada una de las variables. Así, la matriz de pesos para la primera capa se escribe como $\mathbf{W^1}$ y la matriz de pesos para la segunda capa $\mathbf{W^2}$**.** Esta notación se utiliza en la red de tres capas de la figura 6.13.

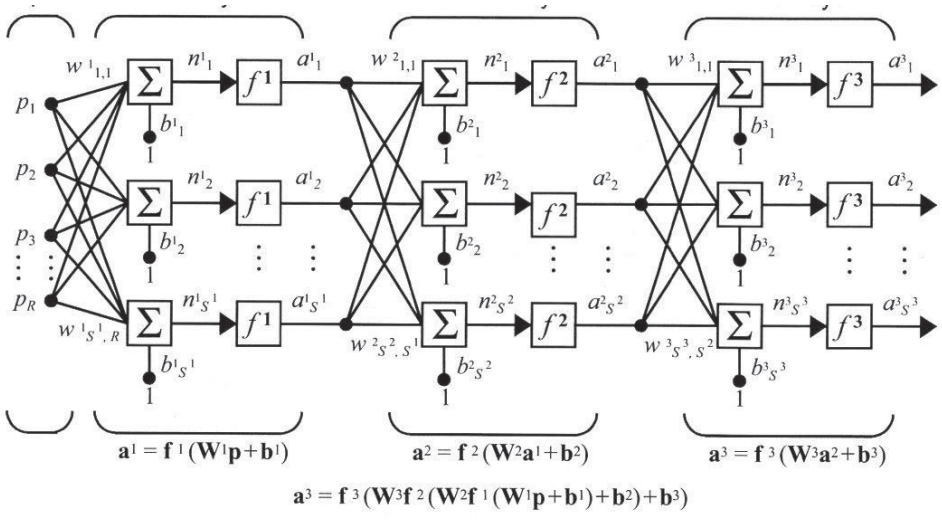


Figura 6.13 Red de tres capas

((Hagan M., 2015) Pág. 2-11)

Como se puede observar, hay R entradas, S^1 neuronas en la primera capa, S^2 neuronas en la segunda capa, etc. Como se puede notar diferentes capas pueden tener diferente número de neuronas.

Las salidas de las capas uno y dos son las entradas de las capas dos y tres. Así la capa 2 puede verse como una red de una capa con $R = S^1$ entradas, $S = S^2$ neuronas y una matriz de pesos $\mathbf{W^2}$ **de** $(S^2 x S^1)$. La entrada de la capa 2 es $\mathbf{a^1}$ y la salida es $\mathbf{a^2}$**.**

A la capa en la cual está la salida de la red se la llama *capa de salida*. Las otras capas se denominan *capas ocultas*. La red de la figura 6.13 tiene una capa de salida (última capa) y dos capas ocultas (las dos primeras capas).

Esta misma red de tres capas mostrada en la figura 6.13 se la puede presentar mediante la notación abreviada, como se muestra en la figura 6.14.

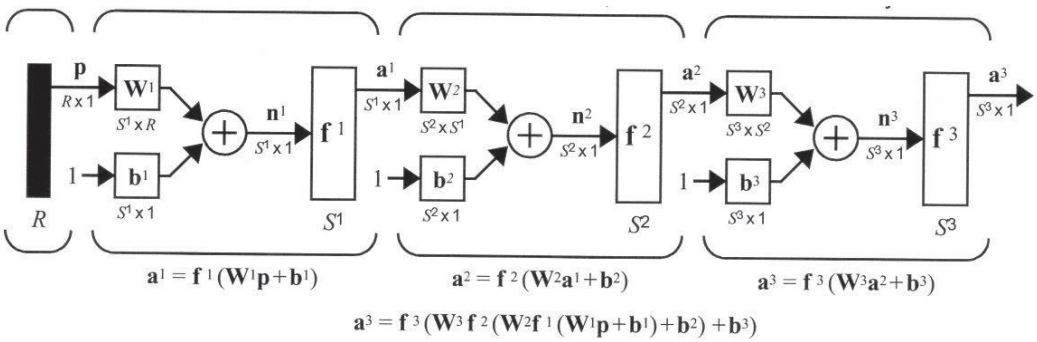


Figura 6.14 Red de tres capas, Notación Abreviada

((Hagan M., 2015) Pág. 2-12)

Las redes multicapa son más poderosas que las redes de una sola capa. Por ejemplo una red de dos capas con una función sigmoidea en la primera capa y una función lineal en la segunda capa puede ser entrenada para aproximar a la mayoría de funciones arbitrariamente bien. Redes de una sola capa no pueden hacer esto.

Para especificar una red neuronal se debe tomar en cuenta lo siguiente: Primero, el número de entradas de la red y el número de salidas de la red son definidas por especificaciones externas del problema a resolver. Acorde con esto se elige el número de entradas a la red y el número de neuronas de salida en la capa de salida. Finalmente, las características deseadas de la señal de salida ayudan a elegir la

función de transferencia de la capa de salida. Si una salida debe ser -1 o 1, entonces una función paso (*Hard Limit)* debería ser utilizada. Si la red tiene una sola capa es casi totalmente definida por las especificaciones del problema, incluyendo el número específico de entradas, salidas y las características particulares de la señal de salida.

Pero si la red tiene más de dos capas, las especificaciones externas del problema no nos dicen directamente el número de neuronas requeridas en las capas escondidas. Este problema se constituye actualmente en una amplia área de investigación. Algunas pistas se verán más adelante cuando veamos el algoritmo de Retropropagación (Backpropagation).

Como una regla práctica las redes neuronales no tienen más de dos o tres capas, ya que cuatro o más capas es muy raro.

Respecto al bías, se puede elegir una red con o sin bías. El bías da a la red una variable extra, por lo tanto una red con bías es más poderosa que una red sin bías. Además el bías nos ayuda a que no se anule la entrada a la red cuando la entrada neta *n* es cero. (Hagan M., 2015).

6.2.2.3 Redes Recurrentes

Una *red recurrente* como la de la figura 6.14, es una red con realimentación; alguna de sus salidas está conectada a su entrada. Estas redes son diferentes a las estudiadas anteriormente ya que estas eran unidireccionales sin conexiones de realimentación.

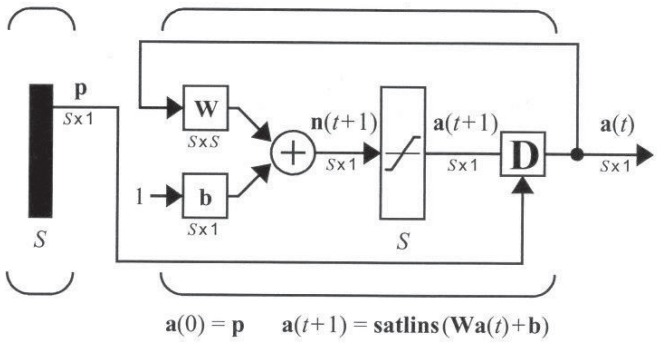


Figura 6.15 Red Recurrente

((Hagan M., 2015) Pág. 2-13)

En la figura 6.15 se puede observar un nuevo elemento **D** que es un bloque de retardo.

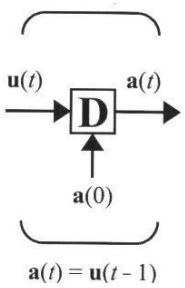


Figura 6.16 Bloque de Retardo

((Hagan M., 2015) Pág. 2-13)

La salida de este bloque es la entrada retrasada un intervalo de tiempo.

$$\mathbf{a}(t) = \mathbf{u}(t-1) \qquad (6.6)$$

La Ec. 6.6 requiere que la salida sea inicializada a $t = 0$. Esta condición inicial se indica en la figura 6.16 con la flecha en la parte de abajo del bloque.

Las redes recurrentes son potencialmente más poderosas que las redes unidireccionales y pueden exhibir un comportamiento temporal. (Hagan M., 2015).

6.2.2.4 Reglas de Aprendizaje

Una regla de aprendizaje es un procedimiento para modificar los pesos y bías de la red (dicho procedimiento también es llamado como algoritmo de entrenamiento). El propósito de la regla de aprendizaje es entrenar la red para realizar alguna tarea. Existen muchos tipos de reglas de aprendizaje, se las puede clasificar dentro de tres grandes grupos: Aprendizaje Supervisado, Aprendizaje No Supervisado y Aprendizaje Reforzado.

En el ***aprendizaje supervisado*** la regla de aprendizaje es provista de varios ejemplos (conjunto de entrenamiento) del comportamiento correcto de la red, junto con la salida deseada u objetivo, e iterativamente ésta ajusta sus pesos y bías hasta que su salida tienda a ser la deseada, utilizando para ello información detallada del error que comete en cada paso. (Del Brío, 2007), (Hagan M., 2015).

En el ***aprendizaje no supervisado*** los pesos y bías son modificados en respuesta solamente a las entradas de la red. No hay salida deseada u objetivo disponible.

El ***aprendizaje reforzado*** es similar al supervisado, excepto que, en lugar de proveer la salida deseada para cada entrada, el algoritmo solamente da un grado. Por ejemplo lo bien o mal que está actuando, pero si dar más detalles. En ocasiones se denomina aprendizaje por *premio-castigo*.

En el proceso de entrenamiento es importante diferenciar entre el nivel de error durante la fase de aprendizaje con un conjunto de datos para entrenamiento y el error que la red ya entrenada comete ante datos no utilizados en el aprendizaje, lo cual mide la capacidad de **generalización** de la red. Interesa más una buena generalización que un error muy pequeño durante el entrenamiento, pues ella indicará que la red ha captado correctamente la relación subyacente de los datos. (Del Brío, 2007).

6.2.3 PERCEPTRÓN MULTICAPA MLP

Si se añaden capas intermedias (ocultas) a una red de una sola capa (perceptrón simple) se obtiene un perceptrón multicapa. Este tipo de arquitectura suele entrenarse mediante el algoritmo de retropropagación de errores BP (Backpropagation) o cualquiera de sus variantes. (Del Brío, 2007).

Aparentemente la descripción de un primer algoritmo para entrenar redes multicapa fue contenida en la tesis doctoral de Paul Werbos (1974). Esta tesis presentaba un algoritmo en el contexto de redes en general, con las redes neuronales como un caso especial y no fue difundido en la comunidad de las redes neuronales. No fue hasta mediados de los 80s que el algoritmo de retropropagación fue redescubierto y ampliamente publicitado. Este fue redescubierto independientemente por David Rumelhart, Geoffrey Hinton y Ronald Williams, David Parker y Yann Le Cun. El algoritmo fue popularizado por su inclusión en el libro Paralell Distributed Processing (Rumelhart, McClelland, & Group, 1987).

El perceptrón multicapa MLP, entrenado por el algoritmo de retropropagación, es actualmente la red neuronal más utilizada. (Hagan M., 2015).

Existen numerosas variantes, como incluir neuronas no lineales en la capa de salida, introducir más capas ocultas, emplear otras funciones de activación, restringir el número de conexiones entre una neurona y las de la capa siguiente y

finalmente la que se utilizará en el presente trabajo para los pronósticos, introducir dependencias temporales o arquitecturas recurrentes. (Del Brío, 2007).

Por comodidad se va a reproducir la figura 6.13 perteneciente a un perceptrón de tres capas en la figura 6.17. Se puede considerar que la figura 6.17 está conformada por tres perceptrones simples en cascada. La salida de la primera red es la entrada a la segunda red y la salida de la segunda red es la entrada de la tercera red. Cada capa podría tener un diferente número de neuronas e incluso diferentes funciones de transferencia. Se debe recordar que se utiliza superíndices para identificar el número de capa. Así la matriz de pesos de la primera capa se representa por $\mathbf{W^1}$ y la matriz de pesos para la segunda capa como $\mathbf{W^2}$**.** (Hagan M., 2015).

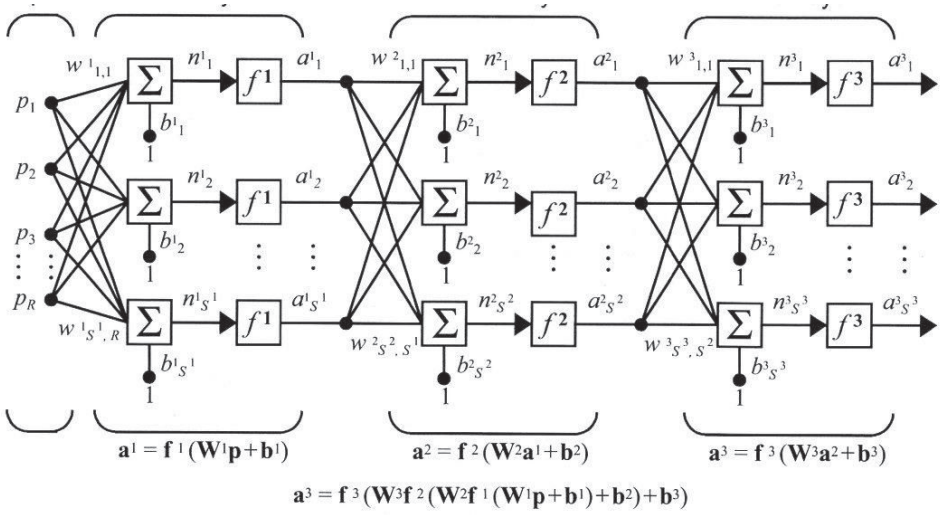


Figura 6.17 Perceptrón de tres capas

((Hagan M., 2015) Pág. 11-3)

A continuación se analizan dos de las capacidades de esta red perceptrón multicapa MLP, Primero la capacidad de esta red para clasificación de patrones en respuesta a la crítica de Minsky y Papert allá por los años 60s, y luego la capacidad como aproximador de funciones.

6.2.3.1 Clasificación de Patrones

Se considerará el clásico problema del O exclusivo (XOR). Los pares Entrada/Objetivo para una compuerta XOR son:

$\{\mathbf{p_1} = \binom{0}{0} | t_1 = 0\}\{\mathbf{p_2} = \binom{0}{1} | t_2 = 1\}\{\mathbf{p_3} = \binom{1}{0} | t_3 = 1\}\{\mathbf{p_4} = \binom{1}{1} | t_4 = 0\}$.

Este problema, el cual es ilustrado gráficamente en la figura 6.18 fue utilizado por Minsky y Papert en 1969 para demostrar las limitaciones del perceptrón simple. Ya que las dos categorías no son linealmente separables, un perceptrón simple no puede ejecutar esta clasificación.

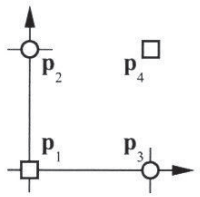


Figura 6.18 Problema de O Exclusivo (XOR)

((Hagan M., 2015) Pág. 11-3)

Una red de dos capas puede resolver este problema del XOR. Una de las soluciones es utilizar dos neuronas en la primera capa para crear dos regiones de decisión. La primera región separa $\mathbf{p_1}$ de los otros patrones y la segunda región separa $\mathbf{p_4}$. Luego la segunda capa es utilizada para combinar las dos regiones de decisión juntas utilizando la operación Y (AND). Las secciones de decisión se muestran en la figura 6.19.

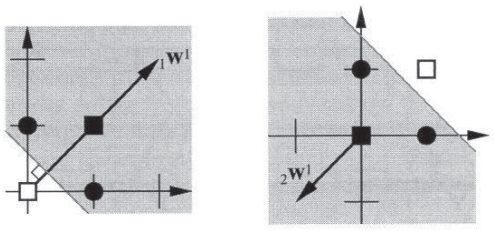


Figura 6.19 Secciones de Decisión Red XOR

((Hagan M., 2015) Pág. 11-4)

La región de decisión para esta red de dos capas 2-2-1 (figura 6.21) se muestra en la figura 6.20.

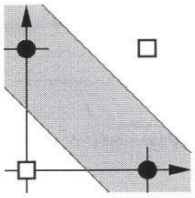


Figura 6.20 Región de Decisión Red Neuronal 2-2-1

((Hagan M., 2015) Pág. 11-4)

La región sombreada de la figura 6.20 indica las entradas de la red que producen una salida igual a 1.

La red neuronal de dos capas 2-2-1, se muestra en la figura 6.21.

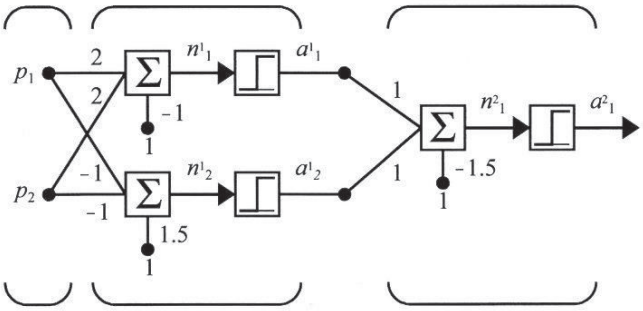


Figura 6.21 Red Neuronal 2-2-1 para resolver el problema del XOR

((Hagan M., 2015) Pág. 11-4)

6.2.3.2 Aproximador de Funciones

Es muy ilustrativo ver las redes neuronales como aproximadores de funciones. En sistemas de control por ejemplo, el objetivo es encontrar una función de realimentación que relacione la salida medida con las entradas de control. En un filtro adaptivo el objetivo es encontrar una función que relacione los valores retrasados de la señal de entrada a una señal de salida aproximada. Para ilustrar la flexibilidad del MLP para implementar funciones se analizará el siguiente ejemplo. Se considerará una red de dos capas, 1-2-1 como se muestra el figura 6.22. Para este ejemplo la función de transferencia de la primera capa es Sigmoidea (log-sigmoid) y la función de transferencia para la segunda capa es lineal.

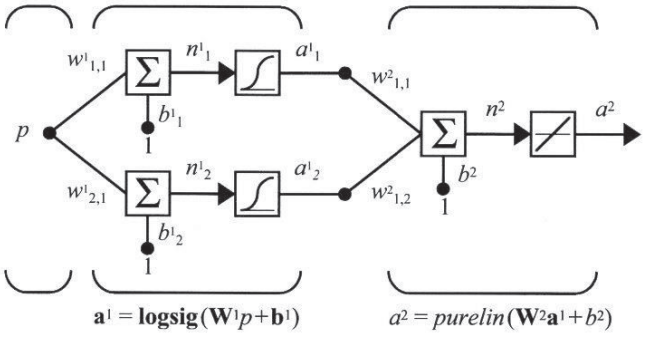


Figura 6.22 Red Neuronal 1-2-1 Aproximador de Funciones

((Hagan M., 2015) Pág. 11-5)

Es decir,

$$f^1(n) = \frac{1}{1+e^{-n}} \text{ y } f^2(n) = n \qquad (6.7)$$

Se supone que los valores nominales de los pesos y bías para esta red son:

$w_{1,1}^1 = 10$, $w_{2,1}^1 = 10$, $b_1^1 = -10$, $b_2^1 = 10$,

$w_{1,1}^2 = 1$, $w_{1,2}^2 = 1$, $b^2 = 0$.

La respuesta de esta red se muestra en la figura 6.23, la cual grafica la salida de la red a^2 mientras la entrada p varía en un rango entre $[-2,2]$.

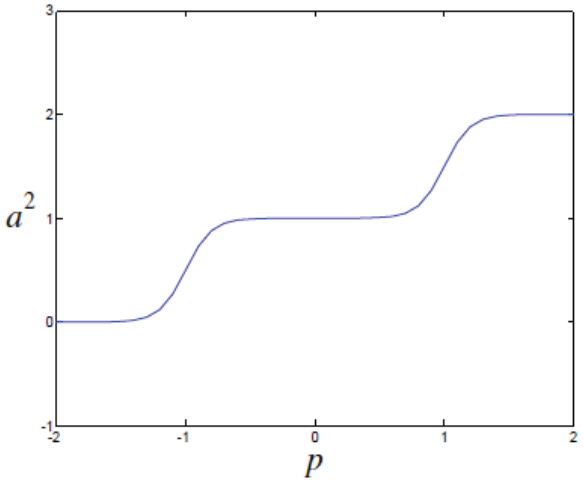


Figura 6.23 Respuesta de la red de la figura 6.22

((Hagan M., 2015) Pág. 11-6)

Se puede notar que la respuesta consiste en dos pasos, producidos por cada neurona con función de transferencia $sigmoidea$ en la primera capa. Ajustando los parámetros de la red se puede cambiar la forma y localización de cada paso.

La figura 6.24 ilustra los efectos del cambio de parámetros en la respuesta de red. La figura 6.24 (a) muestra como el bías en la primera capa (oculta) puede ser utilizado para cambiar la posición de los pasos. La figura 6.24 (b) ilustra como los pesos determinan la pendiente de los pasos. El bías en la segunda capa (capa de salida) mueve la respuesta de la red hacia arriba o hacia abajo, como se puede ver en la figura 6.24 (d).

Mediante este ejemplo se puede notar cuan flexible es la red multicapa. Parecería que se puede usar el perceptrón multicapa para aproximar casi cualquier función, si se tiene el suficiente número de neuronas en la capa oculta.

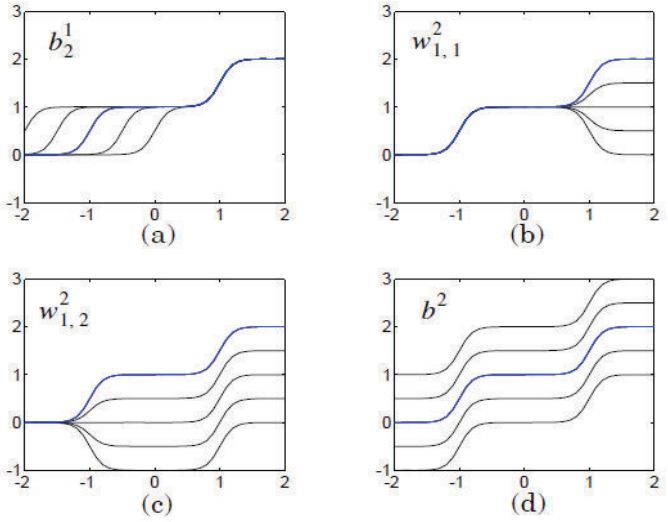


Figura 6.24 Efectos al cambiar los parámetros de la red

((Hagan M., 2015) Pág. 11-7)

Se ha podido ver que una red de dos capas, con funciones de transferencia del tipo $log-sigmoid$ en la capa oculta y una función de transferencia lineal en la capa de salida, pude aproximar cualquier función de interés con cualquier grado de precisión, provisto por el número de neuronas en la capa oculta.

Después de tener una buena idea de lo poderoso que es el perceptrón multicapa para reconocimiento de patrones y aproximación de funciones, el siguiente paso es estudiar un algoritmo para su entrenamiento. (Hagan M., 2015).

6.2.3.3 Algoritmo de Retropropagación

Se simplificará el desarrollo de este algoritmo de retropropagación si se utiliza la notación abreviada del MLP. La notación abreviada para una red de tres capas se muestra en la figura 6.25.

Para redes multicapa la salida de una capa se convierte en entrada de la siguiente capa. La ecuación que describe esta operación es:

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^{m} + \mathbf{b}^{m+1}) \; para \; m = 0,1,\dots,M-1 \qquad (6.8)$$

Donde M es el número de capas de la red. Las neuronas en la primera capa que reciben entradas externas son:

$$\mathbf{a}^{0} = \mathbf{p}\,, \qquad (6.9)$$

Las cuales proveen el punto de inicio de la ecuación 6.8.

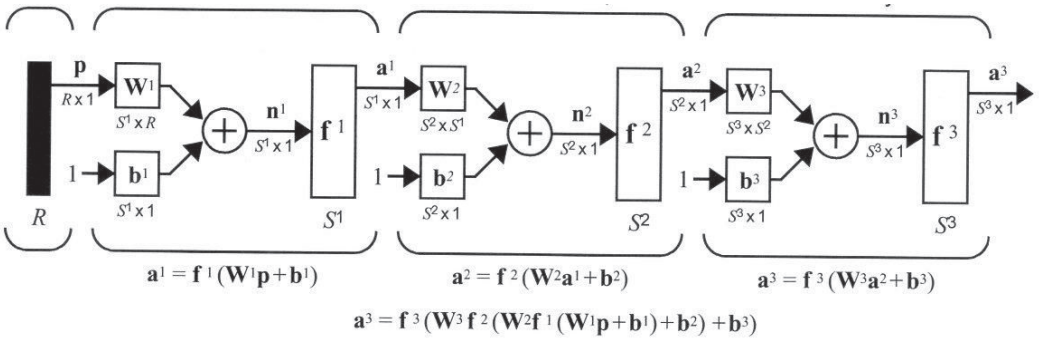


Figura 6.25 Notación Abreviada de una red de tres capas

((Hagan M., 2015) Pág. 11-8)

Las salidas de las neuronas en la última capa son consideradas las salidas de la red:

$$\mathbf{a} = \mathbf{a}^{M} \qquad (6.10)$$

6.2.3.3.1 Índice de Desempeño

Otra clase importante de aprendizaje es el denominado Aprendizaje con Desempeño (Performance Learning), en el cual los parámetros de la red se ajustan para optimizar el desempeño de la red.

Existen dos pasos involucrados en el proceso de optimización. *El primero* es definir qué se entiende por “desempeño”. Es decir encontrar una medida cuantitativa del

desempeño de la red, denominado *índice de desempeño*, el cual será pequeño cuando la red se desempeña bien y grande cuando el desempeño de la red es pobre. (Hagan M., 2015).

El Segundo paso en el proceso de optimización es buscar el espacio de los parámetros (ajuste de pesos y bías) orientado a reducir el índice de desempeño.

El algoritmo de retropropagación para redes multicapa es una generalización del algoritmo de mínimos cuadrados (LMS) y los dos algoritmos utilizan el mismo índice de desempeño: el *error cuadrado medio*. Al algoritmo se le debe proveer de un conjunto de pares entrada – salida deseada:

$$\{\mathbf{p}_1|\mathbf{t}_1\},\{\mathbf{p}_2|\mathbf{t}_2\},\dots,\{\mathbf{p}_Q|\mathbf{t}_Q\}, \tag{6.11}$$

Donde $\mathbf{p}_Q$ es la entrada a la red y $\mathbf{t}_Q$ es la correspondiente salida deseada (objetivo).

Cada vez que una entrada es aplicada a la red, la salida de la red es comparada con la salida deseada. El algoritmo deberá ajustar los parámetros de la red para minimizar el error cuadrático medio:

$$F(\mathbf{x}) = E[e^2] = E[(t-a)^2] \tag{6.12}$$

Donde $\mathbf{x}$ es el vector de pesos y bías. Si la red tiene múltiples salidas se puede generalizar a:

$$F(\mathbf{x}) = E[\mathbf{e}^T\,\mathbf{e}] = E[(\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a})] \tag{6.13}$$

Al igual que en el algoritmo de mínimos cuadrados (apéndice D), se puede aproximar el error cuadrático medio por:

$$\hat{F}(\mathbf{x}) = \big(\mathbf{t}(k)-\mathbf{a}(k)\big)^T\big(\mathbf{t}(k)-\mathbf{a}(k)\big)\big] = [\mathbf{e}^T(k)\,\mathbf{e}(k)] \tag{6.14}$$

Donde el valor esperado del error cuadrático ha sido reemplazado por el error cuadrático a la iteración k.

El algoritmo del descenso por el gradiente (revisar apéndice D) para el error cuadrático medio aproximado es:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \propto \frac{\partial \hat{F}}{\partial w_{i,j}^m}, \tag{6.15}$$

$$b_i^m(k+1) = b_i^m(k) - \propto \frac{\partial \hat{F}}{\partial b_i^m}, \tag{6.16}$$

Donde $\propto$ es la relación de aprendizaje.

La parte difícil de este algoritmo es el cálculo de las derivadas parciales en las ecuaciones (6.15) y (6.16), se verán a continuación.

En una red multicapa el error no es una función explícita de los pesos en la o las capas ocultas, por lo tanto las derivadas no se calculan tan fácilmente.

Ya que el error es una función indirecta de los pesos en la capa oculta, se debe utilizar la *regla de la cadena* para calcular sus derivadas. Así aplicando la regla de la cadena a las derivadas parciales de las ecuaciones (6.15) y (6.16) se tiene:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} x \frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} \tag{6.17}$$

$$\frac{\partial \hat{F}}{\partial b_{i}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} x \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \tag{6.18}$$

El segundo término en cada una de estas ecuaciones puede ser calculado sin problema, ya que la entrada neta a la capa m es una función explícita de los pesos y bías en esta capa:

$$n_{i}^{m} = \sum_{j=1}^{s^{m-1}} w_{i,j}^{m} a_{j}^{m-1} + b_{i}^{m} \tag{6.19}$$

Por lo tanto

$$\frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} = a_{j}^{m-1} \ , \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} = 1 \tag{6.20}$$

Si se define

$$s_{i}^{m} \equiv \frac{\partial \hat{F}}{\partial n_{i}^{m}} \tag{6.21}$$

Como la sensibilidad de $\hat{\mathrm{F}}$ a los cambios en el $i - ésimo$ elemento de la entrada neta a la capa m entonces las ecuaciones (6.17) y (6.18) se pueden simplificar a

$$\frac{\partial \hat{F}}{\partial w_{i,j}^{m}} = s_{i}^{m} \, a_{j}^{m-1} \tag{6.22}$$

$$\frac{\partial \hat{F}}{\partial b_{i}^{m}} = s_{i}^{m} \tag{6.23}$$

Ahora se puede expresar el algoritmo del descenso por el gradiente como

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \propto s_{i}^{m} \, a_{j}^{m-1} \ , \tag{6.24}$$

$$b_{i}^{m}(k+1) = b_{i}^{m}(k) - \propto s_{i}^{m} \ , \tag{6.25}$$

En forma matricial quedaría

$$\boldsymbol{W}^{\mathrm{m}}(k+1) = \mathbf{W}^{\mathrm{m}}(k) - \propto \mathbf{s}^{m} \, (\mathbf{a}^{m-1})^{T} \ , \tag{6.26}$$

$$\mathbf{b}^{m}(k+1) = \mathbf{b}^{m}(k) - \propto \mathbf{s}^{m} \ , \tag{6.27}$$

Donde

$$\mathbf{s}^m \equiv \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix} \tag{6.28}$$

6.2.3.3.2 Retropropagación de las Sensibilidades

Se debe calcular ahora las sensibilidades $\mathbf{s}^m$, que requieren otra aplicación de la regla de la cadena. Es este proceso el que nos da el término *retropropagación*, ya que la sensibilidad de la capa m es calcualda a partir de la sensibilidad de la capa $(m+1)$.

Para obtener la relación recurrente para las sensibilidades, se utilizará la siguiente matriz Jacobiana:

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \cdots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix} \tag{6.29}$$

A continuación se buscará una expresión para esta matriz. Se considerará el i,j elemento de la matriz:

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial (\sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1})}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} \dot{f}^m(n_j^m), \tag{6.30}$$

Donde

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \tag{6.31}$$

Por lo tanto el Jacobiano puede escribirse como

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m), \tag{6.32}$$

Donde

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 \ldots . & 0 \\ 0 & \dot{f}^m(n_2^m) \ldots & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 \ldots . & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \tag{6.33}$$

Ya se puede escribir la relación recurrente para la sensibilidad por medio de la regla de la cadena, en forma matricial:

$$\mathbf{s}^m = \frac{\partial \hat{F}}{\partial \mathbf{n}^m} = \left[\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m}\right]^{\mathrm{T}} \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1}) \frac{\partial \hat{F}}{\partial \mathbf{n}^{m+1}}$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^{\mathrm{T}} \mathbf{s}^{m+1} \tag{6.34}$$

Analizando la ecuación (6.34) se puede entender de donde viene el nombre de retropropagación. Las sensibilidades son propagadas hacia atrás a través de la red desde la última capa hasta la primera capa.

$$\mathbf{s}^M \rightarrow \mathbf{s}^{M-1} \rightarrow \cdots \rightarrow \mathbf{s}^2 \rightarrow \mathbf{s}^1 \tag{6.35}$$

Cabe destacar que el algoritmo de retropropagación utiliza la misma técnica del descenso por gradiente que se utiliza en el algoritmo de mínimos cuadrados. La única complicación es que para calcular el gradiente se necesita primero retropropagar las sensibilidades. (Hagan M., 2015).

Pero se necesita un paso más para completar el algoritmo de retropropagación, las condiciones iniciales ($\mathbf{s}^M$) para la ecuación recurrente (6.34). Estas se obtienen de la capa final:

$$s_i^M = \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial (\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a})}{\partial n_i^M} = \frac{\partial \sum_{j=1}^{s^M} (t_j - a_j)^2}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M} \tag{6.36}$$

Ya que

$$\frac{\partial a_i}{\partial n_i^M} = \frac{\partial a_i^M}{\partial n_i^M} = \frac{\partial f^M(n_i^M)}{\partial n_i^M} = \dot{f}^M(n_i^M) \tag{6.37}$$

Se puede escribir

$$s_i^M = -2(t_i - a_i)\dot{f}^M(n_i^M) \tag{6.38}$$

En forma matricial se tiene

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \tag{6.39}$$

Resumiendo

Primer Paso *propagar la entrada hacia adelante a través de la red:*

$$\mathbf{a}^0 = \mathbf{p}\,, \tag{6.40}$$

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1})\ para\ m = 0,1, \ldots, M-1, \tag{6.41}$$

$$\mathbf{a} = \mathbf{a}^M. \tag{6.42}$$

Segundo Paso *propagar las sensibilidades hacia atrás a través de la red:*

$$\mathbf{s}^{M} = -2\dot{\mathbf{F}}^{M}(\mathbf{n}^{M})(\mathbf{t}-\mathbf{a}) \quad (6.43)$$

$$\mathbf{s}^{m} = \dot{\mathbf{F}}^{m}(\mathbf{n}^{m})(\mathbf{W}^{m+1})^{\mathrm{T}}\mathbf{s}^{m+1}, \text{ para } m = M-1, \dots, 2,1 \quad (6.44)$$

Tercer Paso *actualizar pesos y bías mediante la regla del descenso del gradiente:*

$$\boldsymbol{W}^{\mathrm{m}}(k+1) = \mathbf{W}^{\mathrm{m}}(k) - \propto \mathbf{s}^{m}\,(\mathbf{a}^{m-1})^{T}, \quad (6.45)$$

$$\mathbf{b}^{m}(k+1) = \mathbf{b}^{m}(k) - \propto \mathbf{s}^{m}, \quad (6.46)$$

6.2.3.3.3 Entrenamiento Incremental versus Por Lotes (Batch)

EL algoritmo descrito anteriormente se denomina formalmente algoritmo de descenso por el gradiente estocástico, que implica entrenamiento incremental o "en línea", en el cual los pesos y bías de la red son actualizados después de cada entrada es presentada. Es también posible ejecutar un *entrenamiento por lotes*, en el cual el gradiente completo es calculado (después de que todas las entradas han sido aplicadas a la red) antes de que pesos y bías sean actualizados. Por ejemplo, si cada entrada ocurre con igual probabilidad, el índice de desempeño (error cuadrático medio) puede ser escrito

$$F(\mathbf{x}) = E[\mathbf{e}^{T}\,\mathbf{e}] = E[(\mathbf{t}-\mathbf{a})^{T}(\mathbf{t}-\mathbf{a})] = \frac{1}{Q}\sum_{q=1}^{Q}\left(\mathbf{t}_q-\mathbf{a}_q\right)^{T}\left(\mathbf{t}_q-\mathbf{a}_q\right). \quad (6.47)$$

EL gradiente total de este índice de desempeño es

$$\nabla F(\mathbf{x}) = \nabla\left\{\frac{1}{Q}\sum_{q=1}^{Q}\left(\mathbf{t}_q-\mathbf{a}_q\right)^{T}\left(\mathbf{t}_q-\mathbf{a}_q\right)\right\} = \frac{1}{Q}\sum_{q=1}^{Q}\nabla\left\{\left(\mathbf{t}_q-\mathbf{a}_q\right)^{T}\left(\mathbf{t}_q-\mathbf{a}_q\right)\right\}. \quad (6.48)$$

Por lo tanto, el gradiente total del error cuadrado medio es la media de los gradientes individuales de los errores al cuadrado. Entonces, para implementar la versión por lotes del algoritmo de retropropagación, se ejecutarían las ecuaciones (6.40) hasta (6.44) para todas las entradas de los datos de entrenamiento. Entonces, los gradientes individuales deberían ser promediados para obtener el gradiente total. Las ecuaciones iterativas para el algoritmo del descenso por el gradiente por lotes serían entonces

$$\boldsymbol{W}^{\mathrm{m}}(k+1) = \mathbf{W}^{\mathrm{m}}(k) - \frac{\propto}{Q}\sum_{q=1}^{Q}\mathbf{s}_q^{m}\,(\mathbf{a}_q^{m-1})^{T}, \quad (6.49)$$

$$\mathbf{b}^{m}(k+1) = \mathbf{b}^{m}(k) - \frac{\propto}{Q}\sum_{q=1}^{Q}\mathbf{s}_q^{m}. \quad (6.50)$$

(Hagan M., 2015).

6.2.3.4 Utilización del Algoritmo de Retropropagación

En esta sección se presentarán algunos aspectos relacionados con la implementación práctica del algoritmo de retropropagación. Se discutirá la elección de la arquitectura de la red, problemas de convergencia y generalización.

6.2.3.4.1 Elección de la Arquitectura de la Red

Se ha enfatizado que las redes multicapa pueden aproximar casi cualquier función, si se tiene el suficiente número de neuronas en la capa oculta. Sin embargo no se puede asegurar, en general, cuantas capas o cuantas neuronas son necesarias para un desempeño correcto. A continuación se muestra un ejemplo que da una pista acerca de este problema.

En el primer ejemplo se desea aproximar la siguiente función

$$g(p) = 1 + seno\left(\frac{i\pi}{4}p\right) \text{ para } -2 \leq p \leq 2 \qquad (6.51)$$

Donde i puede tomar valores de $1,2,4\ y\ 8$. Cuando i se incrementa la función se vuelve más compleja, ya que se tienen más períodos de la onda seno en el intervalo $-2 \leq p \leq 2$. Es decir se vuelve más difícil para una red neuronal con un número fijo de neuronas en la capa oculta aproximar $g(p)$ cuando i se incrementa.

Para ese ejemplo se utilizará una red tipo 1-3-1, con una función de transferencia en la primera capa $log-sigmoid$ y la función de transferencia para la segunda capa es $lineal$. Se debe recordar que este tipo de red de dos capas produce una respuesta que es la suma de tres funciones $log-sigmoid$ (o algunas $log-sigmoid$ como neuronas haya en la capa oculta). En la figura 6.26 se puede ver claramente que hay un límite en la complejidad de la función que esta red puede manejar.

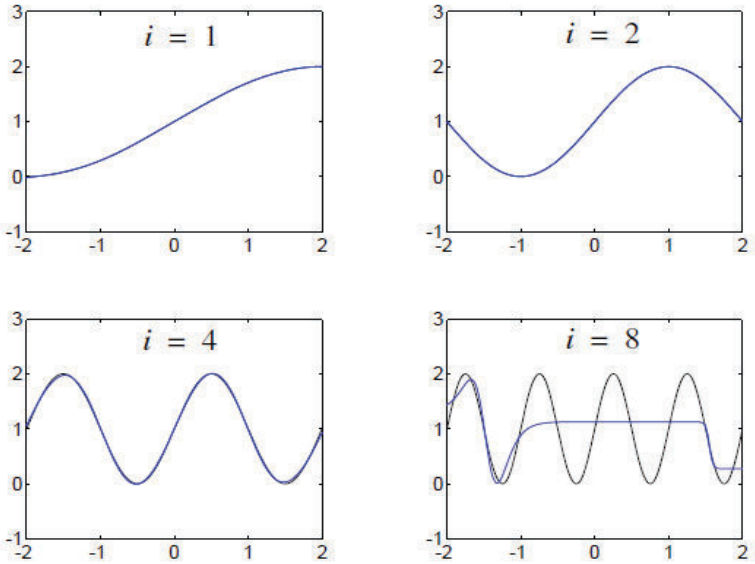


Figura 6.26 Red 1-3-1 Aproximando a la función Seno

((Hagan M., 2015) Pág. 11-19)

Se muestra que para $i = 4$ la red alcanza su máxima capacidad, cuando $i > 4$ la red ya no es capaz de aproximar con precisión la función $g(p)$.

En la parte inferior derecha de la figura 6.26 se puede ver como la red 1-3-1 intenta aproximar $g(p)$ para $i = 8$. El error cuadrático medio es minimizado entre la respuesta de la red y la función, pero la respuesta de la red solamente puede aproximar una parte de la función.

En el siguiente ejemplo abordaremos el problema desde una perspectiva diferente. Esta vez seleccionaremos una función $g(p)$ fija y se irá cambiando la estructura de la red hasta alcanzar una precisión adecuada.

Para $g(p)$ se usará

$$g(p) = 1 + seno\left(\frac{6\pi}{4}p\right) para - 2 \leq p \leq 2. \qquad (6.52)$$

Para aproximar esta función se utilizará una red de dos capas, con una función de transferencia $sigmoidal$ en la primera capa una función de transferencia $lineal$ para la segunda capa.

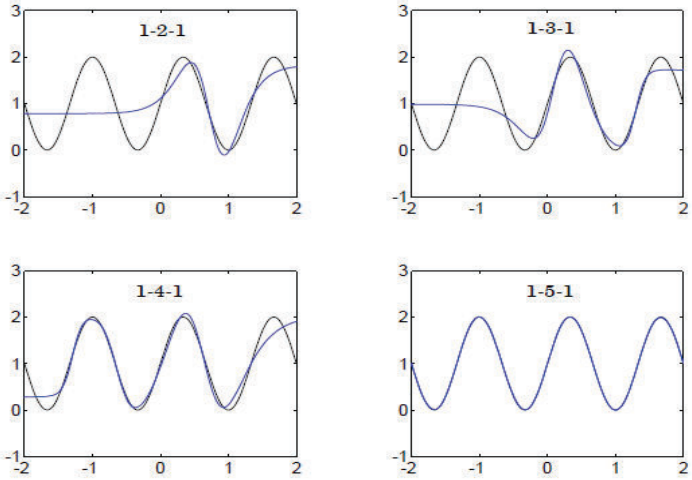


Figura 6.27 Efecto al incrementar el número de neuronas en la capa oculta

((Hagan M., 2015) Pág. 11-20)

La figura 6.27 muestra la respuesta de la red al incrementar el número de neuronas en la capa oculta. A partir de cinco neuronas en la capa oculta la red comienza a aproximar adecuadamente la función $g(p)$.

Resumiendo los resultados, si se desea aproximar una función con muchos puntos de inflexión se necesita un mayor número de neuronas en la capa oculta. (Hagan M., 2015).

6.2.3.4.2 Convergencia

En la sección anterior se presentaron ejemplos en los cuales la respuesta de la red no alcanza una aproximación precisa a la función deseada, aunque el algoritmo de retropropagación produce parámetros de la red que minimizan el error cuadrático medio. Esto ocurre ya que la capacidad de la red se ve limitada por el número de neuronas en la capa oculta. En esta sección se dará un ejemplo en el cual la red es capaz de aproximarse a una función, pero el algoritmo de aprendizaje no es capaz de producir los parámetros de la red que alcancen una aproximación precisa. (Hagan M., 2015).

La función que se necesita que la red aproxime es la siguiente:

$$g(p) = 1 + seno(\pi p) \quad para \ -2 \leq p \leq 2 \tag{6.53}$$

Para aproximar esta función se utilizará una red 1-3-1, con una función $sigmoidal$ en la primera capa y una función de transferencia $lineal$ en la segunda capa.

La figura 6.28 ilustra cuando el algoritmo de aprendizaje converge a la solución que minimiza el error cuadrático medio. La línea gruesa (marcada con 5) representa la respuesta de la red después de varias iteraciones.

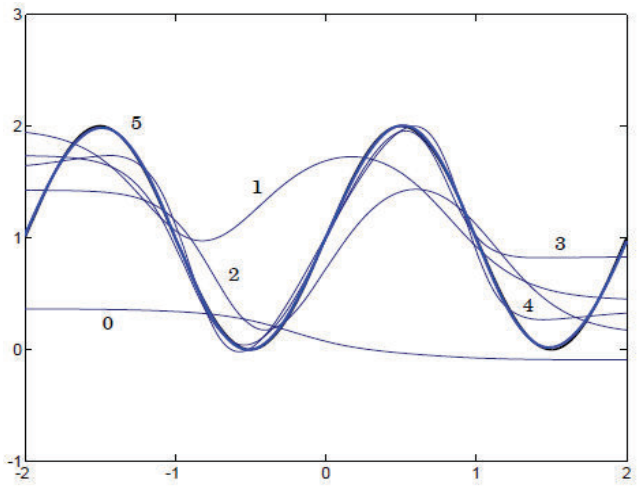


Figura 6.28 Convergencia hacia un Mínimo Global

((Hagan M., 2015) Pág. 11-21)

En la figura 6.29 se ilustra el caso donde el algoritmo converge a una solución que no minimiza el error cuadrático medio. La línea gruesa (marcada con 5) representa la respuesta de la red después de varias iteraciones. El gradiente del error cuadrático medio es cero en la iteración final, por lo tanto se tiene un mínimo local, ya se conoce que existe una mejor solución como se evidencia en la figura 6.28. La única diferencia entre este resultado y el que se muestra de la figura 6.29 es la condición inicial. Es decir desde una condición inicial el algoritmo converge a un *mínimo global*, mientras desde otra condición inicial diferente el algoritmo converge *mínimo local*. (Hagan M., 2015).

Esto no ocurre en redes de una sola capa ya que el error cuadrático medio es una función cuadrática en la cual existe un solo punto mínimo (bajo cualquier condición inicial). Por esta razón el algoritmo de mínimos cuadrados garantiza la convergencia hacia un mínimo global mientras la taza de aprendizaje sea lo suficientemente pequeña.

El error cuadrático medio en redes multicapa es mucho más complejo y tiene varios mínimos locales (como se verá más adelante). Esto significa que cuando el algoritmo de retropropagación converge no se puede estar seguro de que alcanzó

una solución óptima. Es mejor intentar con diferentes condiciones iniciales para asegurarse de que una solución óptima ha sido obtenida.

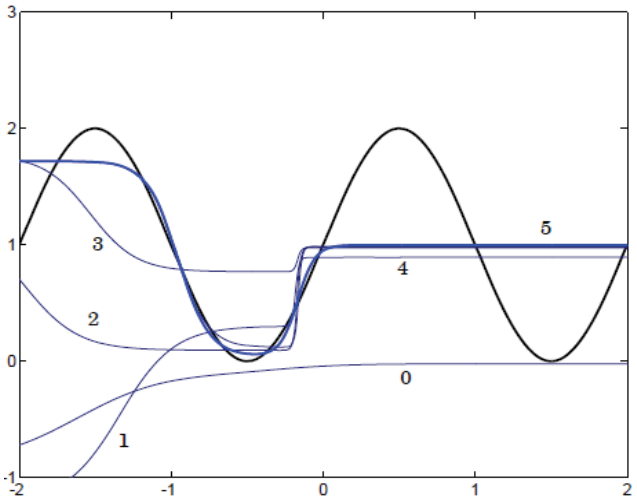


Figura 6.29 Convergencia hacia un Mínimo Local

((Hagan M., 2015) Pág. 11-22)

6.2.3.4.3 Generalización

Normalmente las redes multicapa son entrenadas con un número finito de ejemplos de su comportamiento correcto:

$$\{\mathbf{p}_1|\mathbf{t}_1\},\{\mathbf{p}_2|\mathbf{t}_2\},\ldots,\{\mathbf{p}_Q|\mathbf{t}_Q\}, \tag{6.54}$$

Estos datos de entrenamiento generalmente representan una amplia muestra de los posibles pares entrada/salida. Es muy importante que la red *generalice* lo que ha aprendido a toda la población.

Por ejemplo se va a suponer que los datos de entrenamiento se obtienen por muestreo de la siguiente función:

$$g(p) = 1 + seno\left(\frac{\pi}{4}p\right) \tag{6.55}$$

En los puntos $p = -2, -1.6, -1.2, \ldots, 1.6, 2.$ (existen 11 pares entrada/objetivo).

En la figura 6.30 se puede observar la respuesta de una red 1-2-1 que ha sido entrenada con estos datos.

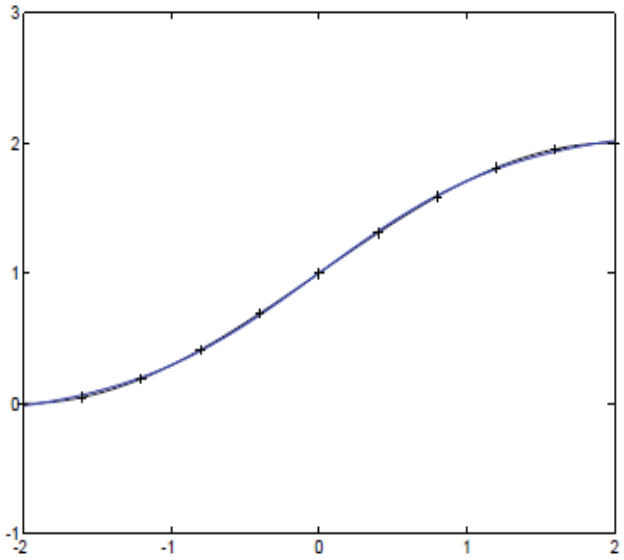


Figura 6.30 Aproximación a la función $g(p)$ con una Red 1-2-1.

((Hagan M., 2015) Pág. 11-23)

Se puede notar que la respuesta de la red es una representación precisa de $g(p)$. Si se desea encontrar la respuesta de la red a un valor de p que no está contenido en los datos de entrenamiento (ejemplo: $p = -0.2$), la red todavía produce una salida cercana a $g(p)$. Entonces se dice que esta red $generaliza$ bien.

En la figura 6.31 se muestra la respuesta de una red 1-9-1 entrenada con los mismos datos anteriores. Se puede ver que la red responde muy bien en los puntos de entrenamiento. Sin embargo si se observa la respuesta para un valor de *p* fuera de los datos de entrenamiento (ejemplo: $p = -0.2$), la red produce una salida alejada de la verdadera respuesta de $g(p)$. Entonces se dice que esta red no $generaliza$ bien.

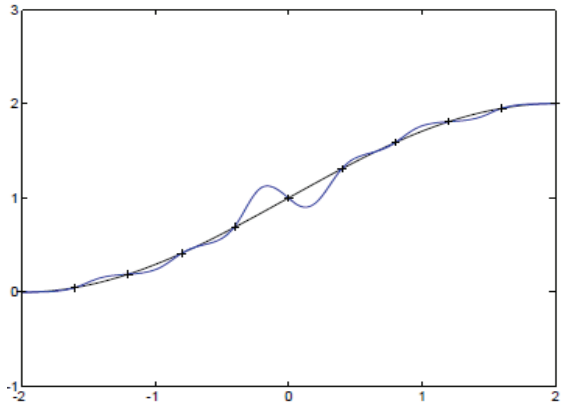


Figura 6.31 Aproximación a la función $g(p)$ con una Red 1-9-1

((Hagan M., 2015) Pág. 11-23)

La red 1-9-1 tiene un exceso de flexibilidad para este problema; esta red tiene un total de 28 parámetros ajustables (18 pesos y 10 bías) y solamente hay 11 puntos para el entrenamiento. La red 1-2-1 tiene solo 7 parámetros (4 pesos y 3 bías) por lo tanto es mucho más restringida en el tipo de funciones que puede manejar.

Para que una red pueda generalizar bien, esta debe tener menos parámetros que datos para el entrenamiento. En una red neuronal, como en todos los problemas de modelamiento, se debe usar la red más simple que represente adecuadamente los datos de entrenamiento. No se debe usar una red más grande cuando una más pequeña trabaja bien. (Este concepto se denomina la regla de Occam).

Una alternativa a usar la red más simple es parar el entrenamiento antes de que la red sobreajuste (overfitting). Estos conceptos y otros más se verán más adelante cuando se trate el tema de generalización de redes neuronales. (Hagan M., 2015).

6.2.3.5 Variaciones del Algoritmo de Retropropagación

El algoritmo básico de retropropagación presentado anteriormente es muy lento para la mayoría de aplicaciones prácticas.

EL algoritmo de retropropagación es un aproximado al algoritmo del descenso por el gradiente. Como se ha visto en el apéndice D el algoritmo de descenso por el gradiente es el más simple y a menudo el método de minimización más lento. El

algoritmo del gradiente conjugado y el método de Newton proveen una convergencia más rápida.

Las investigaciones de algoritmos más rápidos caen más o menos dentro de dos categorías. La primera involucra el desarrollo de técnicas heurísticas tales como: taza de aprendizaje variable, uso de la inercia (momentum) y el preprocesamiento de los datos de entrada. (Hagan M., 2015).

Un detalle simple como utilizar funciones *sigmoideaes bipolares*, por ejemplo en el rango de $[-1,+1]$ (función tangente hiperbólica) en lugar del intervalo $[0,1]$ (función sigmoidea), puede acelerar considerablemente el aprendizaje. (Del Brío, 2007).

Otra circunstancia a tener en cuenta es la magnitud de los pesos iniciales, pues una correcta elección puede suponer un menor tiempo de entrenamiento. Para el caso de la función de activación tangente hiperbólica, el elegir los pesos aleatoriamente en el intervalo $[-2.4/nin, +2.4/nin]$ (siendo nin el número de entradas a la neurona) ya suele dar buenos resultados. (Del Brío, 2007).

En el presente trabajo se discutirá el uso de la inercia (momentum) y la taza de aprendizaje variable.

La segunda categoría abarca las técnicas de optimización numérica o métodos de segundo orden, que se basan en realizar el descenso por el gradiente utilizando también la información proporcionada por el ritmo de cambio de la pendiente, es decir la segunda derivada de la matriz Hesiana. Los algoritmos de gradiente conjugado, Newton y Levenberg Marquadt son ejemplos de ello. En este trabajo se analizará el algoritmo de Levenberg-Marquadt (que es una variación del método de Newton).

Cabe destacar que todos los algoritmos usan el procedimiento de retropropagación, donde las derivadas son procesadas desde la última capa de la red hasta la primera. Por esta razón se denominan algoritmos de *retropropagación*. La diferencia entre los algoritmos radica en la forma en la cual las derivadas resultantes son utilizadas para actualizar los pesos. (Hagan M., 2015).

Para diferenciar el algoritmo básico de retropropagación del resto, se lo denominará retropropagación del descenso por el gradiente (SDBP del inglés steepest descent backpropagation).

6.2.3.5.1 Limitaciones del Algoritmo de Retropropagación

Como se puede ver en el apéndice D en el algoritmo de mínimos cuadrados (LMS) está garantizada la convergencia a una solución que minimiza el error cuadrático medio, siempre y cuando la taza de aprendizaje no sea muy grande. Esto se da ya que el error cuadrático medio para una red de una sola capa es una función cuadrática. La función cuadrática tiene un solo punto estacionario. Adicionalmente el Hessiano de una función cuadrática es constante, por lo tanto la curvatura de la función en una dirección dada no cambia y sus líneas de contorno son elípticas.

El SDBP es una generalización del algoritmo de mínimos cuadrados cuando se usa en una red de una sola capa. Cuando se aplican a redes multicapa, las características del SDBP son muy diferentes. Mientras la hipersuperficie para una red de una sola capa tiene un solo punto mínimo y curvatura constante, la hipersuperficie para una red multicapa podría tener algunos puntos mínimos locales y la curvatura puede variar ampliamente en diferentes regiones del espacio de parámetros. (Hagan M., 2015).

Un estudio detallado de las superficies de desempeño del error cuadrático medio para redes multicapa, como el realizado en la referencia ((Hagan M., 2015) pp:12-3:12-7), proporcionan varias pistas acerca de cómo definir las condiciones iniciales para el algoritmo SDBP.

Primera, no se deben inicializar los parámetros a cero. Ya que el origen del espacio de parámetros puede tener un punto de inflexión en la hipersuperficie. *Segunda*, no se debe inicializar los parámetros con valores grandes. Ya que la hipersuperficie tiende a tener regiones muy planas y alejarse del punto óptimo.

Normalmente se deben elegir para los pesos y bías valores aleatorios y pequeños. De esta forma se puede mantener fuera de un punto de inflexión en el origen y sin moverse en una región muy plana en la hipersuperficie. Además es útil intentar diferentes condiciones iniciales, para estar seguros que el algoritmo converge a un punto mínimo global. (Hagan M., 2015).

6.2.3.5.2 Ejemplo de Convergencia

Para investigar la convergencia del algoritmo de retropropagación se utilizará una red 1-2-1 como la que se mostró en la figura 6.22, con un cambio en la función de transferencia de la salida, en lugar de ser lineal será Sigmoidea.

En esta sección se utilizará la versión por lotes (batching) del algoritmo de retropropagación, en el cual los parámetros son actualizados después de que todos los datos del entrenamiento han sido presentados.

En la figura 6.32 se puede ver dos trayectorias del SDBP (versión por lotes) cuando solo dos parámetros, $w^1_{1,1}$ y $w^2_{1,1}$ son ajustados. Para una condición inicial denominada "a" el algoritmo eventualmente converge a una solución óptima, pero la convergencia es lenta. La razón para que la convergencia sea lenta es el cambio en la curvatura de la superficie a lo largo de la trayectoria. Después de una pendiente inicial moderada, la trayectoria pasa a través de una superficie muy plana, hasta caer dentro de un valle con una pendiente muy suave. Si se *incrementara la tasa de aprendizaje*, la convergencia del algoritmo podría ser rápida mientras pasa por la superficie inicial muy plana, pero podría volverse inestable dentro del valle.

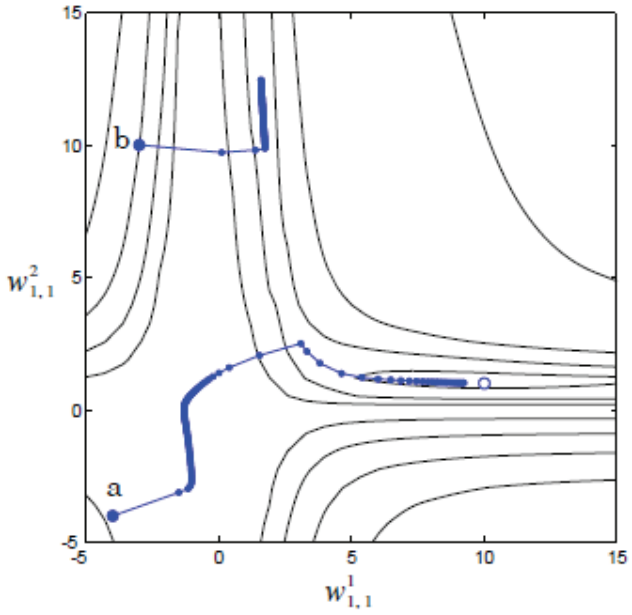


Figura 6.32 Dos Trayectorias del SDBP (Versión por lotes)

((Hagan M., 2015) Pág. 12-8)

La trayectoria denominada “b” ilustra la convergencia del algoritmo a un punto mínimo local. La trayectoria es atrapada en un valle y diverge de la solución óptima. La existencia de múltiples mínimos locales es típica en la hipersuperficie de redes multicapa. Por esta razón se había recomendado anteriormente, intentar con diferentes condiciones iniciales para asegurar que un mínimo global se haya obtenido. (Hagan M., 2015).

El progreso del algoritmo se puede ver en la figura 6.33, que muestra el error cuadrático versus el número de iteraciones.

Se puede notar que las secciones planas de la figura 6.33 corresponden a los momentos cuando el algoritmo atraviesa la sección plana de la hipersuperficie como se muestra en la figura 6.32. En estos períodos se debería incrementar la tasa de aprendizaje, para acelerar la convergencia. Sin embargo, si se incrementa la tasa de aprendizaje el algoritmo se volverá inestable cuando alcance porciones más pronunciadas de la hipersuperficie.

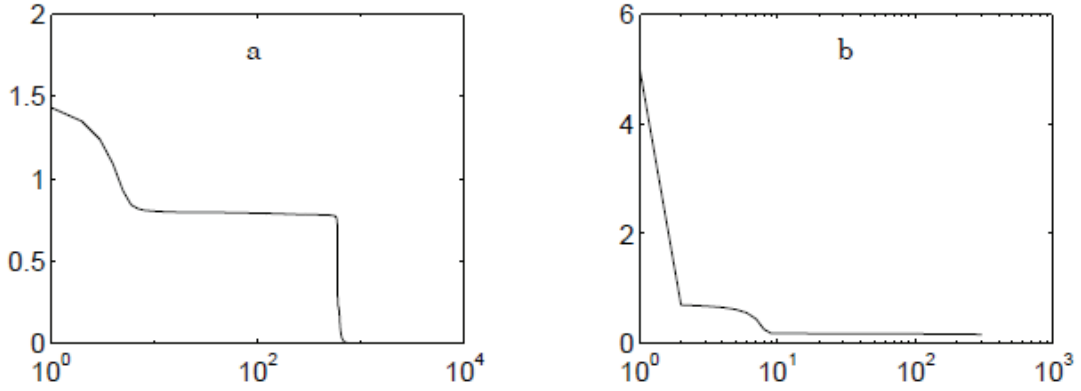


Figura 6.33 Error Cuadrático de las trayectorias “a” y “b”

((Hagan M., 2015) Pág. 12-8)

Este efecto se muestra en la figura 6.34. La trayectoria mostrada aquí corresponde a la trayectoria “a” de la figura 6.32, pero se ha utilizado una tasa de aprendizaje grande. El algoritmo converge rápido al inicio, pero cuando la trayectoria alcanza el valle estrecho que contiene el punto mínimo el algoritmo diverge. Esto sugiere que podría ser útil tener una *tasa de aprendizaje variable*. Se debería incrementar la

tasa en superficies planas y decrecer la misma a medida que la pendiente se incremente.

Otra manera de mejorar la convergencia podría ser *suavizar la trayectoria*. Se debe notar en la figura 6.34 que cuando el algoritmo diverge este está oscilando a través de un valle estrecho. Si se filtraría la trayectoria promediando las actualizaciones de los parámetros, esto podría suavizar las oscilaciones y producir una trayectoria estable. (Hagan M., 2015).

6.2.3.5.3 Modificaciones Heurísticas al Algoritmo de Retropropagación

Después de haber visto algunas limitaciones del algoritmo de retropropagación (descenso por el gradiente), se considerarán dos métodos heurísticos para mejorar el algoritmo.

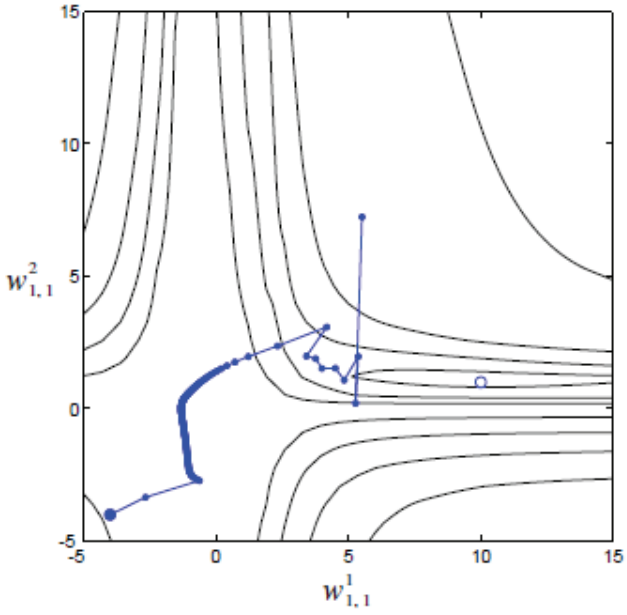


Figura 6.34 Trayectoria con un tasa de aprendizaje grande

((Hagan M., 2015) Pág. 12-8)

6.2.3.5.4 Inercia (Momentum)

Como se ha visto en la sección anterior la convergencia se puede mejorar si se suavizan las oscilaciones de la trayectoria. Esto se puede lograr con un filtro pasa bajos.

Se considerará el siguiente filtro de primer orden, para luego aplicarlo a las redes neuronales.

$$y(k) = \gamma y(k-1) + (1-\gamma)w(k), \tag{6.56}$$

Donde $w(k)$ es la entrada al filtro, $y(k)$ es la salida del filtro y γ es el coeficiente de inercia que debe satisfacer

$$0 \leq \gamma \leq 1 \tag{6.57}$$

El efecto de este filtro se muestra en la figura 6.35. Se puede notar que la oscilación a la salida del filtro es menor que la oscilación a la entrada del mismo (como se espera en un filtro pasa bajos). Adicionalmente cuando γ se incrementa la oscilación en la salida del filtro se reduce y responde más lento. Es decir, el filtro tiende a reducir las oscilaciones mientras mantiene al valor promedio.

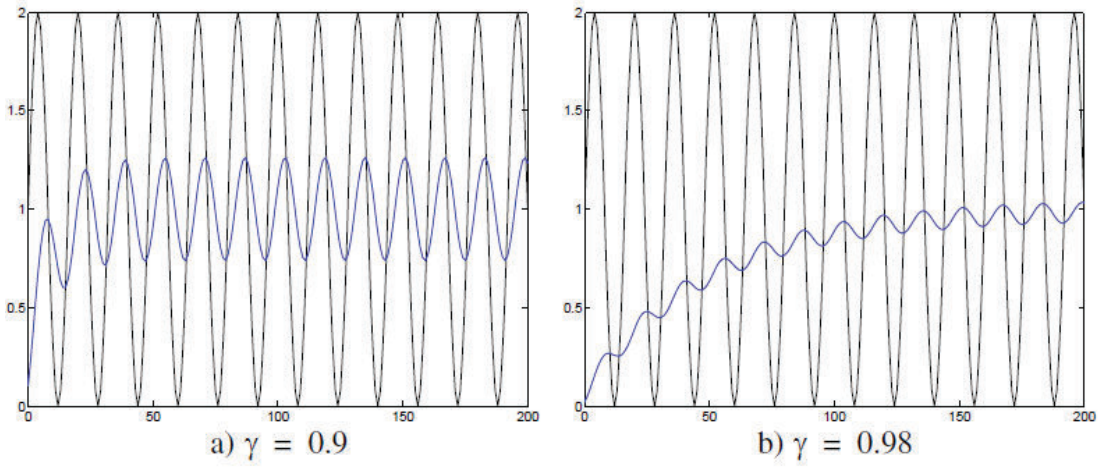


Figura 6.35 Efecto de Suavizamiento por Inercia

((Hagan M., 2015) Pág. 12-10)

Ahora se verá como este filtro trabaja en las redes neuronales. Primero se recordará la actualización de los parámetros para el algoritmo de SDBP ecuaciones (6.26) y (6.27):

$$\Delta \mathbf{W}^{\mathrm{m}}(k) = - \propto \mathbf{s}^{m} \, (\mathbf{a}^{m-1})^{T} \, , \tag{6.58}$$

$$\Delta \mathbf{b}^{m}(k) = - \propto \mathbf{s}^{m} \, . \tag{6.59}$$

Cuando el filtro de inercia o momento se incluye al cambio en los parámetros se obtienen las siguientes ecuaciones para el algoritmo de retropropagación modificado por la inercia (MOBP del inglés momentum modification to backpropagation):

$$\Delta \mathbf{W}^{\mathrm{m}}(k) = \gamma \Delta \mathbf{W}^{\mathrm{m}}(k-1) - (1-\gamma) \propto \mathbf{s}^{m} \, (\mathbf{a}^{m-1})^{T} \, , \tag{6.60}$$

$$\Delta \mathbf{b}^{m}(k) = \gamma \Delta \mathbf{b}^{m}(k-1) - (1-\gamma) \propto \mathbf{s}^{m}. \tag{6.61}$$

Si se aplican estas ecuaciones modificadas al mismo ejemplo de convergencia anterior, se obtienen los resultados mostrados en la figura 6.36. Esta trayectoria corresponde a las mismas condiciones iniciales y tasa de aprendizaje mostradas en la figura 6.34, pero con un coeficiente de inercia de $\gamma = 0.8$. Se puede notar que el algoritmo se estabiliza. Por el uso de la inercia se pueden utilizar tasas de aprendizaje mayores mientras el algoritmo mantiene su estabilidad. Otra característica de la inercia es que tiende a acelerar la convergencia cuando la trayectoria se mueve en una dirección consistente. Además mientras mayor sea el valor de γ, más “inercia” tendrá la trayectoria. (Hagan M., 2015).

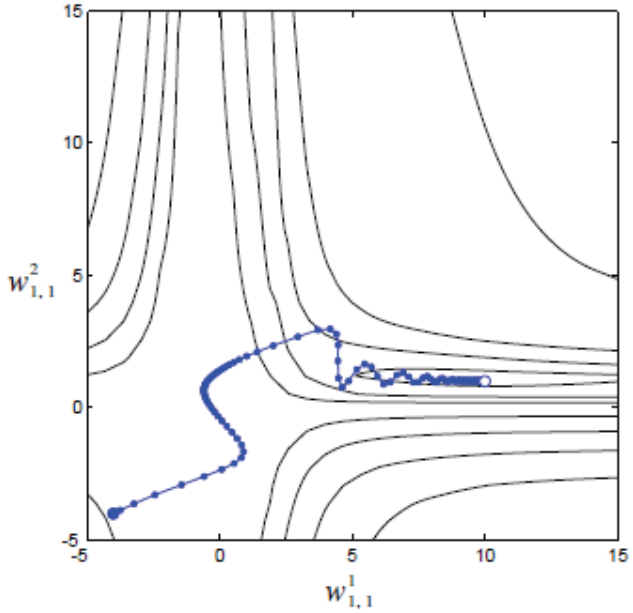


Figura 6.36 Trayectoria con Inercia

((Hagan M., 2015) Pág. 12-11)

6.2.3.5.5 Tasa de Aprendizaje Variable

Como se había visto anteriormente se puede acelerar la convergencia si se incrementa la tasa de aprendizaje en superficies planas y luego se disminuye la misma cuando la pendiente se incrementa.

Se debe recordar que la hipersuperficie del error cuadrático medio para redes lineales de una sola capa es siempre una función cuadrática y la matriz Hessiana es por lo tanto constante. La tasa máxima de aprendizaje para el algoritmo del descenso por el gradiente es dos dividido para el máximo valor propio (λ_{Max}) de la matriz Hessiana. (ver apéndice D).

Como se ha visto, la hipersuperficie del error en una red multicapa no es una función cuadrática. La forma de la superficie puede ser muy diferente en las diferentes regiones del espacio de parámetros. Quizás se pueda acelerar la convergencia ajustando la tasa de aprendizaje durante la etapa de entrenamiento. El truco será determinar cuándo cambiar la tasa de aprendizaje y por cuánto.

A continuación se describirá el procedimiento por lotes (batch), donde la tasa de aprendizaje se varía acorde con el desempeño del algoritmo.

Las reglas para el algoritmo de retropropagación con tasa de aprendizaje variable (VLBP del inglés variable learning rate backpropagation algorithm) son:

1. Si el error cuadrático (sobre todos los datos de entrenamiento) se incrementa por más de cierto porcentaje ξ (típico entre el uno al cinco por ciento) después de una actualización de pesos, entonces esa actualización es descartada, la tasa de aprendizaje es multiplicada por un factor $0 < \rho < 1$ y el coeficiente de inercia γ (si es usado) se pone a cero.
2. Si el error cuadrático decrece después de una actualización de pesos, entonces la actualización de pesos es aceptada y la tasa de aprendizaje se multiplica por factos $\eta > 1$. Si γ fue puesto a cero, se lo regresa al valor original.
3. Si el error cuadrático se incrementa menos de ξ, entonces la actualización de los pesos se acepta pero la tasa de aprendizaje no se modifica. Si γ fue puesto a cero, se lo regresa al valor original.

En la figura 6.37 se muestra la trayectoria del algoritmo usando las mismas condiciones iniciales, tasa de aprendizaje inicial y coeficiente de inercia, que se utilizaron en la figura 6.36. Los nuevos parámetros se asignaron con los valores:

$\eta = 1.05$, $\rho = 0.7$ y $\xi = 4\%$. (6.62)

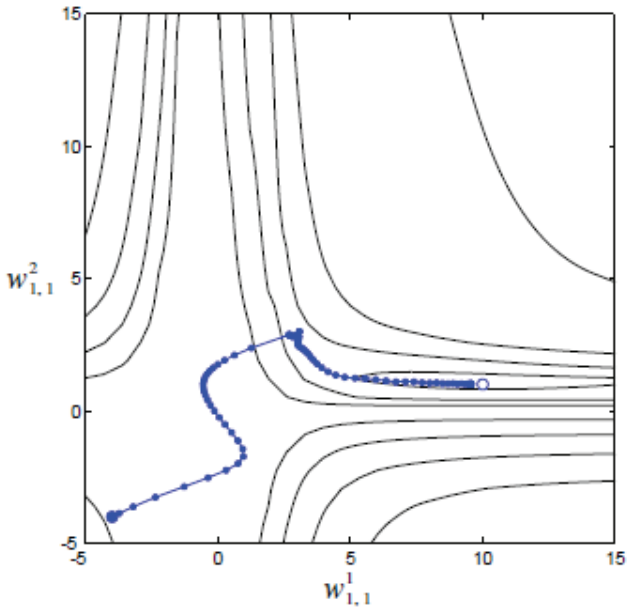


Figura 6.37 Trayectoria con Tasa de Aprendizaje Variable

((Hagan M., 2015) Pág. 12-13)

Se puede notar en la figura 6.37 como la tasa de aprendizaje y por lo tanto el tamaño del paso, tienden a incrementarse cuando la trayectoria viaja en línea recta mientras el error va decreciendo.

Cuando la trayectoria alcanza un valle estrecho, la tasa de aprendizaje es rápidamente disminuida. Caso contrario la trayectoria se vuelve oscilatoria y el error se incrementa drásticamente. Para cada paso donde el error se ha incrementado por más del 4% la tasa de aprendizaje se reduce y la inercia es eliminada, lo que permite cambiar rápidamente la trayectoria para seguir a través del valle hacia el punto mínimo. La tasa de aprendizaje entonces se incrementa nuevamente, lo que acelera la convergencia. La tasa de aprendizaje es reducida nuevamente cuando la trayectoria excede el punto mínimo cuando el algoritmo casi converge. Este proceso es típico de una trayectoria VLBP.

Existen algunas variaciones de la tasa de aprendizaje variable como son: delta-bar-delta, SuperSAB, Quickprop de Fahlman, entre otros.

Las modificaciones heurísticas al SDBP pueden a menudo proveer una convergencia rápida para algunos problemas. Sin embargo existen dos limitaciones para estos métodos. *La primera* es que las modificaciones requieren de algunos parámetros ($\xi, \gamma, \rho \ldots$), mientras el SDBP requiere un solo parámetro, la tasa de aprendizaje. A menudo el desempeño del algoritmo es sensible al cambio en estos parámetros. La elección de los parámetros es dependiente del problema. *La segunda* limitación de estas modificaciones al SDBP es que a veces pueden fallar en la convergencia de problemas para los cuales el SDBP eventualmente encontrará una solución. Mientras más complejos son los algoritmos estas limitaciones tienden a ocurrir más a menudo. (Hagan M., 2015).

6.2.3.5.6 Técnicas de Optimización Numéricas

Además de las modificaciones heurísticas al SDBP analizadas arriba, existen métodos que se basan en técnicas de optimización numéricas estándar. Tales como: gradiente conjugado, gradiente conjugado escalado, método de Newton, algoritmo de Levenberg–Marquardt, entre otros. En la referencia (Beale M., 2015)(pp:9-16 a 9-30), se hace una muy buena comparación entre nueve algoritmos de entrenamiento para redes neuronales en diferentes aplicaciones, se puede concluir de este estudio que para el caso de aproximación de funciones el algoritmo de Levenberg-Marquardt es el mejor, especialmente si la red no tiene una gran cantidad de parámetros. Ya que los pronósticos caen dentro de la categoría de aproximación de funciones en el presente trabajo se investigará el algoritmo de Levenberg–Marquardt que es una variación del método de Newton, este algoritmo es muy seguro para el entrenamiento de redes neuronales.

6.2.3.5.6.1 Algorimto de Levenberg-Marquardt

Este algoritmo fue diseñado para minimizar funciones que pueden ser no lineales. Este método es muy seguro para el entrenamiento de redes neuronales donde el índice de desempeño es el error cuadrático medio.

6.2.3.5.6.2 Algoritmo Básico

En el apéndice D se muestra el método de Newton, por comodidad se repetirá la parte principal.

El método de Newton para optimizar el índice de desempeño $F(\boldsymbol{x})$ es

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1}\,\mathbf{g}_k, \tag{6.63}$$

Donde $\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ y $\mathbf{g}_k \equiv \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$.

Si se asume que $F(\mathbf{x})$ es la función suma de cuadrados:

$$F(\mathbf{x}) = \sum_{i=1}^{N} v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}) \tag{6.64}$$

Entonces el $j-ésimo$ elemento del gradiente sería

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2\sum_{i=1}^{N} v_i(\mathbf{x})\frac{\partial v_i(\mathbf{x})}{\partial x_j}. \tag{6.65}$$

En forma matricial quedaría:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \tag{6.66}$$

Donde

$$\mathbf{J}(\mathbf{x}) \equiv \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{6.67}$$

Es la matriz Jacobiana o Jacobiano.

Luego se busca la matriz Hessiana o Hessiano. El k,j elemento del Hessiano sería

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2\sum_{i=1}^{N} \frac{\partial v_i(\mathbf{x})}{\partial x_k}\frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x})\frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \tag{6.68}$$

La matriz Hessiana puede ser expresada en forma matricial:

$$\nabla^2 F(\mathbf{x}) = 2\,\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}) \tag{6.69}$$

Donde

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^{N} v_i(\mathbf{x})\,\nabla^2 v_i(\mathbf{x})\,. \tag{6.70}$$

Si se asume que $\mathbf{S}(\mathbf{x})$ es muy pequeño, se puede aproximar el Hessiano como

$$\nabla^2 F(\mathbf{x}) \cong 2\,\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x})\,. \tag{6.71}$$

Si se sustituye la Ec.(6.71) y la Ec. (6.66) en la Ec.(6.63) se obtiene el método de *Gauss – Newton*:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [2\,\mathbf{J}^T(\mathbf{x}_k)\,\mathbf{J}(\mathbf{x}_k)]^{-1}\,2\,\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k),$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\,\mathbf{J}^T(\mathbf{x_k})\,\mathbf{J}(\mathbf{x_k})]^{-1}\;\mathbf{J}^T(\mathbf{x_k})\mathbf{v}(\mathbf{x_k}), \tag{6.72}$$

Se puede notar que la ventaja del método de Gauss - Newton sobre el método de Newton estándar es que este no requiere le cálculo de las segundas derivadas.

Un problema con el método de Gauss - Newton es que la matriz $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ puede no ser invertible. Esto se puede superar utilizando la siguiente modificación a la matriz Hessiana aproximada:

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I} \tag{6.73}$$

Para averiguar si esta matriz es invertible se puede suponer que los valore propios y vectores propios de $\mathbf{H}$ son $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ y $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$. Enronces

$$\mathbf{G}\mathbf{z}_i = [\mathbf{H} + \mu\mathbf{I}] = \mathbf{H}\mathbf{z}_i + \mu\mathbf{z}_i = \lambda_i\mathbf{z}_i + \mu\mathbf{z}_i = (\lambda_i + \mu)\,\mathbf{z}_i\ . \tag{6.74}$$

Por lo tanto los vectores propios de $\mathbf{G}$ son los mismos que $\mathbf{H}$ y los valores propios de $\mathbf{G}$ son $(\lambda_i + \mu)$. $\mathbf{G}$ puede ser transformada a matriz definida positiva aumentando μ hasta que $(\lambda_i + \mu) > 0$ para todo i, y por lo tanto la matriz podría ser invertible.

Esto conduce al algoritmo de ***Levenberg-Marquardt*** (Hagan M., 2015):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) + \mu_k\mathbf{I}]^{-1}\,\mathbf{J}^T(\mathbf{x_k})\mathbf{v}(\mathbf{x_k}), \tag{6.75}$$

o

$$\Delta\mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x_k})\mathbf{J}(\mathbf{x_k}) + \mu_k\mathbf{I}]^{-1}\,\mathbf{J}^T(\mathbf{x_k})\mathbf{v}(\mathbf{x_k}), \tag{6.76}$$

Este algoritmo tiene una característica muy útil que cuando μ_k se incrementa este se aproxima al algoritmo del descenso por el gradiente con una tasa de aprendizaje pequeña:

$$\mathbf{x}_{k+1} \cong \mathbf{x}_k - \frac{1}{\mu_k}\mathbf{J}^T(\mathbf{x_k})\mathbf{v}(\mathbf{x_k}) = \mathbf{x}_k - \frac{1}{2\mu_k}\nabla F(\mathbf{x}),\ para\ \mu_k\ grandes \tag{6.77}$$

Y cuando μ_k se disminuye a cero el algoritmo se convierte en Gauss-Newton.

Este algoritmo comienza con μ_k asignado con algún valor pequeño (ej. $\mu_k = 0.01$). Si esta iteración no produce un valor pequeño de $F(\boldsymbol{x})$, entonces se repite la iteración con μ_k multiplicado por algún factor $\varrho > 1$ (ej. $\varrho > 10$). Eventualmente $F(\boldsymbol{x})$ podría decrecer, ya que estaría tomando un pequeño paso en la dirección del descenso por el gradiente. Si la iteración produce un valor pequeño de $F(\boldsymbol{x})$, entonces μ_k es dividido para ϱ para la siguiente iteración, así el algoritmo se aproxima a Gauss-Newton, el cual podría proveer una rápida convergencia.

Este algoritmo provee un compromiso entre velocidad del método de Newton y convergencia garantizada del descenso por el gradiente.

A continuación se presenta como aplicar el algoritmo de Levenberg – Marquardt al entrenamiento de redes neuronales multicapa.

El índice de desempeño para el entrenamiento redes multicapa es el error cuadrático medio

$$F(\mathbf{x}) = E[\mathbf{e}^T\ \mathbf{e}] = E[(\mathbf{t}-\mathbf{a})^T(\mathbf{t}-\mathbf{a})] \quad (6.78)$$

Si cada salida deseada (target) ocurre con igual probabilidad, el error cuadrático medio es proporcional a la suma de los cuadrados de los errores sobre las Q salidas deseadas en los datos para el entrenamiento.

$$F(\mathbf{x}) = \sum_{q=1}^{Q}(\mathbf{t_q}-\mathbf{a_q})^T(\mathbf{t_q}-\mathbf{a_q})$$

$$F(\mathbf{x}) = \sum_{q=1}^{Q}(\mathbf{e_q})^T(\mathbf{e_q}) = \sum_{q=1}^{Q}\sum_{j=1}^{S^M}(e_{j,q})^2 = \sum_{i=1}^{N}(v_i)^2\ , \quad (6.79)$$

Donde $e_{j,q}$ es el $j-ésimo$ elemento del error para el $q-ésimo$ par entrada/salida deseada.

La ecuación (6.79) es equivalente al índice de desempeño, Ec. (6.64), para el cual el algoritmo de Levenberg-Marquardt fue diseñado.

Cálculo de Jacobiano

Un paso clave en el algoritmo de Levenberg-Marquardt es el cálculo del Jacobiano. Para realizar este cálculo se utilizará una variación del algoritmo de retropropagación. Ya que en el algoritmo de retropropagación estándar se calculan las derivadas de los errores al cuadrado respecto a los pesos y bías de la red. Para crear el Jacobiano se necesita el cálculo de las derivadas de los errores, en lugar de las derivadas del cuadrado de los errores.

En la Ec. (6.67) el vector error es

$$\mathbf{v}^T = [v_1\ v_2\ \dots v_N] = \left[e_{1,1}\ e_{2,1}\ \dots\ e_{S^M,1}\ e_{1,2}\ \dots e_{S^M,Q}\right], \quad (6.80)$$

El vector de parámetros es

$$\mathbf{x}^T = [x_1\ x_2\ \dots x_N] = [w^1_{1,1}\ w^1_{1,2}\ \dots\ w^1_{S^1,R}\ b^1_1\ \dots\ b^1_{S^1}\ w^2_{1,1}\ \dots\ b^M_{S^M}]\ , \quad (6.81)$$

$N = QxS^M$ y $n = S^1(R+1) + S^2(S^1+1) + \cdots + S^M(S^{M-1}+1)$.

Por lo tanto, si se hacen las respectivas sustituciones en la Ec. (6.67) el Jacobiano para el entrenamiento de una red multicapa quedaría:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \cdots \\ \vdots & \vdots & & \vdots & & \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \cdots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \cdots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \cdots \end{bmatrix} \tag{6.82}$$

Los términos de este Jacobiano pueden ser calculados mediante una simple modificación al algoritmo de retropropagación.

El algoritmo de retropropagación estándar calcula términos como

$$\frac{\partial \hat{\mathbf{F}}(x)}{\partial x_l} = \frac{\partial \mathbf{e}_q^T \mathbf{e}_q}{\partial x_l} \tag{6.83}$$

Para el Jacobiano del algoritmo de Levenberg – Marquardt se necesitan calcular términos como

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l}. \tag{6.84}$$

Si se observa la Ec. (6.18) del algoritmo de retropropagación se tiene

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} x \frac{\partial n_i^m}{\partial b_i^m} \tag{6.85}$$

Se había definido la sensibilidad como

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m} \tag{6.86}$$

El algoritmo de retropropagación calcula las sensibilidades a través de una relación recursiva desde la última capa hacia atrás hasta la primera capa. Se utilizará este mismo concepto para calcular los términos del Jacobiano Ec.(6.82). Si se define una nueva *sensibilidad de Marquardt* como:

$$\tilde{s}_{i,h}^m \equiv \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \tag{6.87}$$

De la Ec.(6.81) se tiene que, $h = (q-1)S^M + k$.

Ahora se puede calcular los elementos de Jacobiano mediante

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} x \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m x \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = \tilde{s}_{i,h}^m x a_{j,q}^{m-1}, \tag{6.88}$$

o si x_l es el bías,

$$[\mathbf{J}]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial b_i^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} x \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m x \frac{\partial n_{i,q}^m}{\partial b_i^m} = \tilde{s}_{i,h}^m \ . \tag{6.89}$$

Las *sensibilidades de Marquardt* se calculan de la misma forma que las sensibilidades estándar (Ec. 6.34) con una modificación en la última capa, donde para la retropropagación estándar se calcula mediante la Ec. (6.39). Para las sensibilidades de Marquardt en la última capa se tiene

$$\tilde{s}_{i,h}^{M} = \frac{\partial v_h}{\partial n_{i,q}^{M}} = \frac{\partial e_{k,q}}{\partial n_{i,q}^{M}} = \frac{\partial (t_{k,q} - a_{k,q}^{M})}{\partial n_{i,q}^{M}} = \frac{\partial a_{k,q}^{M}}{\partial n_{i,q}^{M}}$$

$$= \begin{cases} -\dot{f}^{M}(n_{i,q}^{M}) \; para \; i = k \\ 0 \qquad\qquad para \; i \neq k \end{cases} \tag{6.90}$$

Por lo tanto cuando la entrada $\mathbf{p}_q$ ha sido aplicada a la red y la correspondiente salida de la red ha sido calculada, la retropropagación de Levenberg – Marquardt se inicializa con

$$\tilde{\mathbf{S}}_q^M = -\dot{\mathbf{F}}^M(\mathbf{n}_q^M), \tag{6.91}$$

Donde $\dot{\mathbf{F}}^M(\mathbf{n}^M)$ se definió en la Ec. (6.33). Cada columna de la matriz $\tilde{\mathbf{S}}_q^M$ debe ser retro propagada a través de la red utilizando la Ec.(6.34) para obtener una fila del Jacobiano. Las columnas también pueden ser retro propagadas juntas utilizando

$$\tilde{\mathbf{S}}_q^m = -\dot{\mathbf{F}}^m(\mathbf{n}_q^m)(\mathbf{W}^{m+1})^T \tilde{\mathbf{S}}_q^{m+1} \tag{6.92}$$

Las matrices de sensibilidades de Marquardt totales para cada capa se crean aumentando las matrices calculadas para cada entrada:

$$\tilde{\mathbf{S}}^m = [\tilde{\mathbf{S}}_1^m | \tilde{\mathbf{S}}_2^m | \ldots | \tilde{\mathbf{S}}_Q^m] \; . \tag{6.93}$$

Se debe notar que para cada entrada que es presentada a la red se retro propagarán S^M vectores de sensibilidad. Esto se debe a que se calculan las derivadas de cada error individualmente en lugar de la suma de los cuadrados de los errores. Por cada entrada aplicada a la red habrá S^M errores (uno por cada elemento de la salida de la red). Por cada error habrá una fila del Jacobiano.

Después que las sensibilidades han sido retro propagadas, el Jacobiano se calcula utilizando las Ecs. (6.88) y (6.89).

El algoritmo de retropropagación de Levenberg-Marquardt (LMBP del inglés Levenberg-Marquardt Backpropagation) se puede *resumir* como sigue:

1.- Presentar todas las entradas a la red y calcular sus correspondientes salidas (utilizando las ecuaciones (6.40) y (6.41)) y los errores $\mathbf{e}_q = \mathbf{t}_q - \mathbf{a}_q^M$. Calcular la suma de los errores al cuadrado sobre todas las entradas, $F(\boldsymbol{x})$, usando la Ecuación (6.79).

2.- Calcular la matriz Jacobiana, Ec. (6.82). Calcule las sensibilidades con la relación recurrente (6.92), después de inicializar con Ec. (6.91). Aumente las matrices individuales para formar la sensibilidad de Marquardt utilizando Ec. (6.93). Calcule los elementos del Jacobiano con Ec.(6.88) y Ec. (6.89).

3.- Resuelva la Ec.(6.76) para obtener $\Delta \mathbf{x}_k$.

4.- Recalcule la suma de los errores al cuadrado usando $\mathbf{x}_k + \Delta \mathbf{x}_k$. Si esta suma de los cuadrados es menor que la calculada en paso 1, entonces divide $\mu \, para \, \varrho$, dando $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_k$ y regrese al paso 1. Si la suma de los cuadrados no se reduce, entonces multiplique $\mu \, por \, \varrho$ y regrese al paso 3.

Se asume que el algoritmo converge cuando la norma del gradiente Ec.(6.66), es menor que algún valor predeterminado, o cuando la suma de los cuadrados ha alcanzado algún objetivo de error.

Para ilustrar el LMBP se lo aplicará al problema de aproximación de funciones de la sección (6.2.3.2). La figura 6.38 ilustra las posibles direcciones que el algoritmo LMBP podría tomar en la primera iteración.

La flecha negra representa la dirección que tomaría para μ_k pequeño, que corresponde a la dirección de Gauss-Newton. *La flecha azul* representa la dirección que tomaría para μ_k grande, que corresponde a la dirección del Descenso por el Gradiente. La curva azul representa la trayectoria del algoritmo LMBP para valores intermedios de μ_k. Nótese que cuando μ_k se incrementa el algoritmo se mueve una pequeña distancia en la dirección del descenso por el gradiente.

Esto garantiza que el algoritmo siempre podrá reducir la suma de los cuadrados en cada iteración

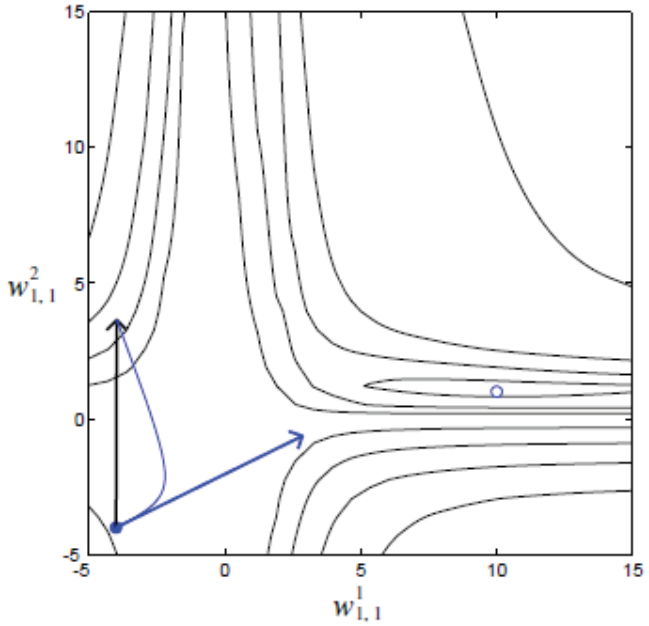


Figura 6.38 Posibles Direcciones Algoritmo de Levenberg-Marquardt

((Hagan M., 2015) Pág. 12-26)

La figura 6.39 muestra la trayectoria del LMBP hasta que converge, con $\mu_0 = 0.01$ y $\varrho = 5$

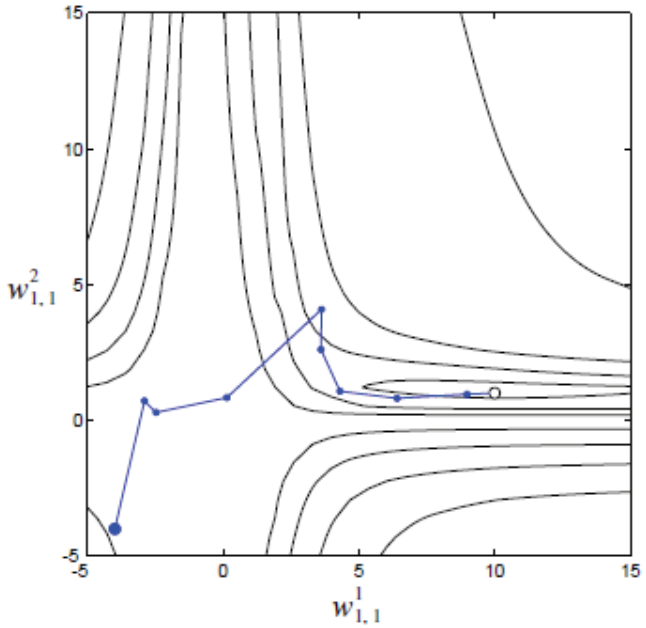


Figura 6.39 Trayectoria LMBP

((Hagan M., 2015) Pág. 12-26)

Nótese que el algoritmo converge en pocas iteraciones. Claro el algoritmo requiere más cálculos por iteración que cualquier otro algoritmo, ya que necesita invertir una matriz. Pese al gran número de cálculos el algoritmo LMPBP es el más rápido para el entrenamiento de redes neuronales moderadas (en número de parámetros). (Hagan M., 2015).

Una limitación del algoritmo LMBP es el requerimiento de almacenamiento. El algoritmo debe almacenar la matriz Hessiana aproximada ($\mathbf{J}^T\mathbf{J}$)**.** Esto es una matriz nxn, donde n es número de parámetros (pesos y bías) en la red. Otros métodos solamente necesitan almacenar el gradiente, que es un vector $n - dimensional$. Cuando el número de parámetros es muy grande, se vuelve impracticable el uso del algoritmo LMBP. (¿Qué es "muy grande"? esto depende de la cantidad de memoria disponible en el computador, pero algo típico es unos pocos miles de parámetros como límite superior.) (Hagan M., 2015).

Es decir cuando el número de parámetros es muy grande métodos como el gradiente conjugado es preferible a LMBP en términos computacionales. (Haykin, 1999).

6.2.4 GENERALIZACIÓN

Uno de los puntos clave en el diseño de redes multicapa es determinar el número de neuronas a usar.

Anteriormente se ha visto que si el número de neuronas es demasiado grande, la red sobre ajustará los datos de entrenamiento. Esto significa que el error en los datos de entrenamiento será muy pequeño, pero la red fallará en su ejecución cuando nuevos datos sean presentados. Una red que generaliza bien se desempeñará bien con nuevos datos como con los datos de entrenamiento.

La complejidad de la red está determinada por el número de parámetros libres que esta tiene (pesos y bías), que a su vez está determinado por el número de neuronas. Si una red es muy compleja para un conjunto de datos dados, es muy probable que sobreajuste y tenga una generalización pobre.

En esta sección se verá que se puede determinar la complejidad de la red para ajustar con la complejidad de los datos. Adicionalmente, esto se puede hacer sin

cambiar el número de neuronas. Se puede ajustar el número efectivo de parámetros sin cambiar el actual número de parámetros libres. (Hagan M., 2015).

Se dice que una red *generaliza* bien cuando la relación entrada-salida (mapping) calculado por la red es correcto para datos de prueba nunca utilizados ni en la creación ni en el entrenamiento de la red. (Haykin, 1999).

La estrategia clave es *ajustar el tamaño de la red a la complejidad del problema que se está tratando*, debiéndose limitar en lo posible su tamaño (principio de Occam o de máxima economía de medios (Haykin, 1999), (Del Brío, 2007).

En términos de redes neuronales, el modelo más simple es aquel que tiene el menor número de parámetros (pesos y bías), o, equivalentemente el menor número de neuronas. Para encontrar una red que *generalice* bien se necesita encontrar la red más simple que ajuste los datos. (Hagan M., 2015).

Existen al menos cinco enfoques diferentes que se han utilizado para producir redes más simples: crecimiento (growing), podado (pruning), búsquedas globales (global searches), regularización y parada temprana (early stopping). En los métodos de crecimiento se inicia sin neuronas en la red y se va adicionando neuronas hasta que el desempeño sea adecuado. Los métodos de podado inician con redes grandes, que probablemente sobre ajusten, y entonces se remueven neuronas una a la vez hasta que el desempeño se degrade significativamente. Si el lector desea profundizar en estos dos métodos, puede consultar la referencia ((Haykin, 1999) pp 218:226).

Búsquedas globales, tal como algoritmos genéticos, buscan en el espacio de todas las posibles arquitecturas de red para localizar el modelo más simple que ajuste los datos.

Los dos enfoques finales, regularización y parada temprana, mantienen la red pequeña limitando la *magnitud* de los pesos de la red en lugar de limitar el número de pesos de la red. (Hagan M., 2015). En esta sección se investigará estos dos métodos.

6.2.4.1 Planteamiento del Problema

El problema de la generalización se inicia con un conjunto de datos para entrenamiento que consisten en entradas $\mathbf{p}$ y sus correspondientes salidas deseadas $\mathbf{t}$.

$$\{\mathbf{p}_1|\mathbf{t}_1\},\{\mathbf{p}_2|\boldsymbol{t}_2\},\dots,\{\mathbf{p}_Q|\mathbf{t}_Q\}\ . \tag{6.94}$$

Para el desarrollo del concepto de generalización, se asumirá que las salidas deseadas son generadas por:

$$\mathbf{t}_q = \mathbf{g}(\mathbf{p}_q) + \varepsilon_q\ , \tag{6.95}$$

Donde $\mathbf{g}(.)$ Es una función desconocida y ε_q es una fuente de ruido aleatorio, independiente y de media cero. El objetivo del entrenamiento será entonces producir un red neuronal que aproxime $\mathbf{g}(.)$, mientras ignora el ruido.

El índice de desempeño estándar para el entrenamiento de la red neuronal es la suma de los errores al cuadrado en los datos de entrenamiento:

$$F(\mathbf{x}) = E_D = \sum_{q=1}^{Q}(\mathbf{t_q} - \mathbf{a_q})^T(\mathbf{t_q} - \mathbf{a_q}) \tag{6.96}$$

Donde $\mathbf{a_q}$ es la salida de la red para una entrada $\mathbf{p}_q$. Se asume que la variable E_D representa suma de los errores al cuadrado en los datos de entrenamiento, ya que luego se modificará el índice de desempeño para incluir un término adicional.

Sobreajuste

El problema del sobreajuste (*overfitting*) se ilustra en la figura 6.40. La curva azul representa la función $\mathbf{g}(.)$. La curva en negro representa la respuesta de la red entrenada y los círculos pequeños con cruces representan la respuesta de la red para los puntos de entrenamiento. En esta figura se puede ver que la respuesta de la red coincide exactamente con los puntos de entrenamiento. Pero se nota una pobre coincidencia con la función subyacente. Esto es sobreajuste.

En la figura (6.40) se pueden reconocer dos tipos de errores: *El primer tipo de error*, es causado por el sobreajuste, ocurre en el intervalo entre $[-3\ y\ 0]$. Esta es la región de entrenamiento. La respuesta de la red en esta región sobre ajusta los datos de entrenamiento y fallará en ajustar valores de entrada que no estén en los datos de entrenamiento. La red hace un trabajo pobre de interpolación; esto es falla

en aproximarse a la función de una manera precisa cerca de los puntos de entrenamiento. (Hagan M., 2015)

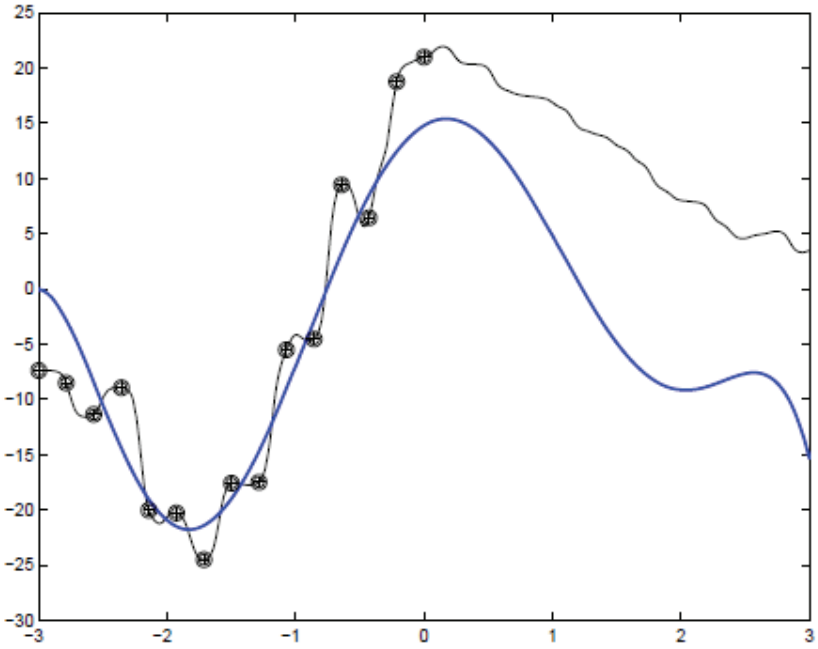


Figura 6.40 Ejemplo de sobreajuste y pobre generalización

((Hagan M., 2015) Pág. 13-4)

El segundo tipo de error ocurre para entradas en el intervalo entre $[0\ y\ 3]$. La red falla su ejecución en esta región, no por sobreajuste, sino porque no hay datos de entrenamiento ahí. La red está *extrapolando* fuera del rango de los datos de entrada.

En esta sección se discutirá métodos para prevenir los errores de interpolación (sobreajuste). No hay forma de prevenir errores de extrapolación, a menos que los datos utilizados en el entrenamiento cubran todas las regiones del espacio de entrada donde se utilizará la red. La red no tiene manera de saber cómo es la verdadera función en regiones donde no hay datos.

En la figura 6.41 se muestra un ejemplo donde una red ha sido entrenada para generalizar bien. La red tiene el mismo número de parámetros que la de la figura 6.40, y fue entrenada con los mismos datos, pero fue entrenada de manera que no

usa todos los pesos disponibles. Esta usa solo los pesos necesarios para ajustar los datos. La respuesta no es perfecta pero mejora mucho. (Hagan M., 2015).

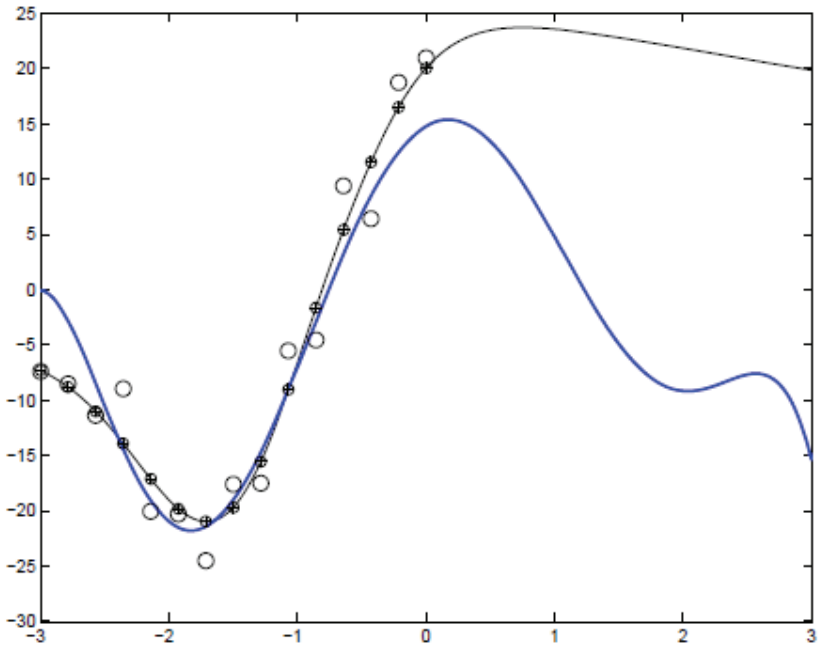


Figura 6.41 Ejemplo de buena interpolación y pobre extrapolación
((Hagan M., 2015) Pág. 13-4)

6.2.4.2 Métodos para Mejorar la Generalización

Como se ha mencionado antes existen algunos métodos para mejorar la generalización de una red neuronal, todos ellos tratan de encontrar la red más simple que ajuste adecuadamente los datos. Estos métodos se los puede agrupar en dos categorías generales: restringiendo el número de pesos (lo que equivale a reducir el número de neuronas) en la red o restringir la magnitud de los pesos. Se analizarán dos métodos que son los más utilizados: Parada Temprana y Regularización. Los dos métodos intentan restringir la magnitud de los pesos.
Una aclaración importante es que se asumirá que se tiene una cantidad limitada de datos para el entrenamiento de la red. Si la cantidad de datos fuera ilimitada, lo que en términos prácticos significaría tener muchos más datos en comparación con el

número de parámetros de la red, no se presentaría en problema de sobreajuste. (Hagan M., 2015).

La esencia de los algoritmos de aprendizaje es codificar la relación entrada-salida (representada por una serie de ejemplos o datos) en los pesos y bías de la red multicapa. El objetivo es tener una red bien entrenada para que aprenda lo suficiente del pasado para generalizar a futuro. (Haykin, 1999).

Dada una cantidad limitada de datos disponibles, es importante separar un conjunto de datos durante el proceso de entrenamiento, llamado datos de prueba o test.

En el proceso de entrenamiento se debe considerar dos tipos de errores: el error de entrenamiento y el error de generalización. El error de entrenamiento se lo calcula como el error cuadrático medio de los resultados proporcionados por la red para los datos de entrenamiento. El error de generalización, se lo puede medir utilizando un conjunto de datos (datos de prueba) diferentes a los utilizados en el entrenamiento. De esta forma, se puede entrenar una red neuronal mediante un conjunto de datos de entrenamiento y comprobar su eficacia real, o error de generalización, mediante un conjunto de prueba o test. (Del Brío, 2007).

Para que los datos de prueba sean un indicador válido de la generalización de la red se deben tener en cuenta dos cosas importantes: *La primera*, los datos de prueba o test nunca deben haber sido utilizados para entrenar la red, este conjunto de datos se deben utilizar cuando el entrenamiento haya terminado. *La segunda*, los datos de prueba deben representar todas las situaciones para las cuales la red será utilizada. (Hagan M., 2015).

6.2.4.2.1 Parada Temprana

La idea general detrás de este método es que mientras el entrenamiento progresa la red utiliza más y más sus pesos, hasta que todos sus pesos son utilizados y alcanza un mínimo en la hipersuperficie del error. Si se incrementan el número de iteraciones en el entrenamiento, en realidad se está incrementando la complejidad de la red resultante.

Si el entrenamiento se lo detiene antes de alcanzar el mínimo, la red efectivamente utilizará menos parámetros y será menos probable que sobreajuste. (Hagan M., 2015).

Validación Cruzada

Para utilizar efectivamente la parada temprana, se necesita saber cuándo parar el entrenamiento. La herramienta estadística denominada ***Validación Cruzada*** provee una guía para este principio. Primero los datos disponibles son aleatoriamente divididos en datos de entrenamiento y datos de prueba. Los datos de entrenamiento son además divididos en: datos de estimación (utilizados para seleccionar el modelo) y datos de validación (utilizados para validar o probar el modelo). Lo importante es validar el modelo con datos diferentes a los utilizados para estimar el modelo. (Haykin, 1999).

Como se puede ver en la figura 6.42 (izquierda) tras una fase inicial, en la que pueden existir oscilaciones en el valor del error, el error de entrenamiento tiende a disminuir monótonamente, mientras que el error de generalización a partir de cierto punto comienza a incrementarse, lo que indicia una degradación progresiva del entrenamiento. (Haykin, 1999).

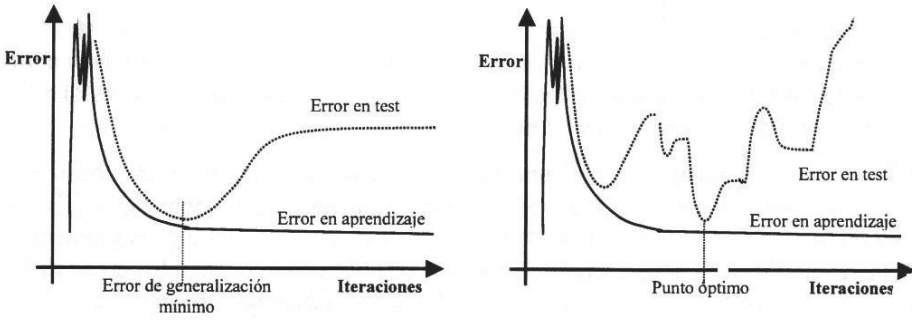


Figura 6.42 Parada Temprana

((Del Brío, 2007) Pág. 72)

Lo que pasa es lo siguiente: Al inicio la red se adapta progresivamente a los datos de entrenamiento, acomodándose al problema y mejorando la generalización. En un momento dado la red se ajusta demasiado a los datos empleados en el entrenamiento, aprendiendo incluso el ruido en ellos presente, por lo que crece el

error de generalización. En ese momento la red no ajusta correctamente la relación entrada – salida (mapping), sino que simplemente está memorizando los datos del entrenamiento, lo que técnicamente se denomina sobreajuste (o sobre entrenamiento).

La situación descrita ha sido idealizada; una situación más realista se muestra en la figura 6.42 derecha, en realidad pueden presentarse varios mínimos para el conjunto de prueba o test, debiendo detener el entrenamiento en el punto que produzca un mínimo error de generalización y no en el primer mínimo que aparezca. (Del Brío, 2007).

En la figura 6.43 se ilustra este proceso. En la parte izquierda, se muestra la respuesta de la red cuando se ha detenido el entrenamiento en el punto marcado como error de generalización mínimo en la fig. 6.42. A la derecha se muestra el gráfico cuando el entrenamiento continua, hasta donde error de validación se ha incrementado y la red sobreajusta.

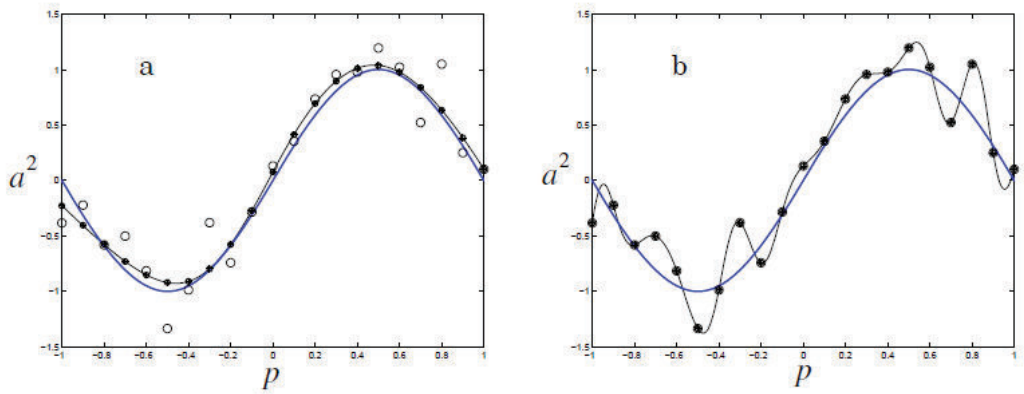


Figura 6.43 Ejemplos de Parada Temprada

((Hagan M., 2015) Pág. 13-7)

El concepto de parada temprana es simple, pero hay un par de cuestiones prácticas que se deben mencionar: *La Primera*, los datos de validación deben ser elegidos da tal manera que representen todas las situaciones para las cuales la red se utilizará. Esto se debe cumplir también para el conjunto de estimación y de prueba. Los datos dependiendo de su aplicación pueden ser divididos aproximadamente como: 70% para estimación, 15% para validación y 15% para prueba.

La segunda, en la parada temprana se debe utilizar un método de entrenamiento relativamente lento. Durante el entrenamiento la red utiliza más y más los parámetros disponibles (como se explicó antes). Si el método de entrenamiento es muy rápido, este puede saltarse el punto en el que el error de validación es minimizado.

6.2.4.2.2 Regularizacion

El segundo método para mejorar la generalización de la red se denomina ***regularización***. En este método se modifica el índice de desempeño (suma de los cuadrados de los errores) Ec. (6.96) para incluir un término que penaliza la complejidad de la red. Esta técnica conocida como la teoría de regularización de Tikhonov aumenta un término de penalidad o regularización, término que involucra la derivada de la función (en este caso la red neuronal), que obliga a la función resultante a ser más suave. Bajo ciertas condiciones, este término de regularización puede escribirse como la suma de los cuadrados de los pesos de la red,

$$F(\mathbf{x}) = \beta E_D + \alpha E_W = \beta \sum_{q=1}^{Q} (\mathbf{t_q} - \mathbf{a_q})^T (\mathbf{t_q} - \mathbf{a_q}) + \alpha \sum_{i=1}^{n} x_i{}^2 \qquad (6.97)$$

La relación $\frac{\alpha}{\beta}$ controla la complejidad de la red. Mientras más grande es esta relación la respuesta de la red es más suave.

Pero la pregunta sería ¿Qué tiene ver el término de regularización con la reducción del número de neuronas?

Para responder esta pregunta se considerará la red de la figura 6.22. Se debe recordar que cuando se incrementa el valor de uno de los pesos se incrementa la pendiente de la función.

Se puede ver este efecto en la figura 6.44, donde se cambia el peso $w_{1,1}^2$ desde 0 hasta 2. Cuando los pesos son grandes, la función creada por la red tiene pendientes grandes y por lo tanto es más probable que sobre ajuste los datos de entrenamiento. Si se limita los pesos a valores pequeños, la función creará una interpolación más suave a través de los datos de entrenamiento, tal como si la red tuviera un menor número de neuronas.

La clave para el éxito del método de regularización es escoger correctamente la relación $\frac{\alpha}{\beta}$. La figura 6.45 ilustra el efecto de cambiar esta relación. Se muestra una red 1-20-1 entrenada con 21 ejemplos de una onda seno.

En la figura, la línea azul representa la función verdadera, los círculos en blanco representan datos con ruido. La curva en negro representa la respuesta de la red entrenada y los círculos llenos con cruces representan la respuesta de la red en los

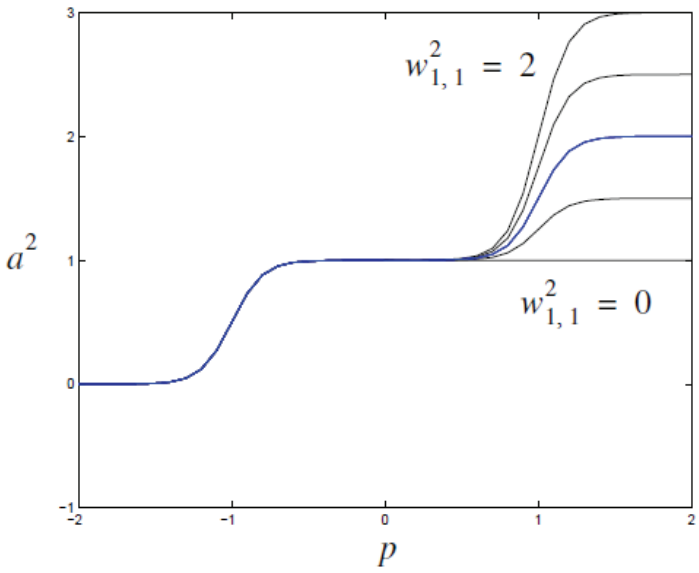


Figura 6.44 Efecto de los Pesos en la Respuesta de la Red

((Hagan M., 2015) Pág. 13-9)

puntos de entrenamiento. En esta figura se puede notar que la relación $\frac{\alpha}{\beta} = 0.01$ produce el mejor ajuste a la función verdadera. Para relaciones mayores a esta, la respuesta de la red es muy suave, y para relaciones más pequeñas a esta, la red sobreajusta. (Hagan M., 2015).

Existen varias técnicas para determinar los valores de los parámetros de regularización. Un enfoque es usar los datos de validación, tal como se describió en la sección de Parada Temprana, los parámetros de regularización se ajustan para minimizar el error al cuadrado en el conjunto de validación.

Otro enfoque es utilizar la regularización Bayesiana, en la cual se determinan automáticamente los parámetros de regularización. El lector puede profundizar este enfoque en la referencia (Hagan M., 2015)pp:(13-10 a 13-19).

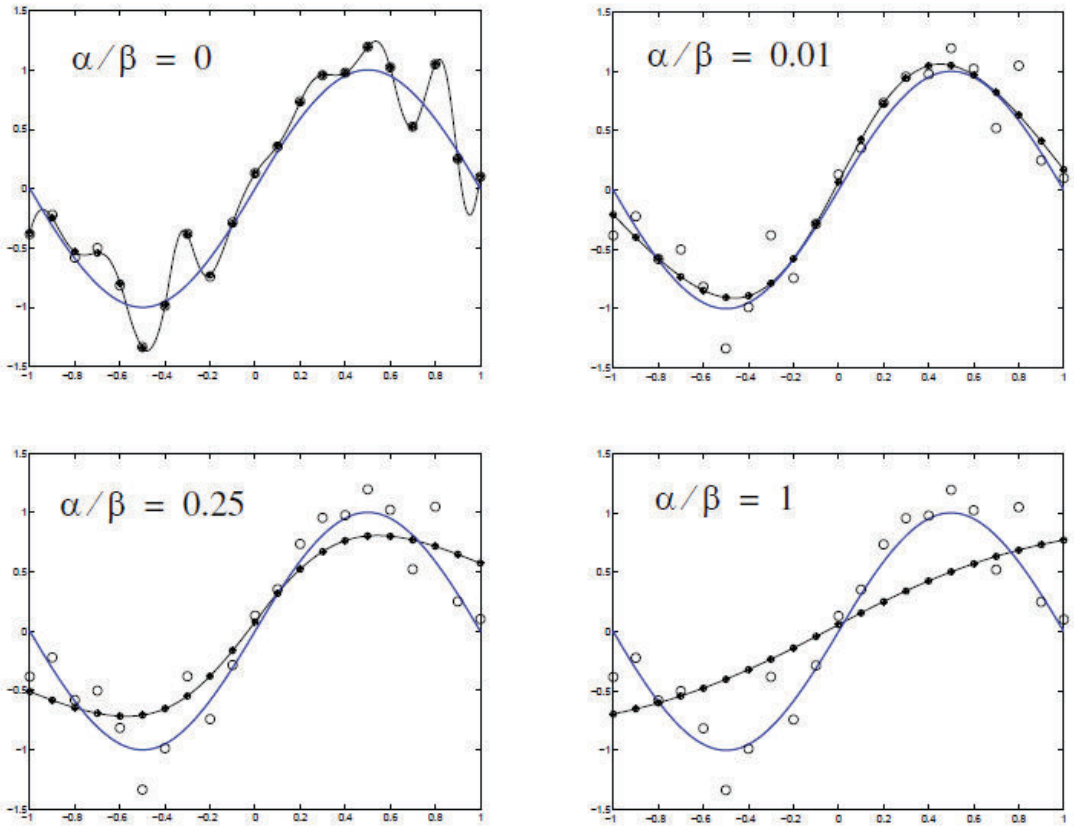


Figura 6.45 Efecto de la Relación de Regularización $\frac{\alpha}{\beta}$

((Hagan M., 2015) Pág. 13-10)

6.2.4.2.2.1 Regularización Bayesiana

Antes de analizar la regularización Bayesiana, será útil recordar el ***Teorema de Bayes:***

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)} \tag{6.98}$$

Donde:

$P(A)$ se denomina probabilidad a priori. Nos dice que se conoce acerca de A antes de conocer el resultado de B.

$P(A/B)$, se conoce como probabilidad posterior. Nos dice que se conoce acerca de A después de conocer acerca de B.

$P(B/A)$ es la probabilidad de B dado A.

$P(B)$ es la probabilidad marginal del evento B, y actúa como un factor de normalización en el teorema de Bayes. (Hagan M., 2015).

En la regularización Bayesiana se aplica el análisis Bayesiano al entrenamiento de redes multicapa. La ventaja de los métodos Bayesianos es que se puede tomar en cuenta un conocimiento a priori a través de la selección de una probabilidad a priori. Para el entrenamiento de las redes neuronales, se asume de manera a priori que la función que estamos aproximando está suavizada. Esto significa que los pesos no pueden ser muy grandes, como se demostró en la figura 6.44. El truco será incorporar este conocimiento a priori en la selección apropiada de la probabilidad a priori. (Hagan M., 2015).

Existen varias maneras de seleccionar automáticamente los parámetros para la regularización, se analizará en esta sección el método desarrollado por David MacKay. Este enfoque coloca al entrenamiento de las redes neuronales dentro de un marco estadístico Bayesiano. Este marco es útil para varios aspectos del entrenamiento, adicionalmente para la selección de los parámetros de regularización. Existen dos niveles en el análisis Bayesiano.

Nivel I del Análisis Bayesiano

En enfoque Bayesiano se inicia con la suposición de que los pesos de la red son variables aleatorias. Entonces se puede elegir los pesos de tal manera que maximicen la probabilidad condicional de los pesos dado los datos. El teorema de Bayes es útil para encontrar esta función de probbilidad:

$$P(\mathbf{x}/D,\alpha,\beta,M) = \frac{P(D/\mathbf{x},\beta,M)P(\mathbf{x}/\alpha,M)}{P(D/\alpha,\beta,M)} \qquad (6.99)$$

Donde $\mathbf{x}$ es el vector que contiene todos los pesos y bías de la red, D representa el conjunto de datos para entrenamiento, los parámetros, $\alpha\, y\, \beta$ son parámetros asociados con las funciones de densidad $P(D/\mathbf{x},\beta,M)\ y\ P(\mathbf{x}/\alpha,M)$ y M es el modelo seleccionado – arquitectura de la red seleccionada (es decir cuántas capas y cuántas neuronas en cada capa).

Se analizarán cada uno de los términos de la Ec. (6.99).

Así $P(D/\mathbf{x},\beta,M)$ es la densidad de probabilidad de los datos, dado un cierto conjunto de pesos $\mathbf{x}$, el parámetro β y el modelo seleccionado M. Si se asume que

la fuente de ruido en la Ec. (6.95) es independiente y tiene una distribución Gaussiana, entonces

$$P(D/\mathbf{x},\beta,M) = \frac{1}{Z_D(\beta)} e^{(-\beta E_D)}, \quad (6.100)$$

Donde $\beta = {}^{1}/_{2{\sigma_e}^2}$, ${\sigma_e}^2$ es la varianza de cada elemento de ε_q, E_D es el error al cuadrado (como se definió en la Ec. (6.96)) y

$$Z_D(\beta) = (2\pi{\sigma_e}^2)^{N/2} = (\pi/\beta)^{N/2}, \quad (6.101)$$

Donde N es QxS^M, como en la Ec. (6.79).

La ecuación (6.100) se denomina ***función de Verosimilitud***. Es una función de los pesos de la red $\mathbf{x}$, y describe que tan probable es que un conjunto de datos ocurran, dado un conjunto específico de pesos. El método del ***máximo de verosimilitud*** selecciona los pesos de tal manera que maximice la función de verosimilitud, el cual en el caso Gaussiano es lo mismo que minimizar el cuadrado de los errores E_D. Por lo tanto el índice de desempeño estándar (suma de los errores al cuadrado) se puede derivar estadísticamente con la asunción de ruido Gaussiano en los datos de entrenamiento y la selección de los pesos estándar es el estimado del máximo de verosimilitud.

El segundo término de la Ec.(6.99) $P(\mathbf{x}/\alpha,M)$ se le conoce como ***densidad a priori***. El análisis Bayesiano permite incorporar un conocimiento a priori a través de la densidad a priori. Por ejemplo, si se asume que los pesos son valores pequeños centrados alrededor de cero, se puede seleccionar una densidad a priori Gaussiana de media cero:

$$P(\mathbf{x}/\alpha,M) = \frac{1}{Z_W(\alpha)} e^{(-\alpha E_W)}, \quad (6.102)$$

Donde $\alpha = {}^{1}/_{2{\sigma_W}^2}$, ${\sigma_W}^2$ es la varianza de cada uno de los pesos, E_W es la suma de los cuadrados de los pesos (como se definió en la Ec. (6.97)) y

$$Z_W(\alpha) = (2\pi{\sigma_W}^2)^{n/2} = (\pi/\alpha)^{n/2}, \quad (6.103)$$

Donde n es el número de pesos y bías de la red, como en la Ec. (6.80).

El término final del lado derecho de la Ec.(6.99) es $P(D/\alpha,\beta,M)$. Es llamado ***la evidencia*** y es un término de normalización que no es función de $\mathbf{x}$. Si nuestro objetivo sería encontrar los pesos $\mathbf{x}$ que maximizan la *densidad posterior*

$P(\mathbf{x}/D,\alpha,\beta,M)$, entonces no se necesita estar preocupado de $P(D/\alpha,\beta,M)$. (Sin embargo si es importante cuando se necesita calcular $\alpha\ y\ \beta$). (Hagan M., 2015).

Si se asume distribuciones Gaussianas se puede escribir la Ec.(6.99) correspondiente a la densidad posterior, de la siguiente forma:

$$P(\mathbf{x}/D,\alpha,\beta,M) = \frac{\frac{1}{Z_W(\alpha)}\frac{1}{Z_D(\beta)}e^{(-(\beta E_D+\alpha E_W))}}{Factor\ de\ Normalización}.$$

$$P(\mathbf{x}/D,\alpha,\beta,M) = \frac{1}{Z_F(\alpha,\beta)}e^{(-F(\mathbf{x}))} \qquad (6.104)$$

Donde $Z_F(\alpha,\beta)$ es una función de $\alpha\ y\ \beta$ (pero no función de $\mathbf{x}$) y $F(\mathbf{x})$ es el índice de desempeño regularizado, el cual fue definido en la Ec. (6.97). Para encontrar los valores más probables para los pesos, se debe maximizar la densidad posterior $P(\mathbf{x}/D,\alpha,\beta,M)$, que es equivalente a minimizar el índice de desempeño regularizado $F(\mathbf{x}) = \beta E_D + \alpha E_W$.

Por lo tanto el índice de desempeño regularizado puede ser obtenido utilizando estadística Bayesiana, con las suposiciones de ruido Gaussiano en los datos de entrenamiento y densidad a priori Gaussiana para los pesos de la red. Se identificarán los pesos de la red para maximizar la densidad posterior como $\mathbf{x}^{MP}$, *más probable* (Del inglés *more probable*). Este debe ser contrastado con los pesos que maximizan la función de verosimilitud: $\mathbf{x}^{ML}$.

Se debe notar que este enfoque estadístico provee un significado físico de los parámetros $\alpha\ y\ \beta$. El parámetro β es inversamente proporcional a la varianza del ruido ϵ_q. Por lo tanto si la varianza del ruido es grande, β será pequeño y la relación de regularización α/β será grande. Esto forzará a que los pesos resultantes sean pequeños y la función de red este suavizada (Fig. 6.45). Mientras más grande sea el ruido, la función de red estará más suavizada, con el fin de promediar los efectos del ruido.

El parámetro α es inversamente proporcional a la varianza en la distribución a priori de los pesos de la red. Si esta varianza es grande, esto significa que se tiene muy poca certeza acerca de los valores de los pesos de la red, y por lo tanto estos serán muy grandes. El parámetro α entonces será muy pequeño y la relación de regularización α/β también será pequeña. Esto permitirá que los pesos de la red sean grandes y la función de red podrá tener más variación (ver figura 6.45).

Mientras más grande es la varianza en la densidad a priori de los pesos de la red, la función de red podrá tener más variación. (Hagan M., 2015).

Nivel II del Análisis Bayesiano

Una vez que se ha conocido el significado físico de los parámetros $\alpha\ y\ \beta$, ahora es necesario conocer una forma de estimar estos parámetros a partir de los datos. Para esto es necesario llevar el análisis Bayesiano a otro nivel. Si se quiere estimar $\alpha\ y\ \beta$ utilizando el análisis Bayesiano, se necesita la densidad de probabilidad $P(\alpha, \beta/D, M)$. Utilizando el teorema de Bayes se puede escribir como:

$$P(\alpha, \beta/D, M) = \frac{P(D/\alpha,\beta,M)P(\alpha,\beta/M)}{P(D/M)} \tag{6.105}$$

Esta ecuación tiene el mismo formato que la Ec. 6.99, con la función de verosimilitud y la densidad a priori en el numerador del lado derecho. Si se asume una densidad a priori uniforme (constante) $P(\alpha, \beta/M)$ para los parámetros de regularización $\alpha\ y\ \beta$, entonces para maximizar la densidad posterior es necesario maximizar la función de verosimilitud $P(D/\alpha, \beta, M)$. Sin embargo, note que esta función de verosimilitud es el factor de normalización (evidencia) de la Ec. (6.99). Ya que se asumió que todas las probabilidades tienen forma Gaussiana, se conoce la forma de densidad posterior de la Ec.(6.99). Esta se muestra en la Ec.(6.104). Ahora se resolverá la Ec.(6.99) para encontrar el factor de normalización (evidencia).

$$P(D/\alpha, \beta, M) = \frac{P(D/\mathbf{x},\beta,M)P(\mathbf{x}/\alpha,M)}{P(\mathbf{x}/D,\alpha,\beta,M)}$$

$$P(D/\alpha, \beta, M) = \frac{\left[\frac{1}{Z_D(\beta)}e^{(-\beta E_D)}\right]\left[\frac{1}{Z_W(\alpha)}e^{(-\alpha E_W)}\right]}{\frac{1}{Z_F(\alpha,\beta)}e^{(-F(\mathbf{x}))}}$$

$$P(D/\alpha, \beta, M) = \frac{Z_F(\alpha,\beta)}{Z_D(\beta)Z_W(\alpha)}\frac{e^{(-(\beta E_D+\alpha E_W))}}{e^{(-F(\mathbf{x}))}} = \frac{Z_F(\alpha,\beta)}{Z_D(\beta)Z_W(\alpha)} \tag{6.106}$$

Note que las constantes $Z_D(\beta) y\ Z_W(\alpha)$ se conocen de las Ecs. (6.101) y (6.103). La única parte desconocida es $Z_F(\alpha, \beta)$. Sin embargo se la puede estimar expandiendo en series de Taylor.

Ya que la función objetivo tiene forma cuadrática en una pequeña área alrededor del punto mínimo, se expandirá $F(\mathbf{x})$ en una serie de Taylor de segundo orden alrededor del punto mínimo, $\mathbf{x}^{MP}$, donde el gradiente es cero:

$$F(\mathbf{x}) \approx F(\mathbf{x}^{MP}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{MP})^T\mathbf{H}^{MP}(\mathbf{x} - \mathbf{x}^{MP}), \tag{6.107}$$

Donde $\mathbf{H} = \beta\nabla^2 E_D + \alpha\nabla^2 E_W$ es la matriz Hessiana de $F(\mathbf{x})$ y $\mathbf{H}^{MP}$ es el Hessiano evaluado en $\mathbf{x}^{MP}$. Se sustituye esta aproximación en la densidad posterior de la Ec. (6.104):

$$P(\mathbf{x}/D,\alpha,\beta,M) \approx \frac{1}{Z_F} e^{\left[-F(\mathbf{x}^{MP})-\frac{1}{2}(\mathbf{x}-\mathbf{x}^{MP})^T\mathbf{H}^{MP}(\mathbf{x}-\mathbf{x}^{MP})\right]}, \quad (6.108)$$

$$P(\mathbf{x}/D,\alpha,\beta,M) \approx \frac{1}{Z_F} e^{(-F(\mathbf{x}^{MP}))} e^{\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}^{MP})^T\mathbf{H}^{MP}(\mathbf{x}-\mathbf{x}^{MP})\right]}, \quad (6.109)$$

La forma estándar de la densidad Gaussiana es:

$$P(x) = \frac{1}{\sqrt{(2\pi)^n \left|(\mathbf{H}^{MP})^{-1}\right|}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}^{MP})^T\mathbf{H}^{MP}(\mathbf{x}-\mathbf{x}^{MP})} \quad (6.110)$$

Igualando (6.109) con (6.110), se puede obtener $Z_F(\alpha,\beta)$:

$$Z_F(\alpha,\beta) \approx (2\pi)^{\frac{n}{2}} (\det(((\mathbf{H}^{MP})^{-1}))^{1/2} e^{(-F(\mathbf{x}^{MP}))} . \quad (6.111)$$

Reemplazando $Z_F(\alpha,\beta)$ en la Ec.(6.106), se puede obtener los valores óptimos para $\alpha\ y\ \beta$ en el punto mínimo. Esto se logra tomando la derivada respecto a cada uno de ellos del logaritmo de la Ec. (6.106) e igualándola a cero. Se obtiene lo siguiente:

$$\alpha^{MP} = \frac{\gamma}{2E_W(\mathbf{x}^{MP})}\ y\ \beta^{MP} = \frac{N-\gamma}{2E_D(\mathbf{x}^{MP})} , \quad (6.112)$$

Donde $\gamma = n - 2\alpha^{MP} tr(\mathbf{H}^{MP})^{-1}$ se denomina ***número efectivo de parámetros*** y n es el número total de parámetros en la red. El término γ es una medida de cuántos parámetros (pesos y bías) en la red neuronal están siendo usados en reducir el error. El rango va desde cero hasta n.

Algoritmo de Regularización Bayesiana

La optimización Bayesiana de los parámetros de regularización requiere del cálculo de la matriz Hessiana de $F(\mathbf{x})$ en el punto mínimo $\mathbf{x}^{MP}$. Se propone el uso de la aproximación de Gauss – Newton para dicho cálculo, el cual está fácilmente disponible en el algoritmo de optimización de Levenberg-Marquardt que es usado para localizar el punto mínimo (Ver Ec. (6.75)). (Hagan M., 2015).

A continuación se muestran los pasos para la optimización Bayesiana de los parámetros de regularización, con la aproximación de Gauss-Newton a la matriz Hessiana:

0 Inicializar α, β y los pesos. Los pesos se inicializan aleatoriamente y entonces se calculan E_D y E_W. Asigne $\gamma = n$, y calcule $\alpha\ y\ \beta$ utilizando (6.112).

1 Aplique un paso del algoritmo de Levenberg-Marquardt para minimizar la función objetivo $F(\mathbf{x}) = \beta E_D + \alpha E_W$.

2 Calcule el número efectivo de parámetros $\gamma = n - 2\alpha tr(\mathbf{H})^{-1}$, utilice la aproximación de Gauss-Newton para la matriz Hessiana disponible en el algoritmo de entrenamiento de Levenberg-Marquardt:
$\mathbf{H} = \nabla^2 F(\mathbf{x}) \approx 2\beta\ \mathbf{J}^T\mathbf{J} + 2\alpha\mathbf{I_n}$, donde $\mathbf{J}$ es el Jacobiano de los errores en los datos de entrenamiento (Ver Ec. (6.82)).

3 Calcule los nuevos estimados de los parámetros de regularización $\alpha = \frac{\gamma}{2E_W(\mathbf{x})}\ y\ \beta = \frac{N-\gamma}{2E_D(\mathbf{x})}$.

4 Repita los pasos 1 a 3 hasta alcanzar la convergencia.

Se debe tomar en cuenta que en cada re-estimación de los parámetros $\alpha\ y\ \beta$ la función objetivo $F(\mathbf{x})$ cambia, por lo tanto, el punto mínimo está en movimiento. Si atraviesa la hipersuperficie de desempeño generalmente se mueve hacia el siguiente punto mínimo, entonces la nueva estimación de los parámetros de regularización será más precisa. Eventualmente, si la precisión es lo suficientemente buena para que la función objetivo no cambie significativamente en varias iteraciones, significa que se alcanzó la convergencia.

Cuando se usa la aproximación de Gauss-Newton para la regularización Bayesiana GNBR (Del Inglés Gauss-Newton Aproximation to Bayesian regularization), mejores resultados se obtienen si los datos de entrenamiento se los normaliza en un rango entre $[-1,1]$ (o alguna región similar). (Hagan M., 2015).

En la figura 6.46 se pueden ver los resultados de una red $1 - 20 - 1$ entrenada con GNBR para los mismos datos presentados en las figuras 6.43 y 6.45. La red ha ajustado la función, sin sobre entrenamiento hacia el ruido. El ajuste se ve muy similar al de la figura 6.45, con la relación de regularización de $\frac{\alpha}{\beta} = 0.01$. En efecto al final del entrenamiento con GNBR la relación de regularización para este ejemplo fue de $\frac{\alpha}{\beta} = 0.0137$. (Hagan M., 2015).

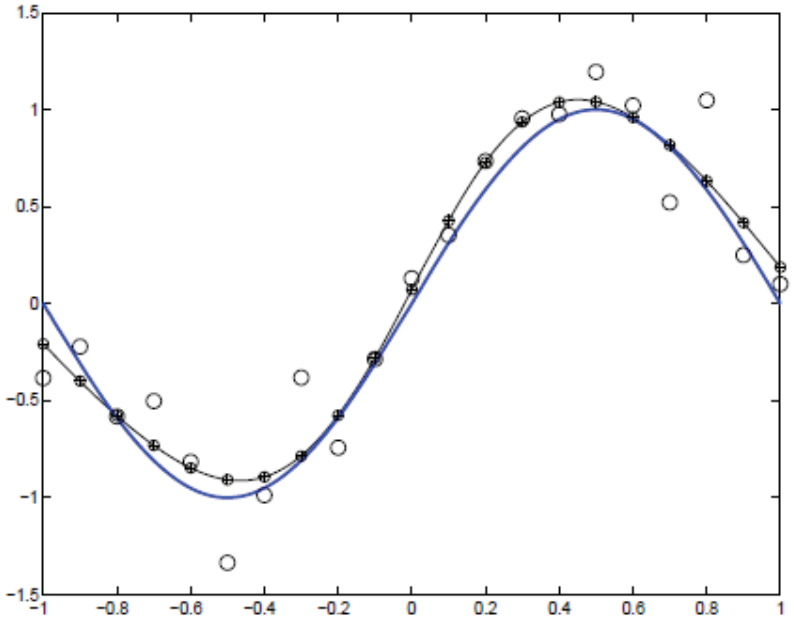


Figura 6.46 Ajuste con Regularización Bayesiana

((Hagan M., 2015) Pág. 13-18)

El número efectivo de parámetros final fue de $\gamma = 5.2$. Este valor está lejos del total de pesos y bías de la red 61.

El hecho de que el número efectivo de parámetros, para este ejemplo, sea mucho menor que el número total de parámetros (6 contra 61) significa que se podría utilizar una red más pequeña para ajustar estos datos. Existen dos desventajas al utilizar una red muy grande: 1) Puede sobreajustar los datos y 2) Se requiere de un mayor tiempo de cálculo para obtener la salida de la red. Se puede superar la primera desventaja entrando la red con GNBR, aunque la red tenga 61 parámetros. La segunda desventaja es solamente importante si el tiempo de cálculo para obtener la respuesta es crítico para la aplicación en la que se está utilizando la red neuronal (Ejemplo implementación de redes neuronales en circuitos integrados electrónicos), en estos casos se debe considerar entrenar una red más pequeña.

Por otro lado cuando el número efectivo de parámetros es cercano al número total de parámetros, esto significa que la red no es lo suficientemente grande para ajustar los datos. En este caso, se debe incrementar el tamaño de la red y entrenarla nuevamente. (Hagan M., 2015).

6.2.5 REDES DINÁMICAS Y PRONÓSTICOS

Las redes neuronales se clasifican dentro de dos categorías estáticas y dinámicas. Hasta ahora se han analizado redes estáticas. Esto significa que las salidas se calculan directamente desde las entradas a través de sus conexiones hasta la salida. En las redes dinámicas, la salida no depende solamente de la entrada actual a la red, sino también de entradas previas, salidas o estados de la red. Esta redes tienen conexiones recurrentes (realimentación), lo que significa que la salida actual es una función de la salida retrasada una o varios períodos.

En cuanto al entrenamiento de este tipo de redes la diferencia radica en la forma como se calculan los gradientes o jacobianos. (Hagan M., 2015).

Las redes dinámicas son redes que contienen retardos (o integradores para redes continuas en el tiempo) y operan con una secuencia de entradas. (Es decir el orden de las entradas es importante para la operación de la red). Las redes dinámicas tienen memoria. Su respuesta a un tiempo dado no dependerá solamente de la entrada actual de la red, sino de la historia de la secuencia de entrada.

Debido a que las redes dinámicas tienen memoria, ellas pueden ser entrenadas para aprender secuencias o patrones variables en el tiempo. En lugar de aproximar funciones, como el perceptrón multicapa MLP, una red dinámica puede aproximar sistemas dinámicos. Estas tienen aplicaciones en diversas áreas como control de sistemas dinámicos, predicción en mercados financieros, detección de fase en sistemas de potencia, detección de fallas, reconocimiento de voz y hasta la predicción de la estructura de las proteínas en genética.

Las redes dinámicas pueden ser entrenadas utilizando los métodos de optimización estándar que se ha visto. Sin embargo los gradientes y jacobianos que se requieren para estos métodos no pueden ser calculados utilizando el algoritmo de retropropagación estándar, se requieren algoritmos de retropropagación dinámica.

Existen dos enfoques generales (con muchas variaciones) para calcular el gradiente y el jacobiano en redes dinámicas: Retropropagación a través del tiempo (BPTT del inglés Backpropagation–through-time) y Aprendizaje recurrente en tiempo real (RTRL del inglés real-time recurrent learning).

En el algoritmo BPTT, la respuesta de la red es calculada para todos los puntos en el tiempo y el gradiente es calculado iniciando desde el último punto en el tiempo y trabajando hacia atrás en el tiempo. Este algoritmo es eficiente para calcular el gradiente, pero es difícil implementar en línea, ya que trabaja hacia atrás en el tiempo desde el último paso.

En el algoritmo RTRL, el gradiente puede ser calculado al mismo tiempo que la respuesta de la red, ya que es calculado iniciando en el primer punto (en el tiempo) y entonces trabajando hacia adelante a través del tiempo. RTRL requiere mayor tiempo de máquina para calcular el gradiente que el BPTT, pero permite la implementación en línea. Para el cálculo del jacobiano el algoritmo RTRL es generalmente más eficiente que el BPTT. (Hagan M., 2015).

Adicionalmente la hipersuperficie del error en las redes dinámicas es mucho más compleja que la de las redes estáticas. El entrenamiento es más probable que se vea atrapado en mínimos locales. Se sugiere entrenar la red varias veces para conseguir el resultado óptimo. (Beale M., 2015).

6.2.5.1 Redes Dinámicas Digitales Multicapa

En esta sección se introducirá el marco adecuado para representar las redes neuronales dinámicas. A este marco se lo denominará Redes Dinámicas Digitales Multicapa (LDDN del inglés Layered Digital Dynamic Networks). No es más que una extensión de la notación que se ha utilizado para representar las redes multicapa estáticas. Con esta nueva notación, se representará convenientemente redes con conexiones recurrentes múltiples (realimentación) y líneas de retardo (TDL del inglés tapped delay lines). (Hagan M., 2015).

En la figura 6.47 se muestra un ejemplo para poder introducir la notación LDNN.

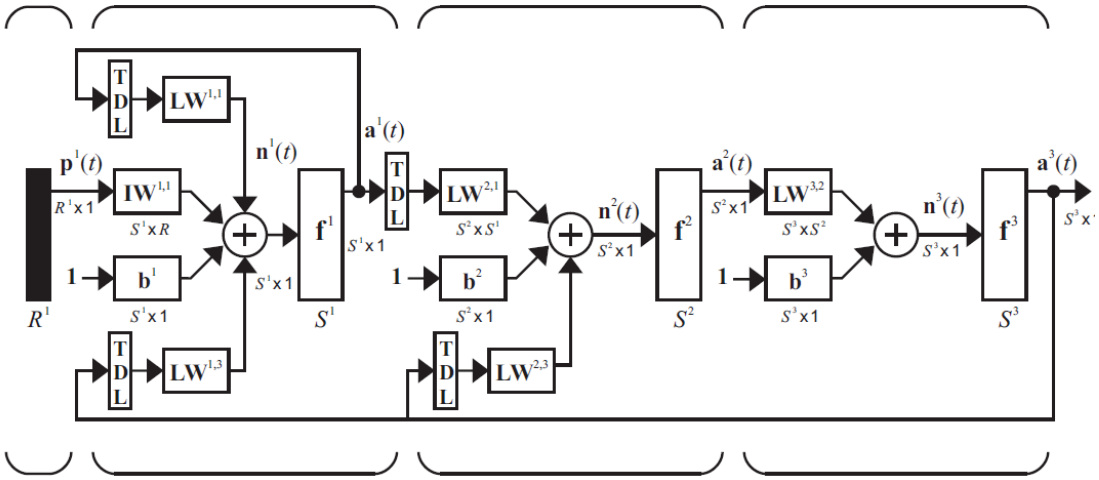


Figura 6.47 Ejemplo Red Neuronal Dinámica

((Hagan M., 2015) Pág. 14-3)

La ecuación general para el cálculo de la entrada neta $\mathbf{n}^m(t)$ para la capa m de la LDDN es:

$$\mathbf{n}^m(t) = \sum_{l\in L_m^f}\sum_{d\in DL_{m,l}} \mathbf{LW}^{m,l}(d)\mathbf{a}^l(t-d) + \sum_{l\in I_m}\sum_{d\in DI_{m,l}} \mathbf{IW}^{m,l}(d)\mathbf{p}^l(t-d) + \mathbf{b}^m \quad (6.113)$$

Donde $\mathbf{p}^l(t)$ es el $l-ésimo$ vector de entrada de la red al tiempo t, $\mathbf{IW}^{m,l}$ es el *peso de entrada* entre la entrada l y la capa m, $\mathbf{LW}^{m,l}$ es el *peso entre capas*, entre la capa l y la capa m, $\mathbf{b}^m$ es el vector bías para la capa m, $DL_{m,l}$ es el conjunto de todos los retrasos en la línea de retraso entre la capa l y la capa m, $DI_{m,l}$ es el conjunto de todos los retrasos en la línea de retraso entre la entrada l y la capa m, I_m es el conjunto de índices de los vectores de entrada que conectan a la capa m, y L_m^f es el conjunto de índices de las capas que se conectan directamente hacia adelante con la capa m. La salida de la capa m se calcula como

$$\mathbf{a}^m(t) = \mathbf{f}^m(\mathbf{n}^m(t)) . \quad (6.114)$$

La LDDN tiene varias capas conectadas a la capa m. Algunas de las conexiones pueden ser recurrentes a través de las líneas de retardo. Este tipo de redes también tienen múltiples vectores de entrada y estos vectores de entrada pueden ser conectados a cualquier capa de la red, en las redes estáticas, se asume que un solo vector de entrada está conectado a la capa 1.

En las redes estáticas multicapa, las capas están conectadas unas con otras siguiendo un orden numérico. Es decir, la capa 1 está conectada con la capa 2, la que a su vez está conectada con la capa 3, etc. En una LDDN, cualquier capa puede estar conectada con cualquier otra capa, o consigo misma. Pero para poder utilizar la ecuación (6.113) es necesario calcular las salidas en un orden específico. Este orden en el cual la salida de cada capa debe ser calculada para obtener la salida correcta de la red se llama *orden de simulación*. (Este orden no es único, puede haber varios órdenes de simulación válidos.) Para calcular el gradiente se debe retropropagar las derivadas, entonces se debe seguir un orden inverso, el cual es llamado *orden de retropropagación*.

Al igual que las redes estáticas, la unidad fundamental de las LDDN es la capa. Cada capa en las LDDN está formada por cinco componentes:

1.- Un conjunto de matrices de pesos que vienen dentro de la capa (las cuales pueden conectarse desde otras capas o desde entradas externas),

2.- líneas de retardo (representadas por ($DL_{m,l}\ o\ DI_{m,l}$) que aparecen a la entrada de un conjunto de matrices de pesos (Cualquier conjunto de matrices de pesos puede ser precedida por una línea de retardo (TDL). Así en la figura 6.47 la primera capa contiene los pesos $\mathbf{LW}^{1,3}(d)$ y su correspondiente TDL.),

3.- Un vector bías

4.- Un sumador y

5.- Una función de transferencia.

La salida de una LDDN es una función no solamente de los pesos, bías y entradas actuales de la red, sino también de algunas salidas de las capas retardadas. Por esta razón, no es simple calcular el gradiente de la salida de la red con respecto a los pesos y bías. Los pesos y bías tienen dos efectos diferentes en la salida de la red: El primero es un efecto directo, el mismo que puede ser calculado usando el algoritmo de retropropagación estándar visto anteriormente. El segundo es un efecto indirecto, ya que algunas entradas a la red y salidas previas, son también función de los pesos y bías. (Hagan M., 2015).

6.2.5.2 Pronósticos

La predicción cae dentro de las categorías del análisis de series de tiempo, identificación de sistemas, filtrado o modelación dinámica. La idea es predecir valores futuros de algunas series de tiempo.

Los pronósticos requieren el uso de redes neuronales dinámicas. La forma específica de la red dependerá de la aplicación en particular. La red más simple para predicción no lineal es la red neuronal enfocada con retardo en el tiempo FTDNN (Del inglés Focused time-delay neural network.) que se muestra en la figura 6.48 (Hagan M., 2015).

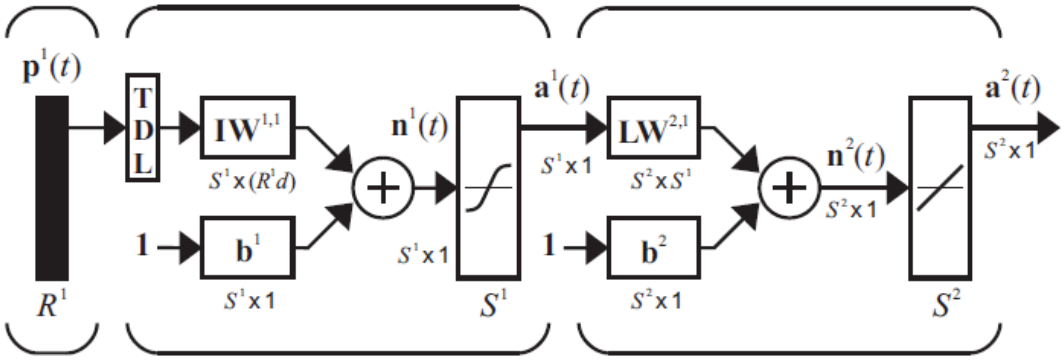


Figura 6.48 Red Neuronal FTDNN

((Hagan M., 2015) Pág. 17-10)

Esta red es parte de una clase más general de redes dinámicas, llamadas redes enfocadas, en las cuales la dinámica solamente aparece en la capa de entrada de una red perceptrón multicapa. Esta red tiene la ventaja que puede ser entrenada con algoritmos de retropropagación estáticos, ya que la línea de retardo a la entrada de la red puede ser reemplazada por un vector extendido con valores de la entrada con sus respectivos retardos. (Hagan M., 2015).

Esta red es muy adecuada para la predicción de series de tiempo. (Beale M., 2015).

Para problemas de modelamiento dinámico y control, la red NARX (Del inglés Non linear AutoRegressive model with eXogenous input) es muy popular. Esta red se muestra en la figura 6.49.

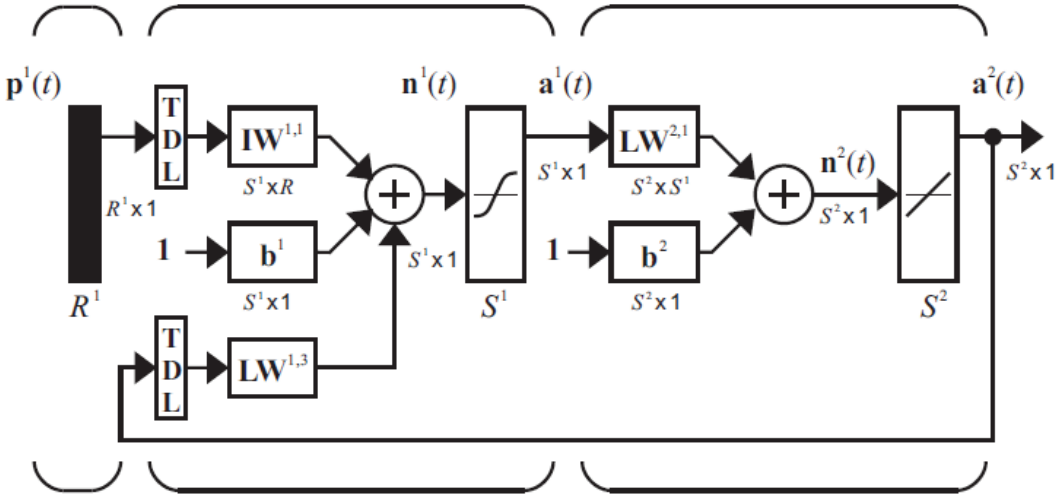


Figura 6.49 Red Neuronal NARX

((Hagan M., 2015) Pág. 17-11)

La señal de entrada podría representar, por ejemplo, el voltaje aplicado a un motor, y la salida podría representar la posición angular del brazo de un robot. Al igual que la FTDNN, la red NARX puede ser entrenada con retropropagación estática. Las dos líneas de retardo TDL, pueden ser reemplazadas con vectores extendidos de entradas y salidas deseadas con sus respectivos retardos. Se puede usar salidas deseadas, en lugar de realimentar la salida (lo cual requeriría de un entrenamiento de retropropagación dinámica), ya que la salida de la red debería coincidir con las salidas deseadas cuando el entrenamiento se haya completado. (Hagan M., 2015). Si se considera la salida de la red NARX como un estimado de la salida de un sistema dinámico no lineal que se está intentando modelar. La salida es realimentada hacia la entrada del perceptrón MLP como parte de la arquitectura de la red NARX estándar, como se muestra en la figura 6.50 izquierda, ya que la verdadera salida es disponible durante el entrenamiento de la red, se puede crear una arquitectura serie – paralelo como se indica en la figura 6.50 derecha, en la cual la verdadera salida se utiliza en lugar de realimentar la salida estimada. Esto tiene dos ventajas. *La primera*, es que la entrada al MLP es más precisa. *La segunda,* es que la red resultante es un MLP puro y se puede usar retropropagación estándar (estática) para su entrenamiento.

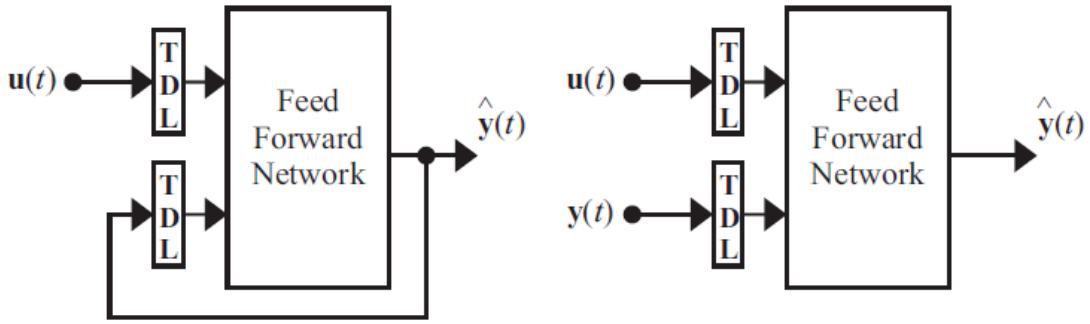


Figura 6.50 Red Neuronal NARX: Arquitectura Paralela y Serie - Paralelo

((Beale M., 2015) Pág. 4-23)

Si a la FTDNN de la figura 6.49 se le aplica como entrada la misma salida de la red $\mathbf{a}^2(t)$ a esta red neuronal resultante se la puede entrenar para predecir series de tiempo a partir de los valores pasados de la serie y se denomina red tipo NARnet (Del Inglés nonlinear autoregressive Network). (Beale M., 2015).

Existen dos conceptos importantes que se utilizan para analizar una red entrenada para predecir:

1.- Los errores de predicción no debe estar correlacionados en el tiempo y

2.- Los errores de predicción no debe estar correlacionados con la secuencia de entrada.

Si los errores de pronóstico estuvieran correlacionados en el tiempo, se podrían predecir los errores de pronóstico, por lo tanto mejorar la predicción original. También si los errores de pronóstico estuvieran correlacionados con la secuencia de entrada entonces se podría utilizar esta correlación para predecir los errores.

Para analizar la correlación de los errores de pronóstico en el tiempo, se puede usar la ***función de autocorrelación muestral***:

$$R_e(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} e(t)e(t+\tau)\,. \qquad (6.115)$$

Si los errores de pronóstico están no correlacionados (ruido blanco), entonces se podría esperar que $R_e(\tau)$ sea aproximadamente cero, excepto cuando $\tau = 0$. Para determinar si $R_e(\tau)$ es aproximadamente cero, se puede establecer un intervalo de confianza del 95% utilizando el rango

$$-\frac{2R_e(0)}{\sqrt{Q}} < R_e(\tau) < \frac{2R_e(0)}{\sqrt{Q}}\,. \qquad (6.116)$$

Se puede decir que $e(t)$ es ruido blanco, si $R_e(\tau)$ satisface la Ec. (6.116) para $\tau \neq 0$ este concepto se ilustra es las figuras 6.51 y 6.52. La figura 6.51 muestra la función de autocorrelación muestral de los errores de predicción de una red que no ha sido adecuadamente entrenada.

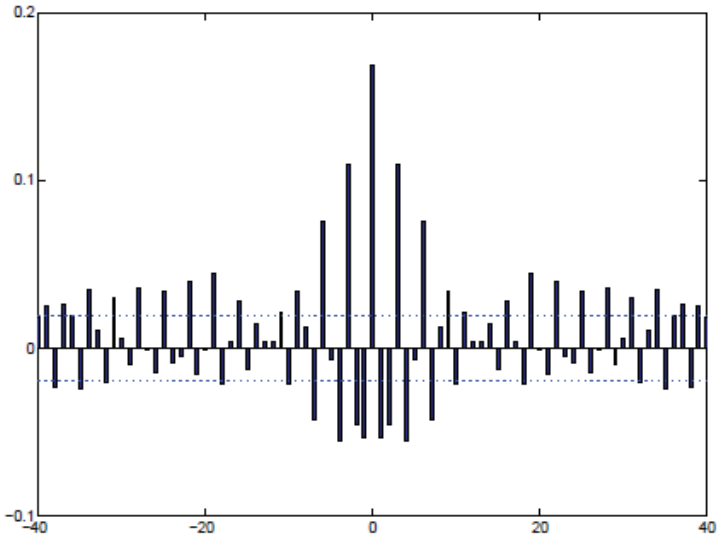


Figura 6.51 $R_e(\tau)$ para una red entrenada inadecuadamente

((Hagan M., 2015) Pág. 17-25)

Se puede notar que la función de autocorrelación no cae dentro de los límites definidos por la Ec. (6.116), que se indican con líneas horizontales azules entrecortadas.

La figura 6.52 muestra la función de autocorrelación de una red que ha sido correctamente entrenada. $R_e(\tau)$ está dentro los límites, excepto para $\tau = 0$.

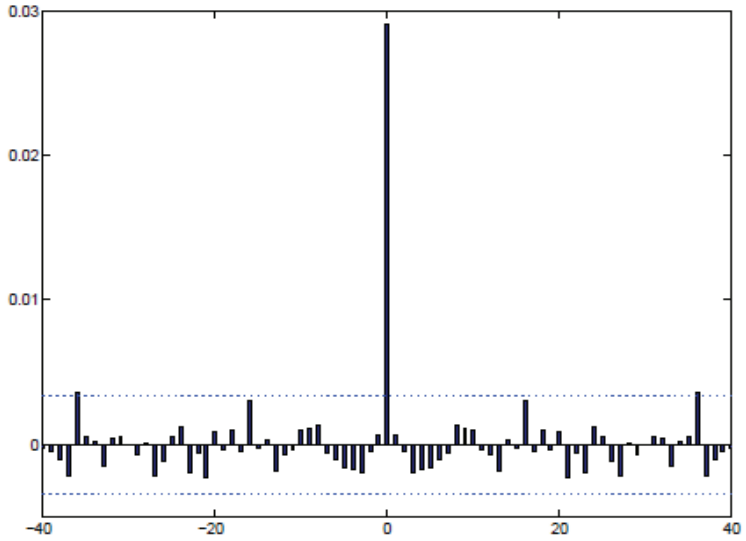


Figura 6.52 $R_e(\tau)$ para una red entrenada correctamente

((Hagan M., 2015) Pág. 17-25)

La correlación en los errores de predicción indica que el retardo en la red debería incrementarse.

Para chequear la correlación entre los errores de predicción y la secuencia de entrada, se puede utilizar la ***función muestral de correlación cruzada***:

$$R_{pe}(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} p(t)e(t+\tau)\,. \qquad (6.117)$$

Si no hay correlación entre los errores de predicción y la secuencia de entrada, se esperaría que $R_{pe}(\tau)$ sea aproximadamente cero para todo τ. Para determinar si $R_{pe}(\tau)$ es cercana a cero, se puede utilizar el intervalo con un 95% de confianza cuyos límites son:

$$-\frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}} < R_{pe}(\tau) < \frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}}. \qquad (6.118)$$

Este concepto se ilustra es las figuras 6.53 y 6.54. La figura 6.53 muestra la función muestral de correlación cruzada de los errores de predicción de una red que no ha sido adecuadamente entrenada.

Se puede notar que la función de correlación cruzada no cae dentro de los límites definidos por la Ec. (6.118), que se indican con líneas horizontales azules entrecortadas.

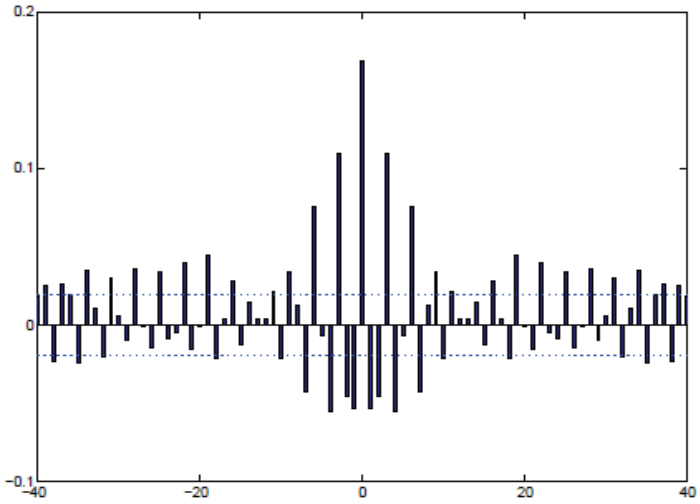


Figura 6.53 $R_{pe}(\tau)$ para una red entrenada inadecuadamente

((Hagan M., 2015) Pág. 17-26)

La figura 6.54 muestra la función de correlación cruzada de una red que ha sido correctamente entrenada. $R_{pe}(\tau)$ cae dentro los límites para todo τ.

Cuando se usa una red NARX, la correlación entre el error de predicción y la entrada sugiere que el retardo en la entrada y en el lazo de realimentación debería ser incrementado.

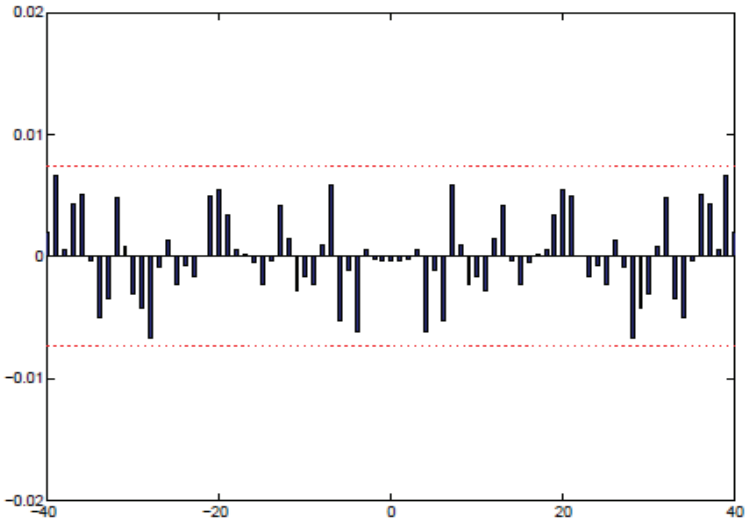


Figura 6.54 $R_{pe}(\tau)$ para una red entrenada correctamente

((Hagan M., 2015) Pág. 17-27)

6.3 EJEMPLO PRÁCTICO CON EL METODO DE REDES NEURONALES

El ejemplo escogido para la predicción con la metodología de redes neuronales es el de un sistema de levitación magnético, el mismo se encuentra en las referencias (Beale M., 2015)(pp: 4-22 a 4-29 y 5-19) y (Hagan M., 2015)(pp:22-2 a 22-14).
En este ejemplo se utilizará el módulo de redes neuronales del programa Matlab ver 15, junto con el programa mostrado en el Anexo D, desarrollado para este efecto por el autor del presente trabajo, con la ayuda de las siguientes referencias (Beale M., 2015), (Heath, 2016).

6.3.1 DESCRIPCIÓN DEL SISTEMA DE LEVITACIÓN MAGNETICO

El objetivo de un sistema de levitación magnética es el control de la posición de un magneto suspendido sobre un electro magneto, el magneto solo se puede mover en dirección vertical, como se indica en la figura 6.55.
La ecuación de movimiento del sistema es:

$$\frac{\delta^2 y(t)}{\delta t^2} = -g + \frac{\alpha}{M}\frac{i^2(t)}{y(t)} - \frac{\beta}{M}\frac{\partial y(t)}{\partial t} \qquad (6.119)$$

Donde $y(t)$ es la distancia del magneto sobre el electro magneto, $i(t)$ es la corriente que circula por el electro magneto, M es la masa del magneto, g es la gravedad. El parámetro β es el coeficiente de fricción viscosa que lo determina el material por donde se mueve el magneto y α es la constante del campo de fuerza que está determinada por el número de vueltas del cable en el electro magneto y la fuerza del magneto. Para este caso de estudio el valor de los parámetros son $\beta = 12$, $\alpha = 15$, $g = 9.8$, $M = 3$.
En este ejemplo se desarrollará una red neuronal que pueda predecir el siguiente valor de la posición del magneto, basado en valores anteriores de la posición del magneto y la corriente de entrada. Una de las aplicaciones prácticas de esta red sería encontrar un controlador que determine la corriente necesaria que se debe aplicar al electro magneto, para que el magneto se mueva a una posición deseada.

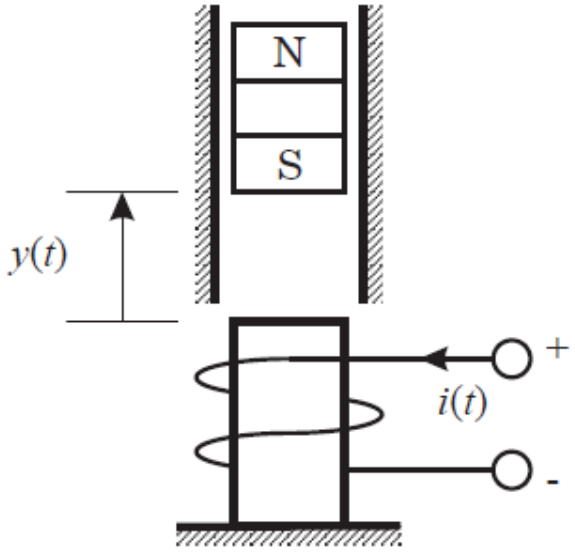


Figura 6.55 Sistema de Levitación Magnético

((Hagan M., 2015; Beale M., 2015) Pág. 5-19)

6.3.2 RECOLECCIÓN DE DATOS

Para este ejemplo el rango permitido de corriente es de [-1 a 4] Amperios. Los datos se han tomado cada 0.01 segundos.

La figuras 6.56 y 6.57 muestran gráficamente la corriente y la correspondiente posición del magneto. Un total de 4000 datos se han recogido.

Como se indicó anteriormente en el caso de series de tiempo se deben utilizar redes neuronales dinámicas. Una arquitectura popular, para este ejemplo, es la red autoregresiva no lineal con entrada exógena NARX (Del inglés nonlinear autoregressive network with exogenous input), que se discutió anteriormente.

La red NARX es una red dinámica recurrente, con realimentación desde su salida.

La ecuación que define el modelo NARX es:

$$y(t) = f(y(t-1), y(t-2), \dots, (y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (6.120)$$

Dónde: el valor siguiente de la salida $y(t)$ es regresada en los valores anteriores de la salida y valores anteriores de la señal de entrada independiente (exógena) $u(t)$. Para este ejemplo, $y(t)$ es la posición del magneto y $u(t)$ es la corriente de entrada al electro magneto. Se puede implementar un modelo NARX utilizando un

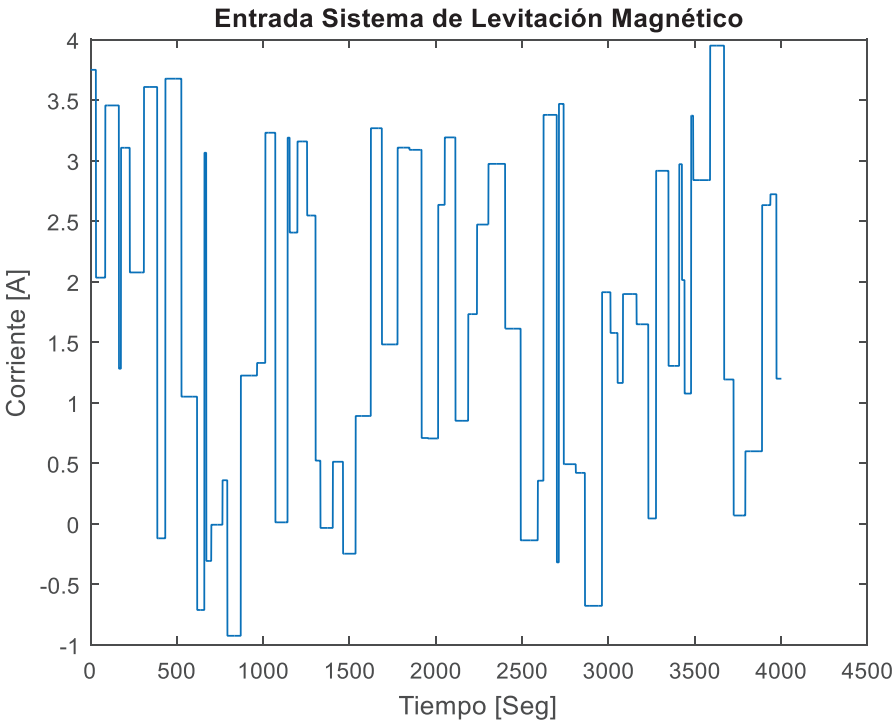


Figura 6.56 Corriente de Entrada

(Gráfico Obtenido con el programa Matlab ver 15)

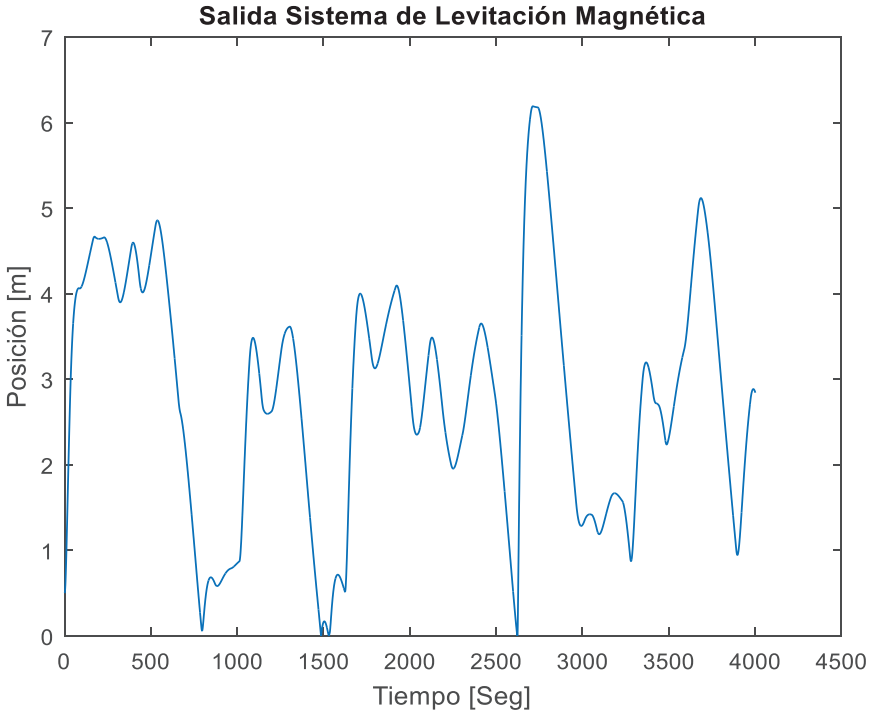


Figura 6.57 Posición del Magneto

(Gráfico Obtenido con el programa Matlab ver 15)

MLP para aproximar la función $f(\,)$. Un diagrama de la red a utilizar se muestra en la figura 6.58, donde un MLP de dos capas se utilizará para la aproximación. La salida de la última capa es la predicción del siguiente valor de la posición del magneto. La entrada de la red es la corriente que ingresa al electro magneto.

6.3.3 ARQUITECTURA DE LA RED

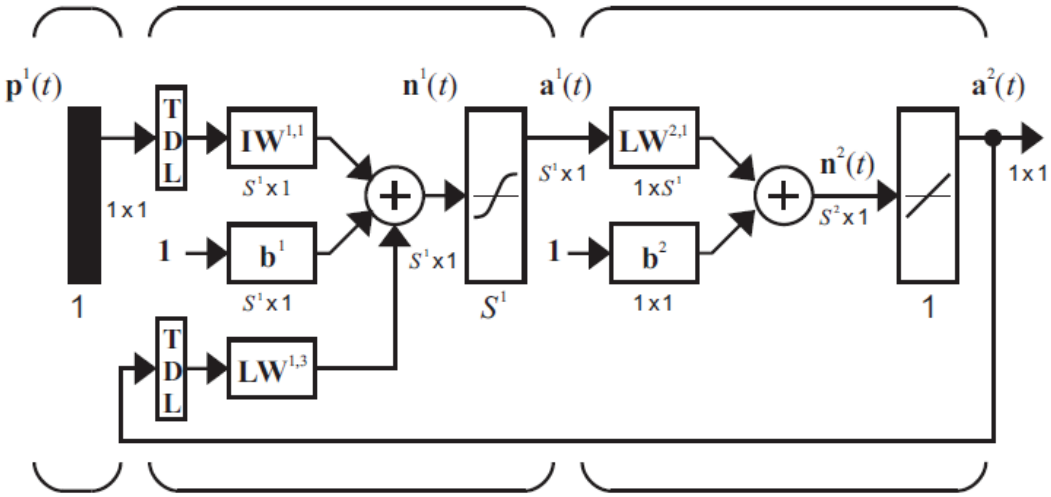


Figura 6.58 Arquitectura de la Red NARX

((Hagan M., 2015; Beale M., 2015) Pág. 22-5)

Se utilizará una función de transferencia tipo $tan - sigmoid$ en la capa oculta y una función $lineal$ en la capa de salida, como se explicó anteriormente, esta es una red estándar para la aproximación de funciones.

Además se debe definir el tamaño del retardo en las líneas TDL. Ya que la Ec. (6.119) es de segundo orden se iniciará con $n_y = n_u = 2$. Para después del entrenamiento ver si este valor es satisfactorio.

Ya que la salida verdadera está disponible durante el entrenamiento de la red, se utilizará una arquitectura serie-paralelo mostrada en la figura 6.50 (derecha), con las ventajas explicadas anteriormente.

Utilizando esta configuración serie-paralelo, se puede utilizar un MLP estándar para implementar el modelo NARX. Se crea un vector de entrada que consiste en los valores anteriores de la entrada y salida del sistema:

$$\mathbf{p} = \begin{bmatrix} u(t-1) \\ u(t-2) \\ y(t-1) \\ y(t-2) \end{bmatrix} \tag{6.121}$$

La salida deseada (objetivo) es el valor siguiente de la salida:

$$\mathbf{t} = [y(t)] \tag{6.122}$$

6.3.4 ENTRENAMIENTO DE LA RED

La cantidad de datos que se tienen disponibles en este ejemplo es muy grande (4000 puntos) comparado con el número de parámetros que se puedan utilizar en la red (alrededor de 100) como se verá luego. Por lo tanto la probabilidad de tener un *sobreajuste* es muy pequeña, es decir se podría utilizar la regularización Bayesiana o el algoritmo de Levenberg-Marquardt para el entrenamiento de esta red. Sin embargo ya que la regularización Bayesiana calcula el número efectivo de parámetros γ que utiliza la red, se iniciará el entrenamiento con este método.

La figura 6.59 muestra el error cuadrático medio (mse) de la red después de 1000 iteraciones durante el entrenamiento. Para este cálculo se han utilizado 10 neuronas en la capa oculta ($S^1 = 10$). Se debe indicar que después de varias simulaciones se subió el tamaño del retardo en las TDLs a $\boldsymbol{n_y = n_u = 4}$, ya que los errores de predicción no fueron satisfactorios (existió correlación entre ellos) con los valores iniciales ($n_y = n_u = 2$).

Se ha repetido cinco veces el entrenamiento con condiciones iniciales diferentes y el error cuadrático medio es similar en cada caso ($4.4\ x\ 10^{-7}$), lo que asegura haber alcanzado un mínimo global.

En la figura 6.60 (pantalla de entrenamiento del Matlab) se puede ver que el número efectivo de parámetros converge a 74, habiendo un total de 101 parámetros en la red (8-10-1), es decir se utilizan aproximadamente las dos terceras partes de los pesos y bías. Si el número efectivo de parámetros estuviera cerca del número total de parámetros, se debería incrementar el número de neuronas en la capa oculta y re entrenar la red. Pero este no es el caso en este ejemplo.

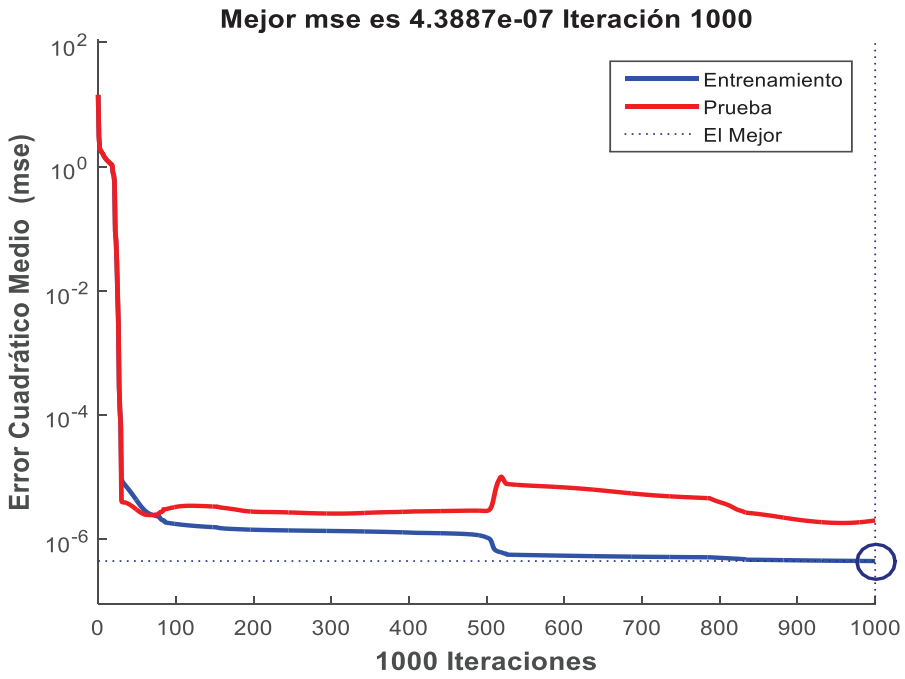


Figura 6.59 MSE Regularización Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

Por otro lado no hay necesidad de disminuir el número de neuronas en la capa oculta, ya que el tiempo de máquina no es crucial en este ejemplo. La otra razón para disminuir el número de neuronas en la capa oculta sería para prevenir el sobreajuste, que tampoco es el caso en este ejemplo por la cantidad de datos que se tiene a disposición. En términos de prevenir el sobreajuste, la ventaja de la regularización Bayesiana es que utiliza el número necesario de parámetros para cada problema, así se tenga un número mayor de parámetros potenciales en la red.

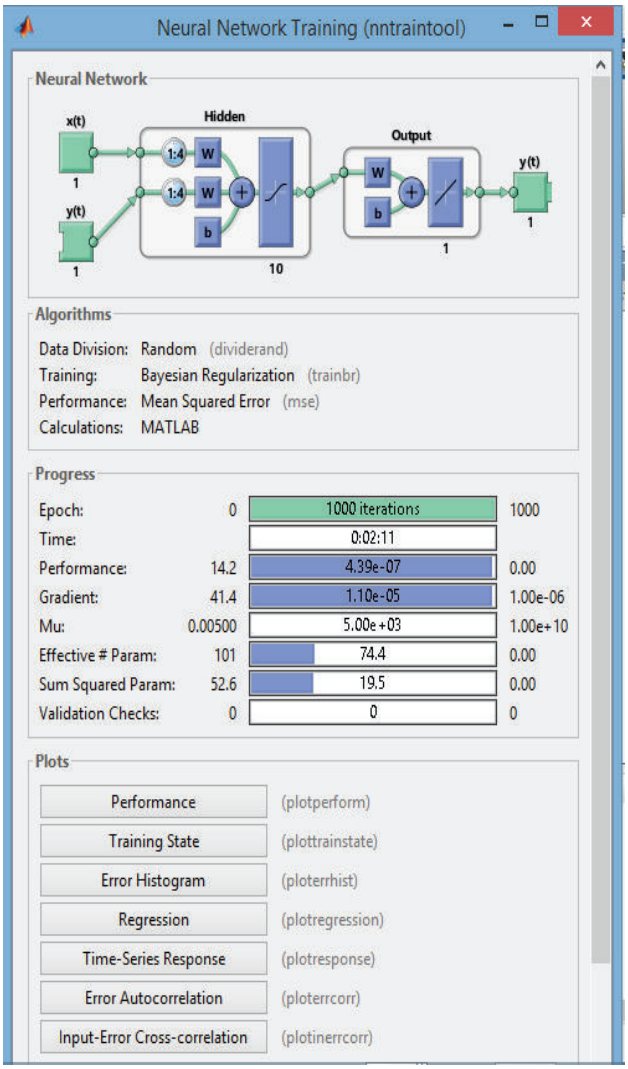


Figura 6.60 Pantalla de Entrenamiento Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

6.3.5 VALIDACIÓN DE LA RED

Recuerde que los datos en las redes neuronales se dividen en tres partes: entrenamiento, validación y prueba (o chequeo). El conjunto de entrenamiento se utiliza para calcular gradientes y calcular la actualización de los pesos. Los datos para validación se utilizan para detener el entrenamiento antes de que ocurra el sobreajuste (overfitting). (Si se utiliza la regularización Bayesiana los datos de validación se unen con los de entrenamiento). Los datos de prueba se usan para

predecir el desempeño futuro de la red. Este índice de desempeño nos da una idea de la calidad de la red. (Hagan M., 2015).

El programa Matlab tiene una importante herramienta para validar la red, muestra los resultados de la regresión entre salidas y objetivos de la red en los datos de prueba y entrenamiento. En la figura 6.61 se puede ver un excelente ajuste tanto en los datos de entrenamiento como en los

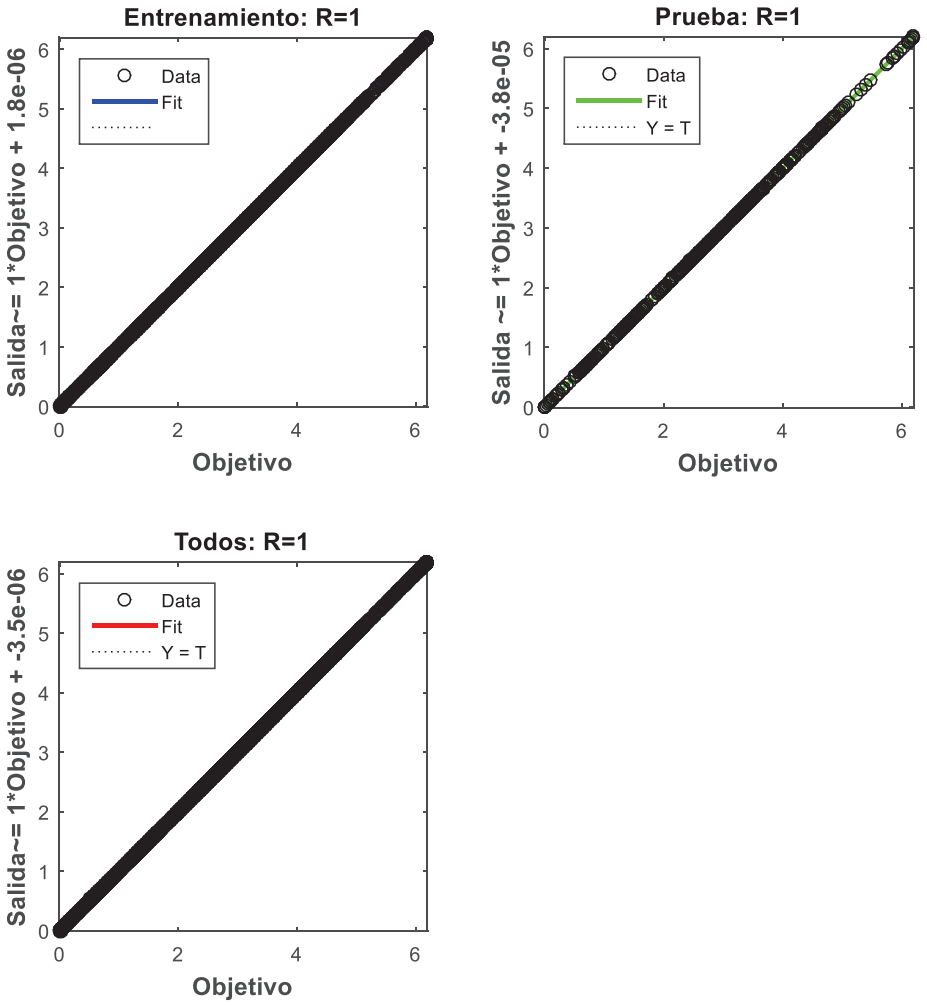


Figura 6.61 Regresión Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

datos de prueba, el coeficiente de regresión ***R*** es igual a 1, es decir ajuste perfecto.

Ya que los datos de prueba ajustan tan bien como los datos de entrenamiento, se puede confiar en que la red no sobre ajustará.

Cuando de pronósticos se trata, se deben utilizar otras dos herramientas para validación. Estas dos herramientas se basan en dos propiedades básicas para la precisión de los modelos de pronósticos.

La primera propiedad es que en el error de predicción,

$$e(t) = y(t) - \hat{y}(t) = y(t) - \ a^2(t), \qquad (6.123)$$

sus valores deben no estar correlacionados entre sí. La segunda propiedad es que los errores de predicción no deben estar correlacionados con la secuencia de entrada $u(t)$. Si existe correlación entre los errores de predicción, entonces se podrá usar esa correlación para mejorar las predicciones. Se puede utilizar el mismo argumento entre la secuencia de entrada y los errores de predicción.

En este ejemplo se muestra el gráfico de la función de autocorrelación muestral en la figura 6.62, se puede notar que es ruido blanco, es decir la función de autocorrelación tiene un impulso a $\tau = 0$ y los otros valores son prácticamente cero (dentro de los límites de confianza Ec.6.116).

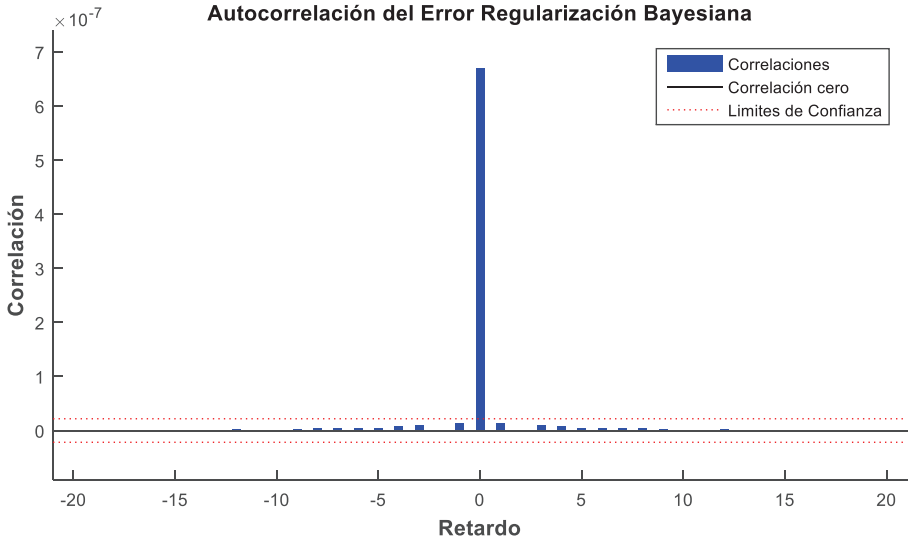


Figura 6.62 Autocorrelación del error Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

En la figura 6.63 se muestra el gráfico de la función de correlación cruzada, todos los valores se encuentran dentro de los límites de confianza (líneas punteadas) por lo que no hay indicios de problemas.

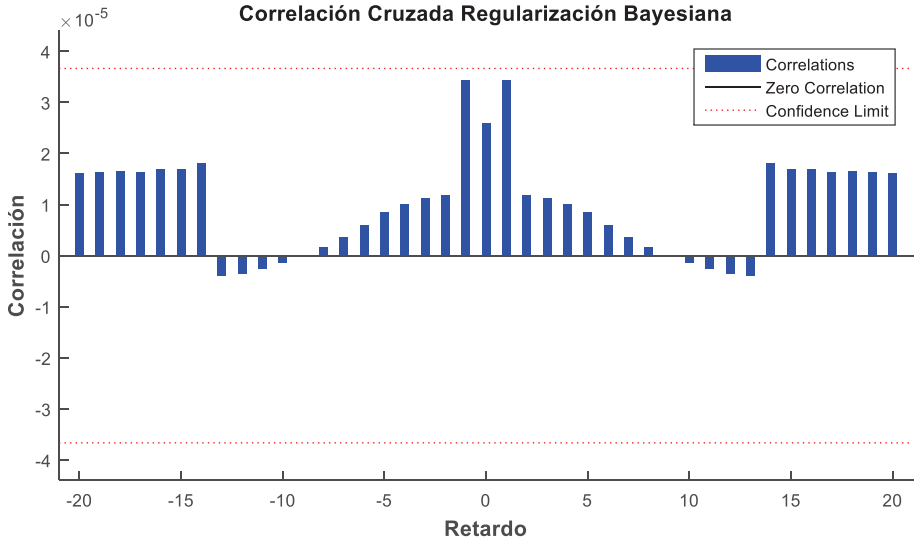


Figura 6.63 Correlación Cruzada Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

Después de analizar estas dos funciones importantes, se puede confiar en que se tiene un modelo de predicción preciso.

La figura 6.64 muestra la respuesta del modelo de predicción final y sus errores, estos errores son muy pequeños. Sin embargo, debido a la configuración serie-paralelo, solo se puede realizar la predicción de un solo paso hacia adelante. Una prueba más rigurosa sería reestablecer la forma original (paralela – lazo cerrado) y predecir algunos pasos hacia adelante, es lo que se verá en la siguiente sección.

6.3.6 PRONÓSTICO DE LA RED

El programa Matlab cuenta con una instrucción (***closeloop***) para convertir redes tipo NARX y otras desde la configuración serie – paralelo (lazo abierto), la cual se usa para el entrenamiento, a la configuración paralelo (lazo cerrado), la cual es muy útil para la predicción varios pasos adelante (ver figura 6.50).

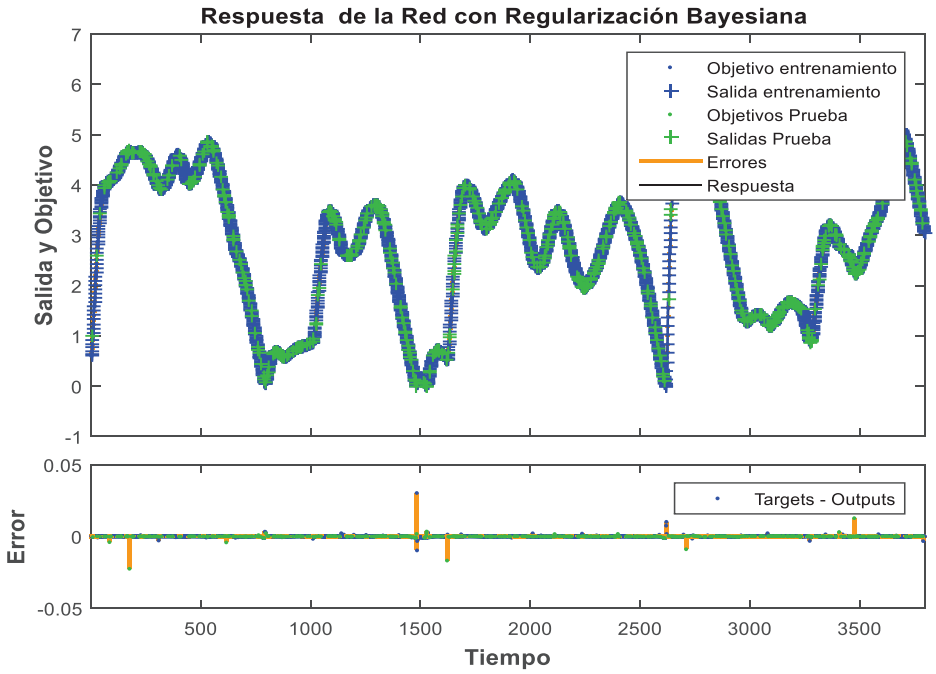


Figura 6.64 Serie de Tiempo con Reg. Bayesiana y sus Errores

(Gráfico Obtenido con el programa Matlab ver 15)

En el Anexo D se presenta el programa *Test_Mag_Data_Narx.m* desarrollado para la predicción del sistema de levitación magnético, debidamente comentado en cada una de las instrucciones importantes.

La instrucción $view(net)$ en el programa presenta el diagrama de la red en la configuración serie – paralelo, la figura 6.65 muestra esta configuración para la red de este ejemplo:

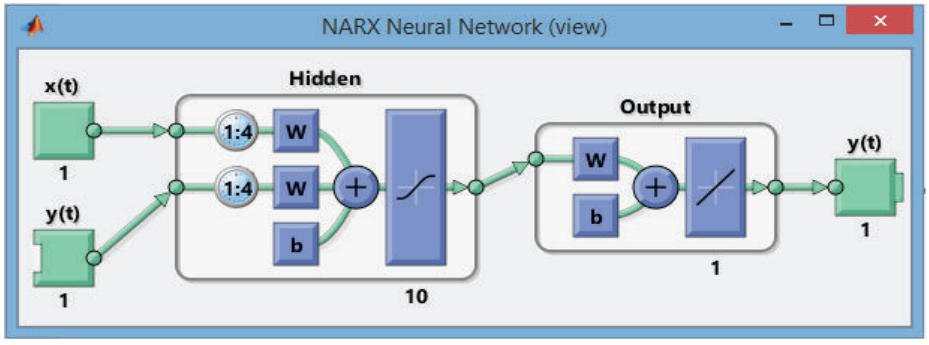


Figura 6.65 Esquema configuración Serie –Paralelo para Pronósticos

(Gráfico Obtenido con el programa Matlab ver 15)

Después de aplicar la instrucción $netc = closeloop(net)$, se muestra nuevamente la red en la figura 6.66.

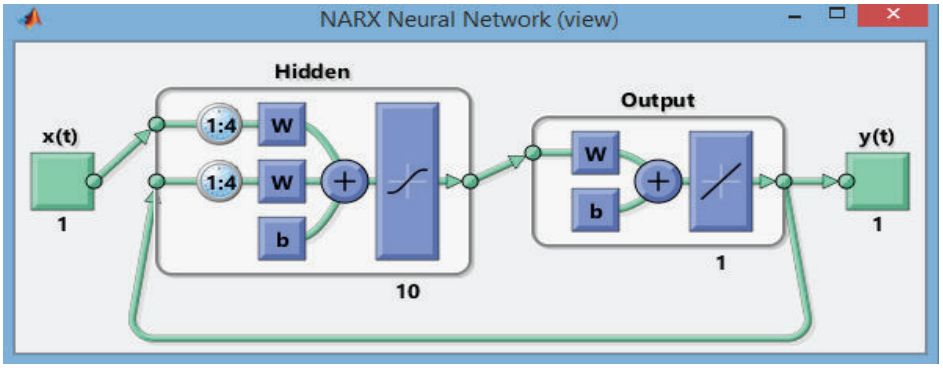


Figura 6.66 Esquema configuración Paralelo para Pronósticos

(Gráfico Obtenido con el programa Matlab ver 15)

Resumiendo, la estrategia es la siguiente: Todo el entrenamiento se hace en lazo abierto (arquitectura serie-paralelo), incluyendo la validación y chequeo. Una vez terminado el entrenamiento (incluyendo la validación y chequeo) la red se transforma a lazo cerrado (arquitectura paralela) para realizar la predicción varios pasos adelante. (Beale M., 2015).

A continuación, se muestra en la figura 6.67 el resultado (configuración en lazo cerrado (figura 6.66)) de la predicción de 200 pasos adelante, para el sistema de levitación magnético.

La figura 6.67 muestra en amarillo la salida esperada y en café el pronóstico de la red (en lazo cerrado), aunque son 200 pasos hacia adelante, la predicción es muy precisa.

Es importante recalcar que la predicción en lazo cerrado será más precisa mientras menor sea el error de entrenamiento en la configuración serie paralelo (lazo abierto). (Beale M., 2015).

En la figura 6.68 se muestra la predicción ampliada, se puede apreciar mejor lo adecuado del pronóstico.

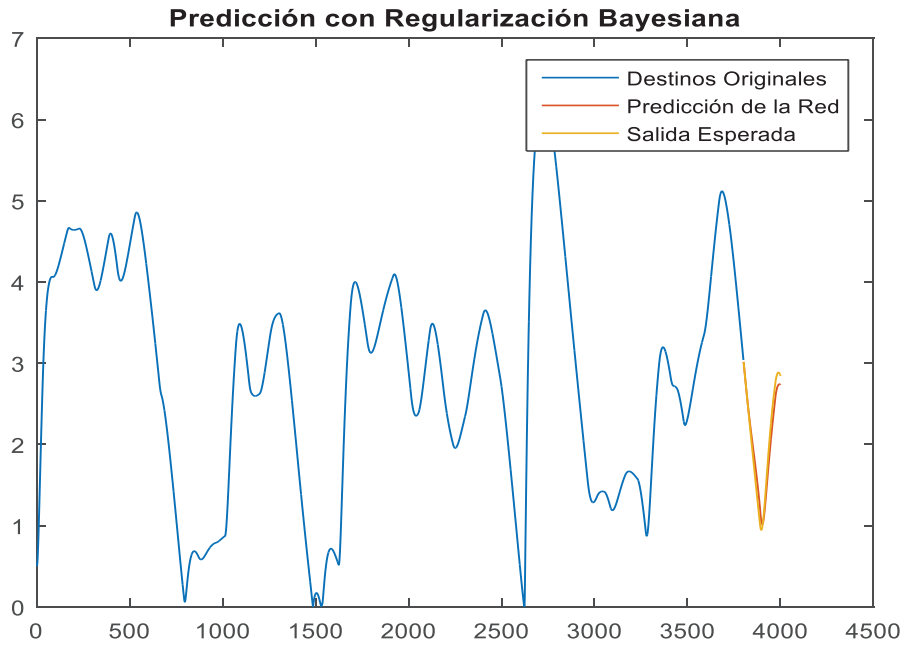


Figura 6.67 Pronóstico 200 pasos adelante configuración Paralelo

(Gráfico Obtenido con el programa Matlab ver 15)

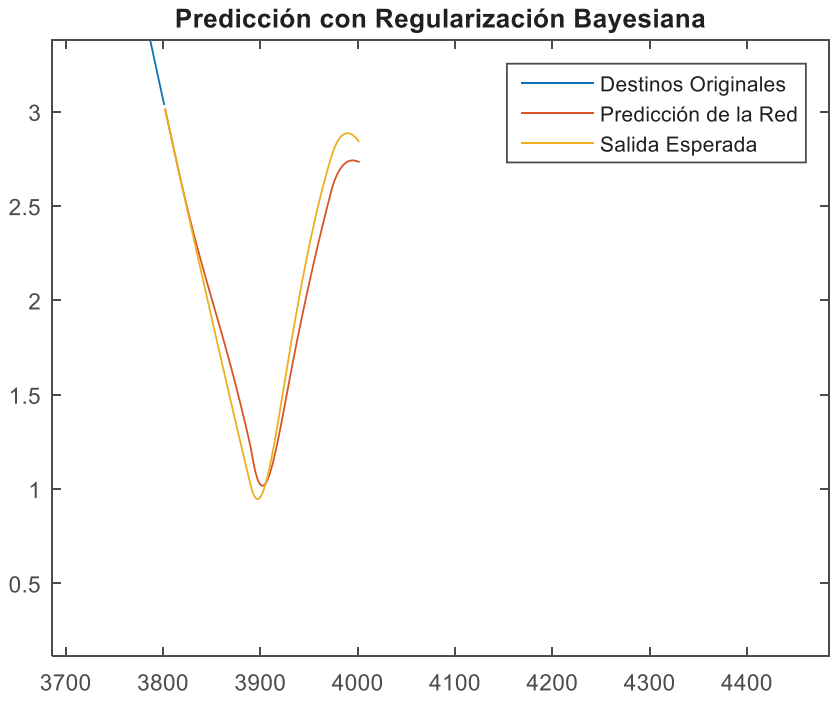


Figura 6.68 Pronóstico 200 pasos adelante configuración Paralelo Ampliado

(Gráfico Obtenido con el programa Matlab ver 15)

Para terminar, cabe indicar que cada vez que una red neuronal es entrenada, esta nos presenta un resultado diferente, esto se debe a que los valores iniciales de pesos y bías para cada entrenamiento son diferentes y también la división de datos para entrenamiento, validación y chequeo. Para asegurar que una precisión adecuada se ha encontrado (mínimo global), es necesario entrenar varias veces la red. (Beale M., 2015).

Como se trata de un ejemplo pedagógico, se van a presentar los resultados de la misma arquitectura de red neuronal, pero cambiando el método de entrenamiento por el de Levenberg-Marquardt (LM).

La diferencia de este algoritmo con la regularización Bayesiana es que el algoritmo de LM si realiza la validación cruzada.

La figura 6.69 muestra el desempeño de la red, como se indicó anteriormente si el

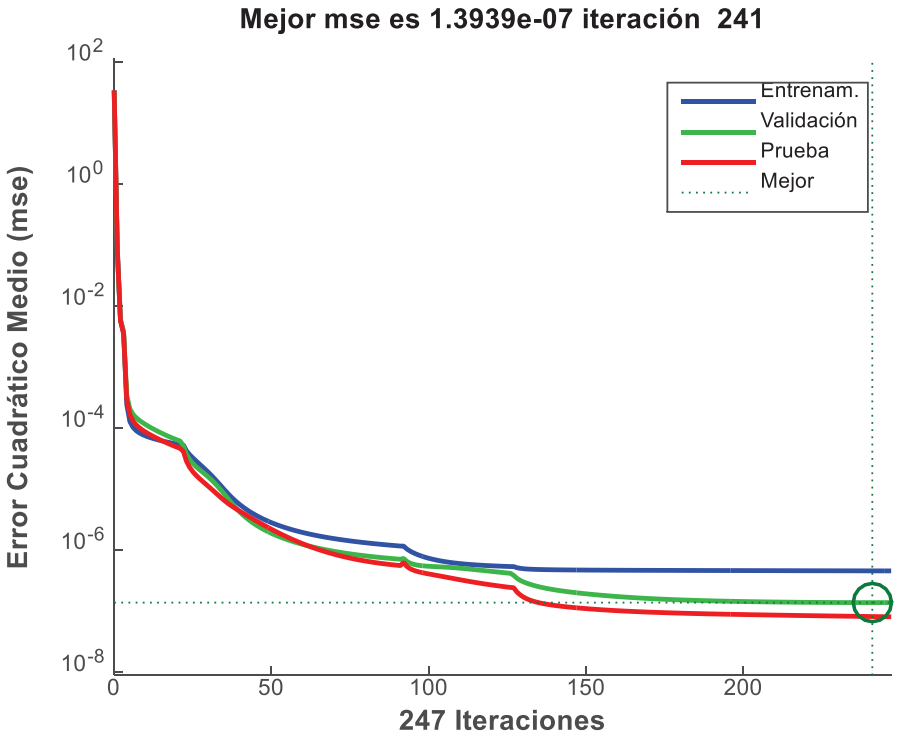


Figura 6.69 MSE Algoritmo Levenberg-Marquardt

(Gráfico Obtenido con el programa Matlab ver 15)

error de validación crece por más de seis iteraciones el programa produce una parada temprana, para evitar el sobreajuste. Eso es justamente lo que pasa en este caso, el programa detiene el entrenamiento en la iteración número 247.

Además, se puede notar que el índice de desempeño de la red (mse) es de $1.37x10^{-7}$, menor que para la regularización Bayesiana ($4.38x10^{-7}$) por lo tanto se espera un mejor pronóstico.

Ya que se ha producido una parada temprana, se puede utilizar la red con la confianza de que no se producirá el sobre ajuste.

Las figuras 6.70 y 6.71 muestran la función de autocorrelación y correlación cruzada con el algoritmo de LM.

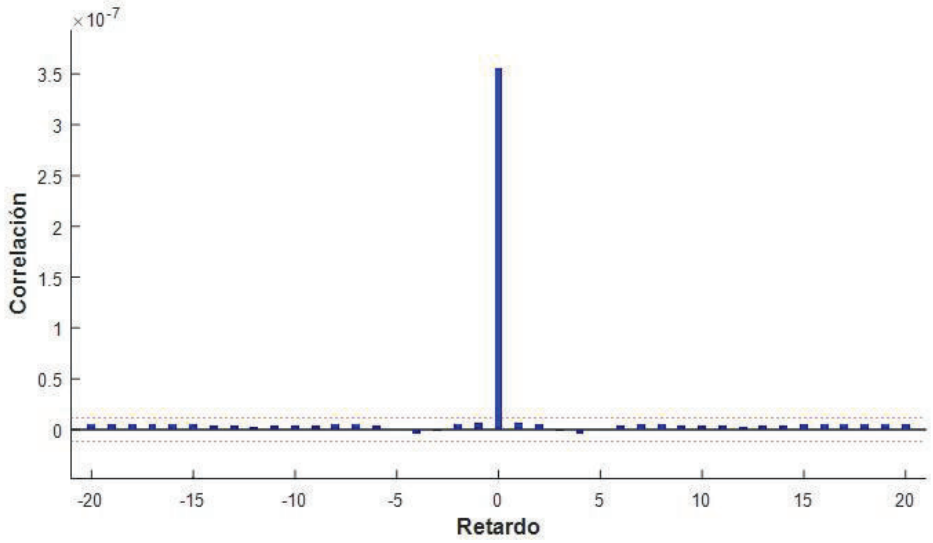


Figura 6.70 Autocorrelación del error con Levenberg-Marquardt

(Gráfico Obtenido con el programa Matlab ver 15)

Se puede notar en los dos casos que sus valores están dentro de los límites de confianza, por lo tanto se puede garantizar que los valores de $\boldsymbol{n_y = n_u = 4}$ son también adecuados en este caso.

Siguiendo la misma estrategia de pronóstico (entrenamiento, validación y chequeo – configuración serie-paralelo y predicción varios pasos adelante – configuración paralela) se calculan las predicciones 200 pasos adelante, los resultados se muestran en las figuras 6.72 y 6.73.

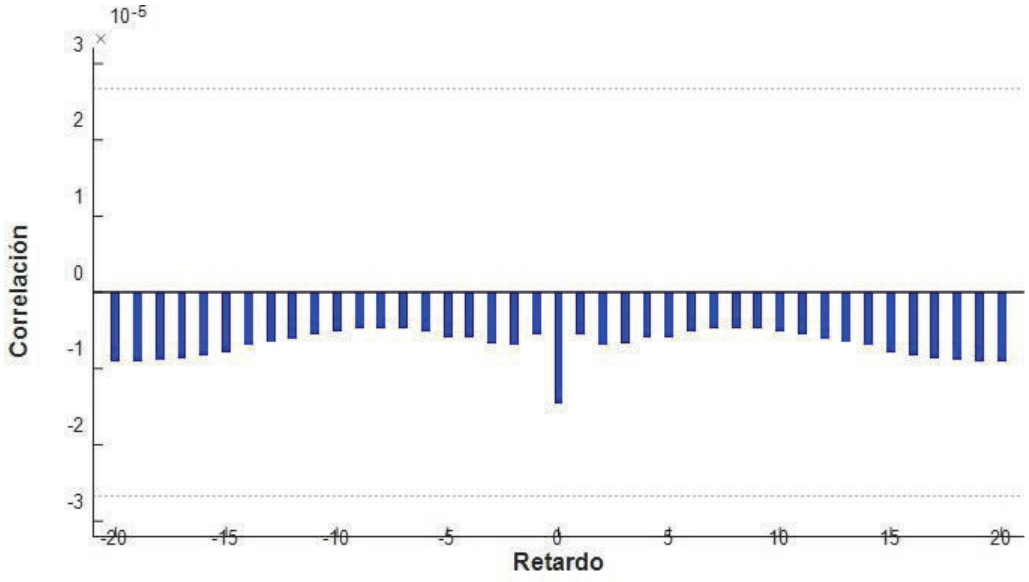


Figura 6.71 Correlación Cruzada con Levenberg-Marquardt

(Gráfico Obtenido con el programa Matlab ver 15)

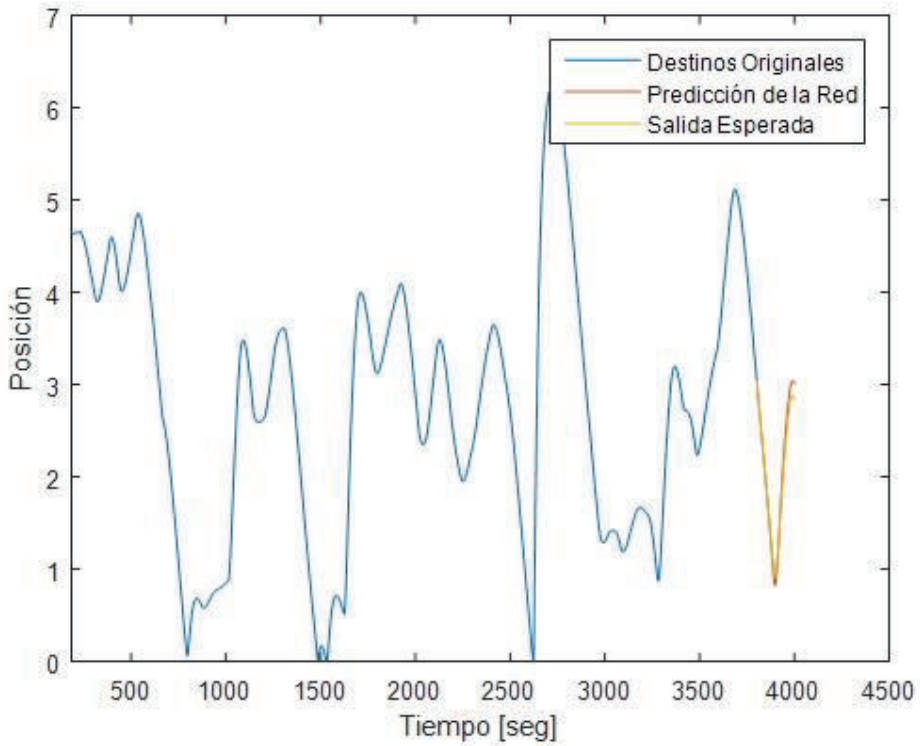


Figura 6.72 Pronóstico 200 pasos adelante configuración Paralelo con LM

(Gráfico Obtenido con el programa Matlab ver 15)

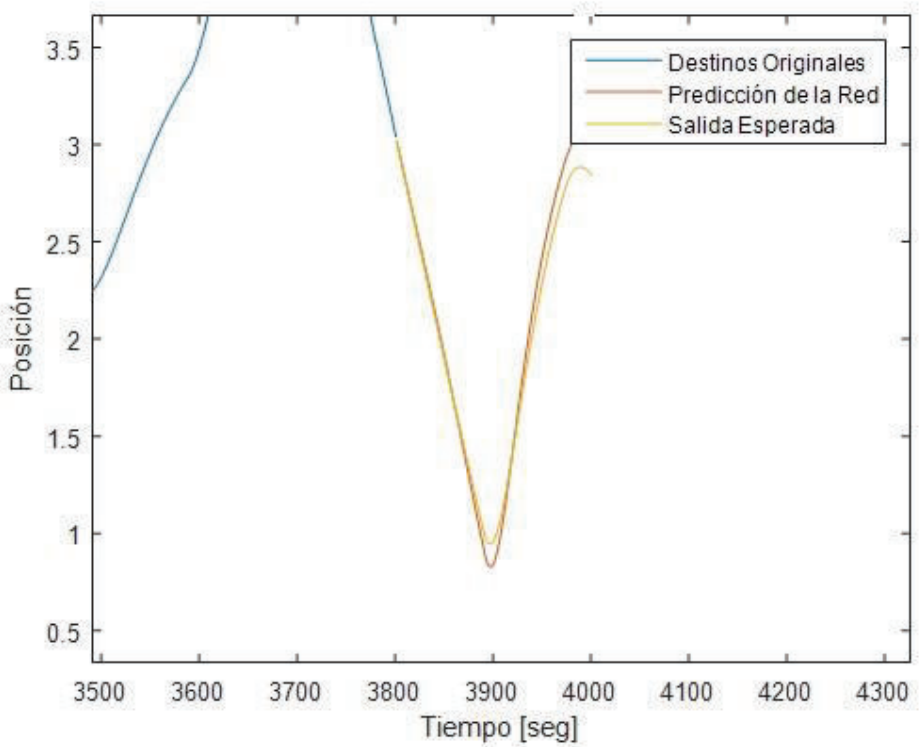


Figura 6.73 Pronóstico 200 pasos adelante configuración Paralelo con LM Ampliado

(Gráfico Obtenido con el programa Matlab ver 15)

Efectivamente, después de comparar las figuras 6.67 y 6.68 con las figuras 6.72 y 6.73, en este ejemplo en particular el entrenamiento con el algoritmo de Levenberg-Marquardt produce un pronóstico ligeramente superior.

6.4 PRONÓSTICO PARA LA DEMANDA DE PLACAS DIGITALES EN EL MERCADO GRÁFICO QUITEÑO DESDE EL AÑO 2009 HASTA EL AÑO 2015, REDES NEURONALES.

6.4.1 DATOS PARA EL PRONÓSTICO

En la tabla 6.2 se muestra la información del consumo de placas digitales formato 510x400x0.15 a pronosticar (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo GTO_52, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utilizará el paquete Matlab ver 15 y el programa *Prog_Narnet_Tesis_OK.m* desarrollado específicamente por el autor del presente trabajo, el programa utiliza el módulo de redes neuronales de la plataforma Matlab ver 15 para el pronóstico mediante este método, luego se calculará el error de pronóstico MAPE y se comparará con los otros dos métodos de pronósticos tratados en capítulos anteriores.

Tabla 6.2 Consumo de Placas Formato 510x400x0.15 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	16600	16750	15300	17400	32390	45300	33700
2	16600	18000	18750	20350	30400	39672	30660
3	26850	26500	17750	24950	36300	34000	37900
4	16250	22800	14400	23800	37200	36862	35885
5	18150	23000	16300	37600	37956	42900	36400
6	18550	22150	19100	25323	32044	39233	36190
7	22200	20650	16100	29050	37700	38025	33900
8	21250	22265	16400	35600	34526	36300	32400
9	19300	25230	17700	30350	38880	37680	38300
10	21400	23100	18200	37100	41982	41911	36000
11	19950	18700	15150	42260	47700	47457	39300
12	28850	22800	19000	40900	48464	47563	39300

En la figura 6.74 se muestran el gráfico de la demanda de placas digitales desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Se trabajará con los datos hasta el primer semestre del año 2015 (78 observaciones) y se reservarán los datos del segundo semestre del año 2015 (6 observaciones) para comparar el pronóstico generado mediante redes neuronales con los valores reales, finalmente se compararán los tres métodos de pronósticos.

6.4.2 ARQUITECTURA DE LA RED

Como se indicó anteriormente en el caso de series de tiempo se deben utilizar redes neuronales dinámicas. En la referencia (Beale M., 2015)(pp:4-13 - 4-18) recomienda que una arquitectura adecuada para este problema (una sola salida), es la red autoregresiva no lineal NARnet (Del inglés nonlinear autoregressive network), que se discutió anteriormente en la sección 6.2.5.2. (pp:274-276)

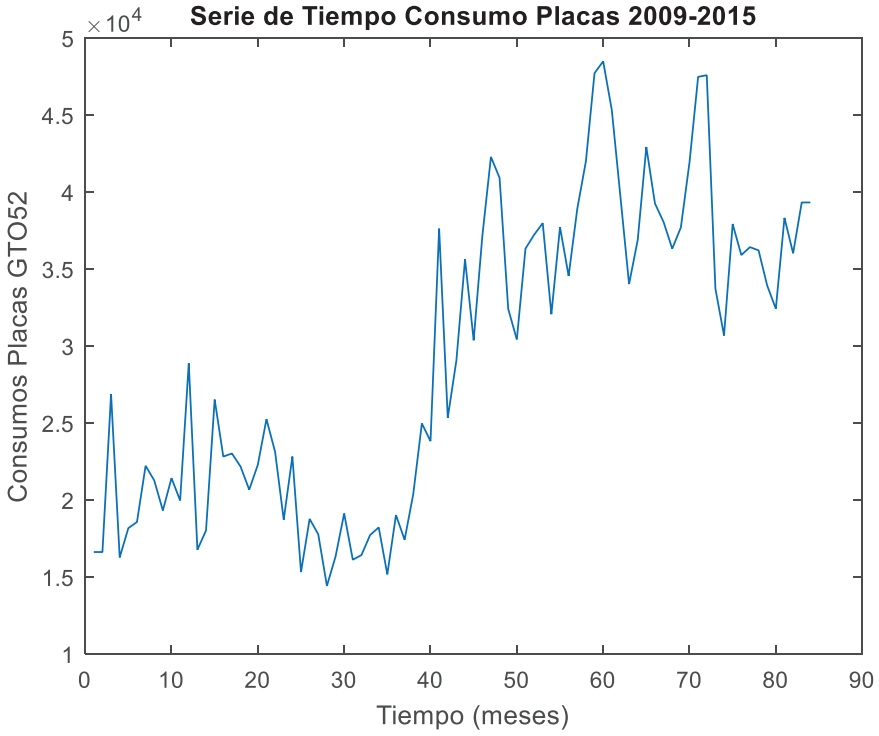


Figura 6.74 Consumo de placas digitales formato 510x400x0.15 (GTO_52) (2009 – 2015)

(Gráfico Obtenido con el programa Matlab ver 15)

La red tipo NAR es una red dinámica recurrente, con realimentación desde su salida.

La ecuación que define el modelo NAR es:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y)) \quad (6.124)$$

Dónde: el valor siguiente de la salida $y(t)$ es regresada en los valores anteriores de la salida. Para este problema, $y(t)$ es la predicción de la demanda de placas digitales formato GTO_52. Se puede implementar un modelo Narnet utilizando un MLP para aproximar la función $f(\,)$. Un diagrama de la red a utilizar se muestra en la figura 6.75, donde un MLP de dos capas se utilizará para la aproximación. La salida de la última capa $a^2(t)$ es la predicción del siguiente valor de la demanda.

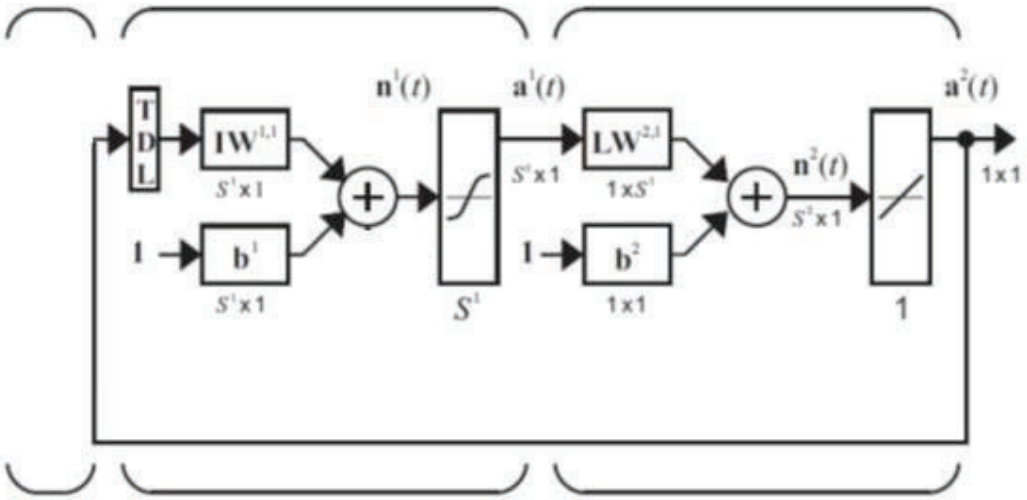


Figura 6.75 Arquitectura de una Red Tipo NAR

(Beale M., 2015)

Se utilizará una función de transferencia tipo $tan - sigmoid$ en la capa oculta y una función $lineal$ en la capa de salida, como se explicó anteriormente en la sección 6.2.3.2, esta es una red estándar para la aproximación de funciones. Se utiliza la función $tan - sigmoid$, ya que esta produce salidas (que son entradas para la siguiente capa) que son prácticamente centradas a cero, lo que no pasa con la función $log - sigmoid$ o sigmoidal que produce salidas positivas. Esta característica es muy útil para pronósticos. (Hagan M., 2015).

El número de neuronas en la capa oculta, S^1, dependerá de que tan compleja es la función a aproximar. Esto es algo que normalmente no se conoce antes del entrenamiento. En la literatura la mayoría de autores coincide en que este tema

está todavía bajo investigación, hay dos criterios relevantes: *el primero* que se muestra en la referencia (Hagan M., 2015) (pp:17-12) que recomienda iniciar con más neuronas de lo necesario y luego utilizar la regularización Bayesiana o parada temprana para evitar el sobreajuste, *el segundo* se da en la referencia (Azoff, 1994)(pp:49-51) el mismo que su vez hace referencia al teorema de Kolmogorov que sugiere tener al menos $(2N + 1)$ neuronas en la capa oculta, para luego ir incrementando este número durante el entrenamiento. En este caso iniciaremos con $S^1 = 10$ neuronas (mayor a 5 que recomienda Kolmogorov) y se utilizará la regularización Bayesiana para evitar el sobreajuste.

Además, se debe definir el tamaño del retardo en las líneas TDL. Se iniciará con valor de $n_y = 2$. Para después del entrenamiento ver si este valor es satisfactorio.

Ya que la salida verdadera está disponible durante el entrenamiento de la red, se utilizará una arquitectura serie-paralelo mostrada en la figura 6.50 (derecha), con las ventajas explicadas anteriormente. (Sección 6.2.5.2) (pp:274-276).

Utilizando esta configuración serie-paralelo, se puede utilizar un MLP estándar para implementar el modelo NARnet. La línea de retardo (TDL) se reemplazará por un vector de entrada que consiste en los valores anteriores de la salida del sistema:

$$\mathbf{p} = \begin{bmatrix} y(t-1) \\ y(t-2) \end{bmatrix} \tag{6.125}$$

La salida deseada (objetivo) es el valor siguiente de la salida:

$$\mathbf{t} = [y(t)] \tag{6.126}$$

Es decir se tendrán dos nodos de entrada y uno de salida. Como el número de neuronas en la capa de salida es igual al número de elementos del vector de salida deseada $\mathbf{t}$, en este caso será uno.

6.4.3 ENTRENAMIENTO DE LA RED

La cantidad de datos disponibles en este caso es muy pequeña (78 puntos) comparado con el número de parámetros que se puedan utilizar en la red (red 2-10-1 con 41 parámetros). En este punto en las referencias (Azoff, 1994)(pp-51) y (Del Brío, 2007) (pp-74), se recomienda tener como número de datos para el

entrenamiento al menos diez veces el número de parámetros (en este caso al menos 410 puntos), para que la generalización no sea deficiente.

Por lo tanto en este caso la probabilidad de tener un *sobreajuste* es muy alta, por esta razón y ya que la regularización Bayesiana calcula el número efectivo de parámetros γ que utiliza la red, se iniciará el entrenamiento con este método.

En la referencia (Beale M., 2015)(pp:9-16 a 9-30), se hace una muy buena comparación entre nueve algoritmos de entrenamiento para redes neuronales en diferentes aplicaciones, se puede concluir de este estudio que para el caso de aproximación de funciones el algoritmo de Levenberg-Marquardt es el mejor, especialmente si la red no tiene una gran cantidad de parámetros. Ya que los pronósticos caen dentro de la categoría de aproximación de funciones se utilizará este algoritmo también para comparar con la regularización Bayesiana.

Entrenamiento con Regularización Bayesiana

La figura 6.76 muestra el error cuadrático medio (mse) de la red después de 1000 iteraciones durante el entrenamiento. Para este cálculo se han utilizado 10 neuronas en la capa oculta ($S^1 = 10$). La línea de retardo TDL se asigna $\boldsymbol{n_y = 2}$.

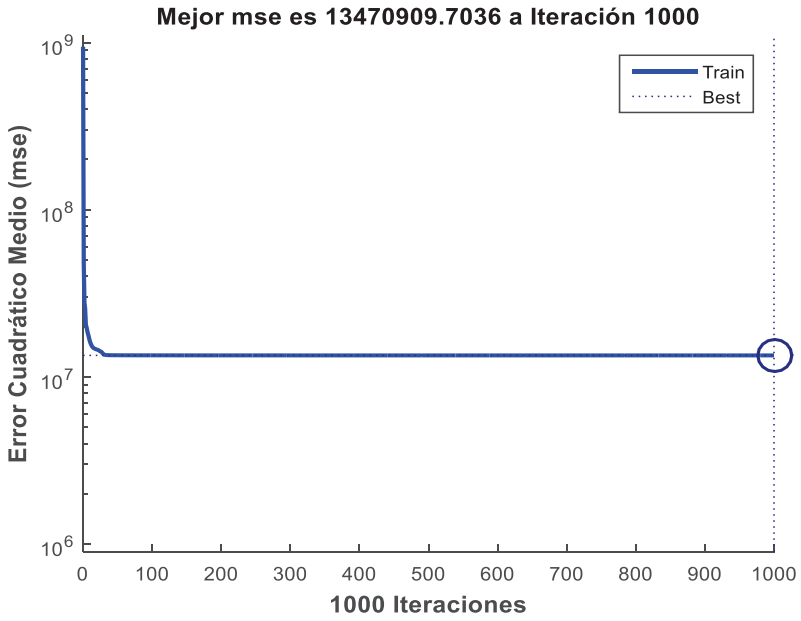


Figura 6.76 MSE Regularización Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

Se han realizado muchas simulaciones para garantizar que el entrenamiento se ejecute con condiciones iniciales diferentes, se puede asegurar que la red ha alcanzado un mínimo global ya que produce el menor error de pronóstico MAPE (como se verá más adelante).

En la figura 6.77 (pantalla de entrenamiento del Matlab) se puede ver que el número efectivo de parámetros converge a 10.7, habiendo un total de 41 parámetros en la red (2-10-1), es decir se utiliza aproximadamente la cuarta parte de los pesos y bías. Después de realizar varias simulaciones el número efectivo de parámetros converge entre los valores de 9 y 12. Esto podría indicar que se puede utilizar una red más pequeña, si el tiempo de máquina sería importante, como no es el caso dejaremos el número de neuronas en la capa oculta en 10, ya que la función a aproximar es complicada, ver figura 6.74.

La otra razón para disminuir el número de neuronas en la capa oculta sería para prevenir el sobreajuste, que tampoco es el caso en este problema ya que se está utilizando la regularización Bayesiana, la misma que utiliza el número necesario de parámetros para cada problema, así se tenga un número mayor de parámetros potenciales en la red.

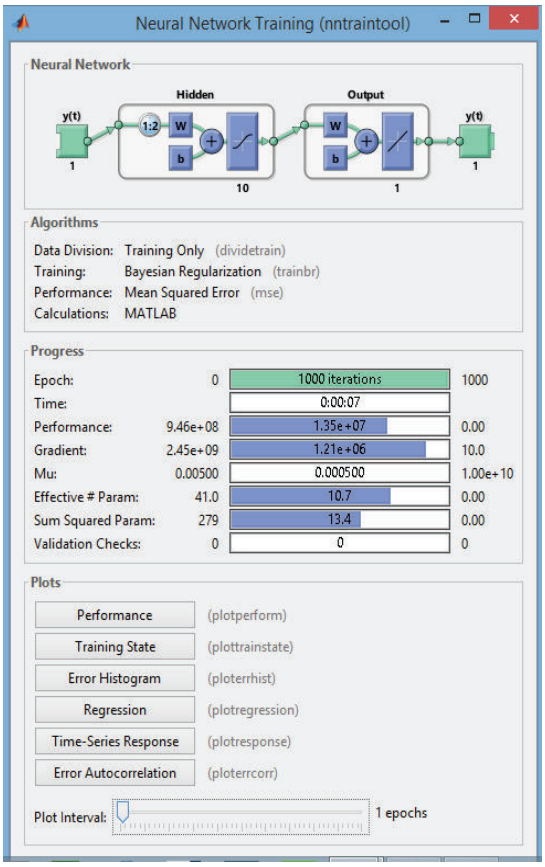


Figura 6.77 Pantalla de Entrenamiento Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

6.4.4 VALIDACIÓN DE LA RED

Recuerde que los datos en las redes neuronales se dividen en tres partes: entrenamiento, validación y prueba (o chequeo). En este caso, y después de realizar muchas simulaciones, por la cantidad limitada de datos no se realizará esta división y más bien se utilizarán todos los datos para el entrenamiento de la red.

El programa Matlab tiene una importante herramienta para validar la red, muestra los resultados de la regresión entre salidas y objetivos de la red en los datos de entrenamiento. En la figura 6.78 se puede ver un buen ajuste $R = 0.928$ (mayor a 0.9).

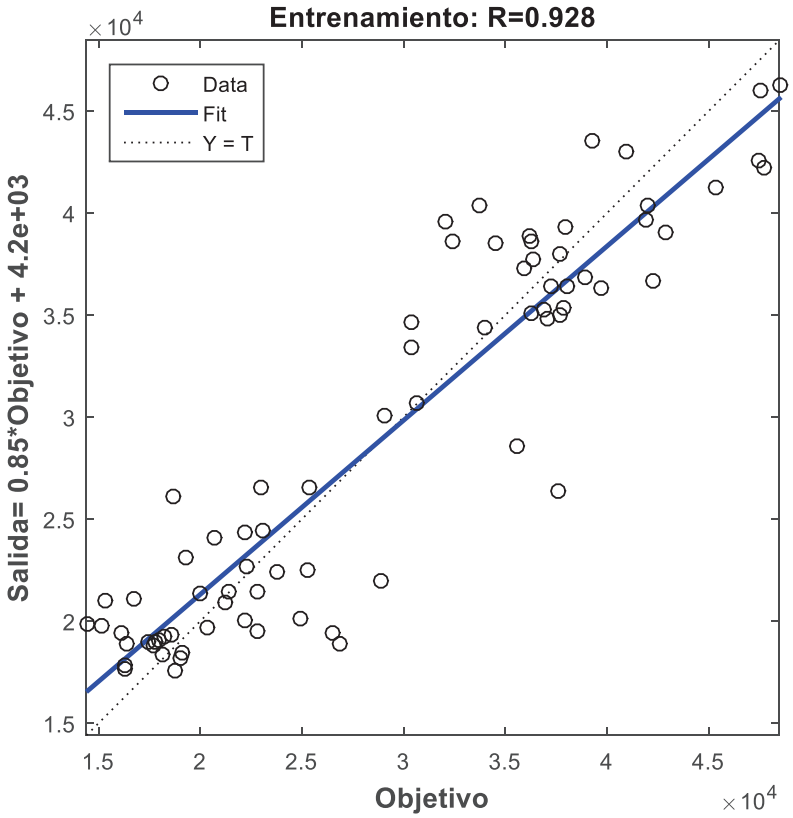


Figura 6.78 Regresión Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

Cuando de pronósticos se trata, en la red Narnet se debe utilizar otra herramienta para validación. La función de autocorrelación del error, que se analizó en detalle en el ejemplo anterior. No se utiliza la función de autocorrelación cruzada, ya que no existe entrada exógena.

En este caso se muestra el gráfico de la función de autocorrelación del error en la figura 6.79, se puede notar que no es ruido blanco, es decir la función de autocorrelación tiene algunos valores que salen de los límites de confianza. Lo que significa que se puede mejorar el pronóstico.

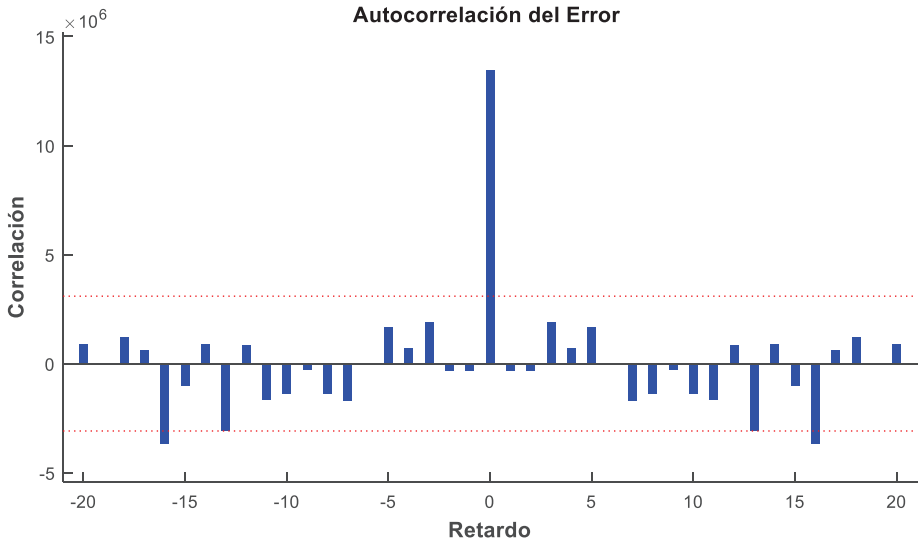


Figura 6.79 Autocorrelación del error Reg. Bayesiana

(Gráfico Obtenido con el programa Matlab ver 15)

La figura 6.80 muestra la respuesta de este modelo de predicción final y sus errores, estos errores son bastante moderados. Sin embargo, debido a la configuración serie-paralelo, solo se puede realizar la predicción de un solo paso hacia adelante. Una prueba más rigurosa sería reestablecer la forma original (paralela – lazo cerrado) y predecir algunos pasos hacia adelante, es lo que se verá en la siguiente sección.

6.4.5 PRONÓSTICO DE LA RED

El programa Matlab cuenta con una instrucción (***closeloop***) para convertir redes tipo NAR y otras desde la configuración serie – paralelo (lazo abierto), la cual se usa para el entrenamiento, a la configuración paralelo (lazo cerrado), la cual es muy útil para la predicción varios pasos adelante (ver figura 6.50).

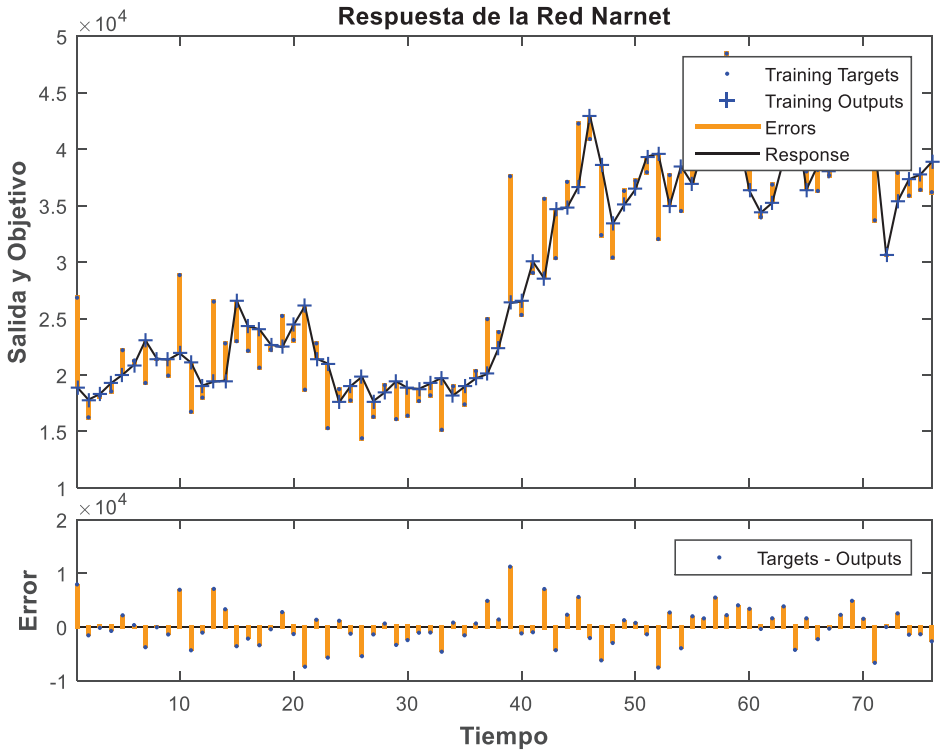


Figura 6.80 Serie de Tiempo con Reg. Bayesiana y sus Errores

(Gráfico Obtenido con el programa Matlab ver 15)

En el Anexo E se presenta el programa *Prog_Narnet_Tesis_OK.m* desarrollado para la predicción de la demanda de placas digitales en el mercado gráfico quiteño desde el año 2009 hasta el año 2015, por el autor del presente trabajo, debidamente comentado en cada una de las instrucciones importantes.

La instrucción $view(net)$ en el programa presenta el diagrama de la red en la configuración serie – paralelo, la figura 6.81 muestra esta configuración para la red Narnet:

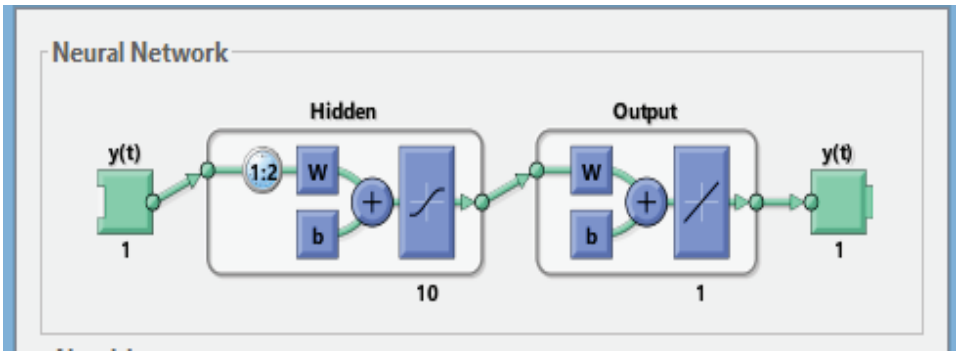


Figura 6.81 Esquema configuración Serie –Paralelo para Pronósticos

(Gráfico Obtenido con el programa Matlab ver 15)

Después de aplicar la instrucción $netc = closeloop(net)$, se muestra de nuevo la red en la figura 6.82.

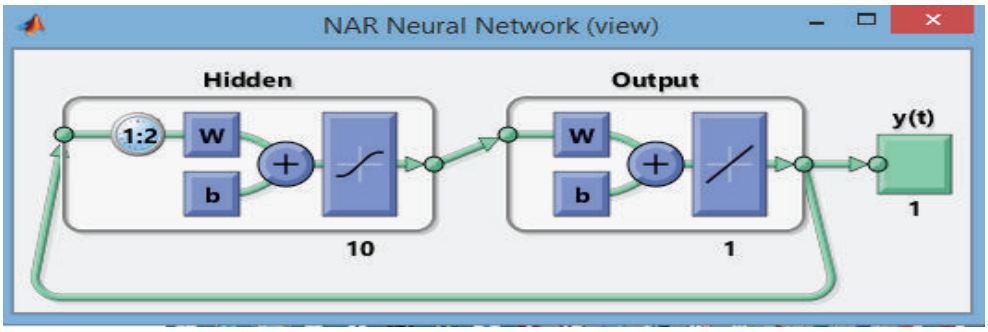


Figura 6.82 Esquema configuración Paralelo para Pronósticos

(Gráfico Obtenido con el programa Matlab ver 15)

Resumiendo, la estrategia es la siguiente: Todo el entrenamiento se hace en lazo abierto (arquitectura serie-paralelo). Una vez terminado el entrenamiento la red se transforma a lazo cerrado (arquitectura paralela) para realizar la predicción varios pasos adelante. (Beale M., 2015).

A continuación, se muestra en la figura 6.83 el resultado (configuración en lazo cerrado (figura 6.82)) de la predicción de 6 pasos adelante, para demanda de placas digitales.

La figura 6.83 muestra en amarillo la salida esperada y en café el pronóstico de la red (en lazo cerrado), se puede ver que la predicción no es muy precisa.

En la figura 6.84 se muestra la predicción ampliada, para poder apreciar mejor el pronóstico.

La figura 6.84 muestra los resultados numéricos de la predicción en Matlab.

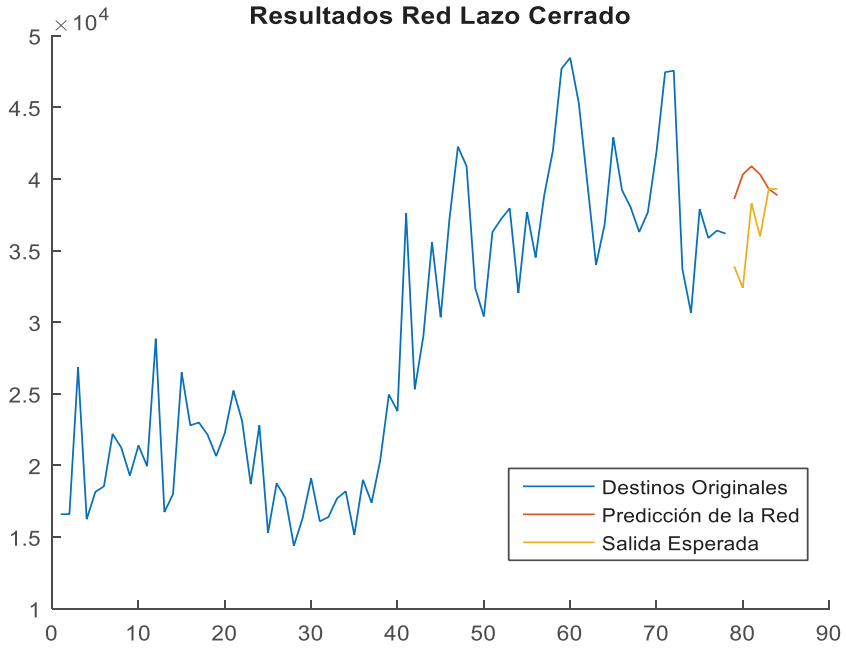


Figura 6.83 Pronóstico 6 pasos adelante configuración Paralelo

(Gráfico Obtenido con el programa Matlab ver 15)

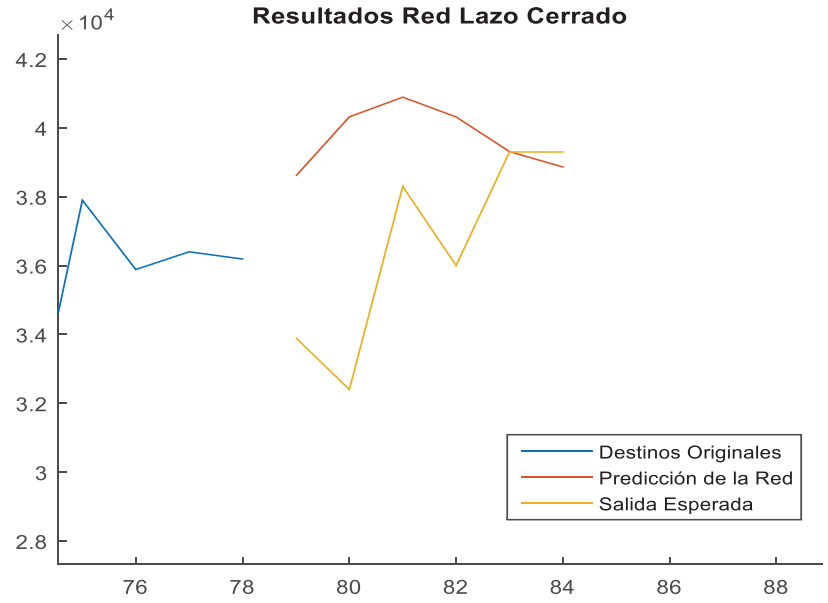


Figura 6.84 Pronóstico 6 pasos adelante configuración Paralelo Ampliado

(Gráfico Obtenido con el programa Matlab ver 15)

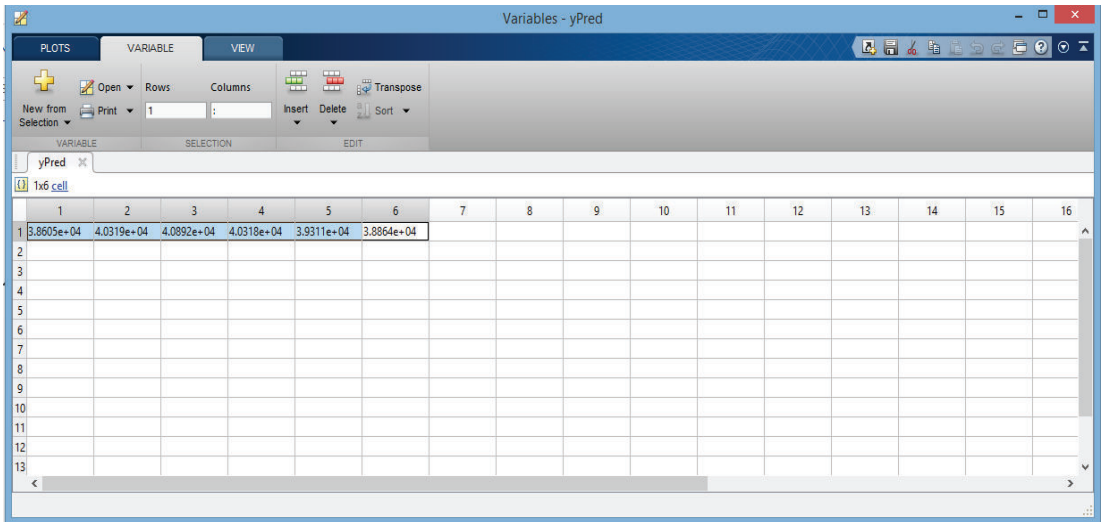


Figura 6.85 Resultado Numérico del Pronóstico 6 pasos adelante

(Gráfico Obtenido con el programa Matlab ver 15)

A continuación, se van a presentar los resultados de la misma arquitectura de red neuronal, pero cambiando el método de entrenamiento por el de Levenberg-Marquardt (LM).

Igual que en la regularización Bayesiana, se utilizarán todos los datos para el entrenamiento, no se realizará la división de datos, ya que el conjunto de datos es muy pequeño y conviene utilizarlos a todos para el entrenamiento.

Para esto se utiliza la instrucción net.divideFcn = 'dividetrain', en lugar de net.divideFcn = 'dividerand', como se puede observar en el programa Prog_Narnet_Tesis_OK.m en el Anexo E.

La figura 6.86 muestra el desempeño de la red, se puede notar que el índice de desempeño de la red (mse) es de $7.92x10^6$, menor que para la regularización Bayesiana ($1.34x10^7$) por lo tanto se espera un mejor pronóstico.

Las figura 6.87 muestra la función de autocorrelación con el algoritmo de LM.

Se puede notar que sus valores están dentro de los límites de confianza, por lo tanto se puede garantizar que los valores de $\boldsymbol{n_y = n_u = 2}$ son adecuados en este caso y el pronóstico será más preciso.

Siguiendo la misma estrategia de pronóstico (entrenamiento – configuración serie-paralelo y predicción varios pasos adelante – configuración paralela) se calculan las predicciones 6 pasos adelante, los resultados se muestran en las figura 6.88.

La figura 6.88 muestra el gráfico del pronóstico ampliado, se puede notar un mejor pronóstico que en la regularización Bayesiana (figura 6.83).

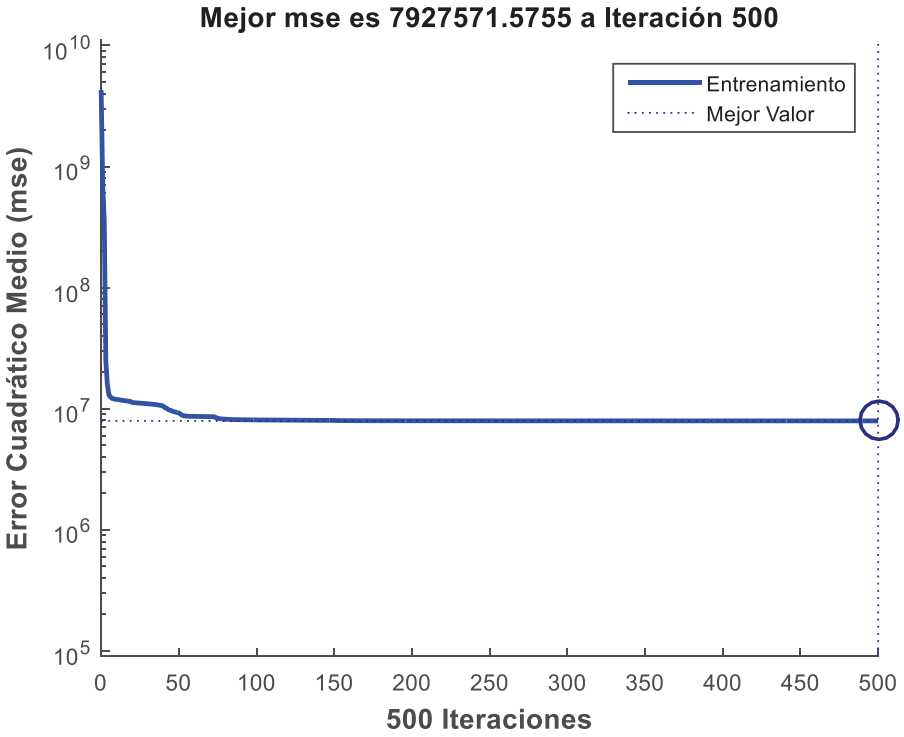


Figura 6.86 MSE Algoritmo Levenberg-Marquardt

(Gráfico Obtenido con el programa Matlab ver 15)

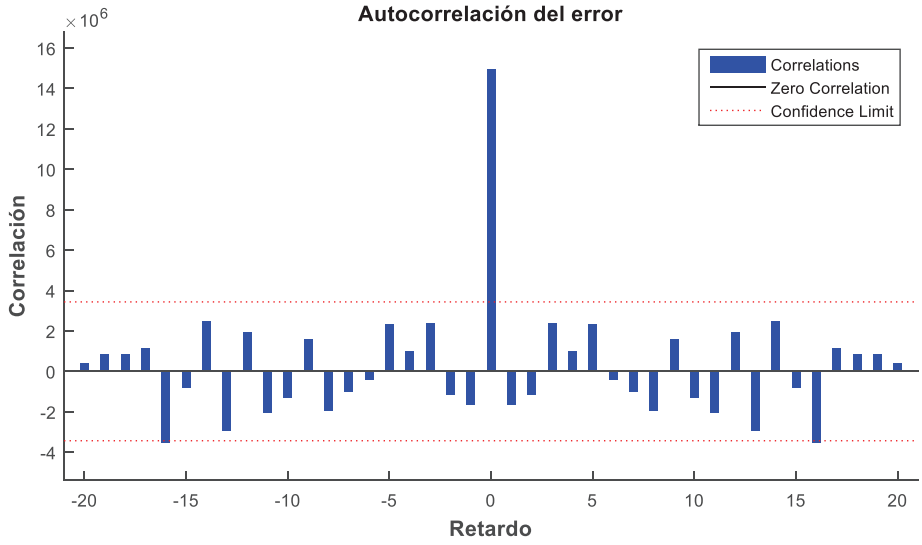


Figura 6.87 Autocorrelación del error con Levenberg-Marquardt

(Gráfico Obtenido con el programa Matlab ver 15)

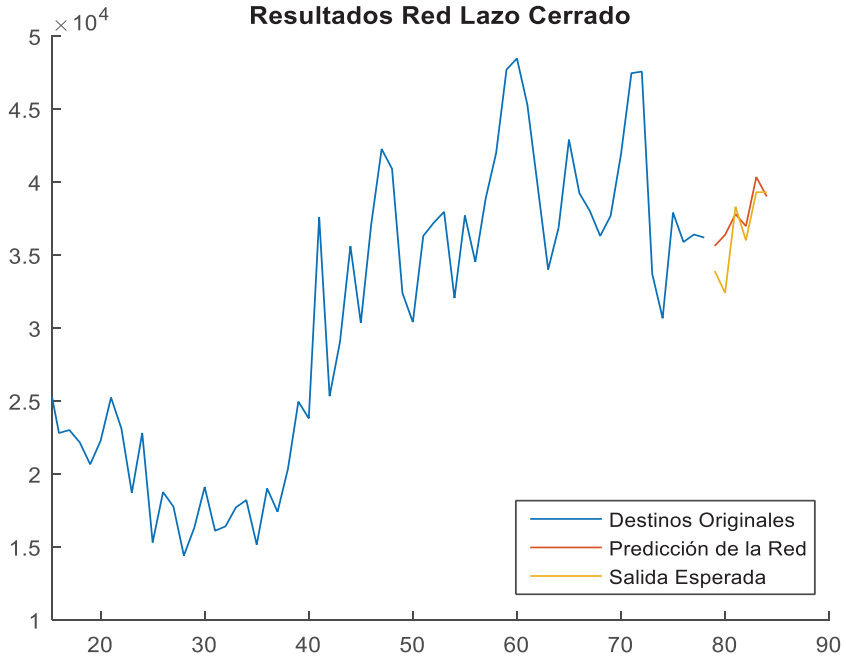


Figura 6.88 Pronóstico 6 pasos adelante configuración Paralelo con LM

(Gráfico Obtenido con el programa Matlab ver 15)

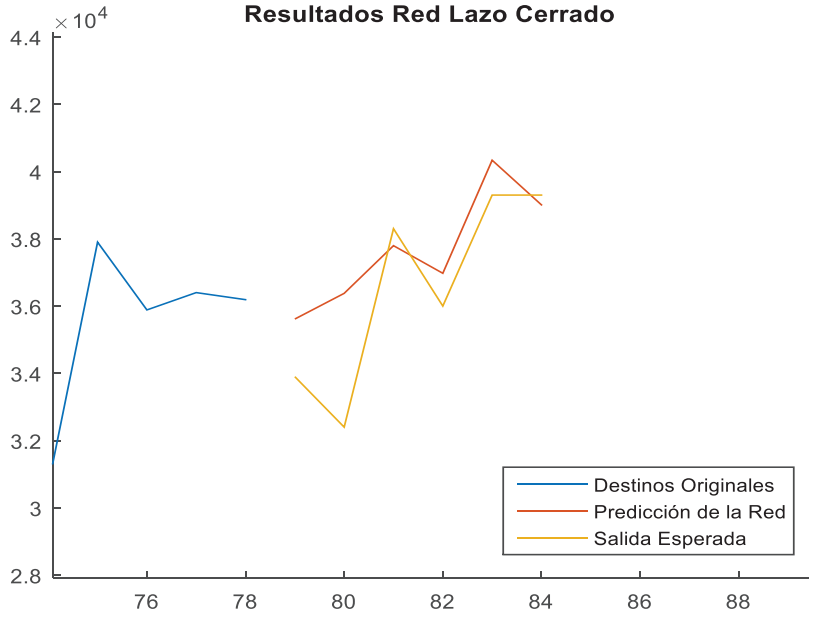


Figura 6.89 Pronóstico 6 pasos adelante configuración Paralelo con LM (Ampliado)

(Gráfico Obtenido con el programa Matlab ver 15)

Efectivamente, después de comparar las figuras 6.84 y 6.89 en este caso también el entrenamiento con el algoritmo de Levenberg-Marquardt produce un mejor pronóstico.

Al igual que con los dos métodos anteriores, Holt – Winters y Box-Jenkins, con cada uno de los entrenamientos arriba indicados se realizará el pronóstico, se calcularán los errores y se elegirá el que presente el menor error.

En las tablas 6.3 y 6.4 se muestran los pronósticos y sus errores de una Red Narnet con los distintos tipos de entrenamiento.

Tabla 6.3 Pronósticos y Errores Regularización Bayesiana

Mes	**Pronóstico Narnet**	Reales	Error	(Error)^2	PEt	ABS PEt
79	38605	33900	-4705	22137025	-13.88%	13.88%
80	40319	32400	-7919	62710561	-24.44%	24.44%
81	40892	38300	-2592	6718464	-6.77%	6.77%
82	40318	36000	-4318	18645124	-11.99%	11.99%
83	39311	39300	-11	121	-0.03%	0.03%
84	38864	39300	436	190096	1.11%	1.11%
			MSE	18400231.8	**MAPE**	**9.70%**

Tabla 6.4 Pronósticos y Errores Levenberg-Marquardt

Mes	**Pronóstico Narnet**	Valores Reales	Error	(Error)^2	PEt	ABS PEt
79	35614	33900	-1714	2937796	-5.06%	5.06%
80	36378	32400	-3978	15824484	-12.28%	12.28%
81	37795	38300	505	255025	1.32%	1.32%
82	36974	36000	-974	948676	-2.71%	2.71%
83	40336	39300	-1036	1073296	-2.64%	2.64%
84	39008	39300	292	85264	0.74%	0.74%
			MSE	3520756.83	**MAPE**	**4.12%**

La red Narnet con entrenamiento Levenberg-Marquardt será la representante de la metodología de Redes Neuronales para la comparación final con el resto de métodos analizados en este trabajo.

7 RESULTADOS Y DISCUCIONES

7.1 RESULTADOS DE LOS TRES MÉTODOS DE PRONÓSTICOS

A continuación se presentan los mejores resultados de cada uno de los métodos de pronósticos tratados en la presente investigación. Con cada uno de los métodos se realizó la predicción hasta seis pasos adelante.

7.1.1 MÉTODO DE HOLT-WINTERS

Tabla 7.1 Mejor Predicción con el Método de Holt-Winters

Mes	Pronóstico Holt - Winters	Reales	Error	(Error)^2	PEt	ABS PEt
79	34825	33900	-925	855625	-2.73%	2.73%
80	34427	32400	-2027	4108729	-6.26%	6.26%
81	34863	38300	3437	11812969	8.97%	8.97%
82	37192	36000	-1192	1420864	-3.31%	3.31%
83	37599	39300	1701	2893401	4.33%	4.33%
84	38832	39300	468	219024	1.19%	1.19%
			MSE	3551768.67	**MAPE**	**4.46%**
			RMSE	1884.61		

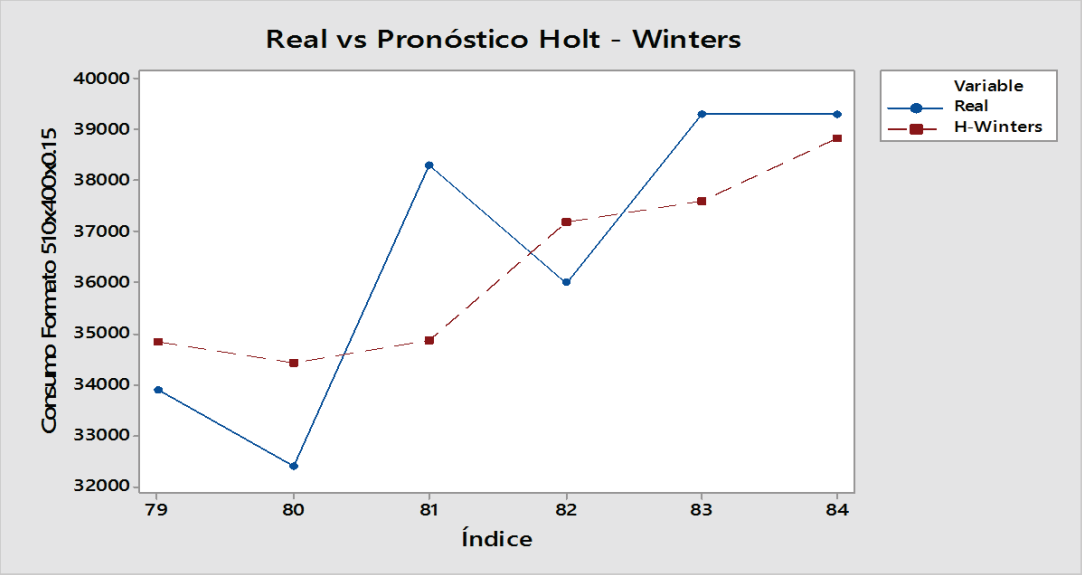


Figura 7.1 Pronóstico con el método de Holt – Winters vs Real

(Gráfico Obtenido con el programa Minitab ver 17)

7.1.2 MÉTODOLOGÍA DE BOX – JENKINS

Tabla 7.2 Mejor Predicción con el Método de Box-Jenkins

Mes	Pronóstico Box - Jenkins	Reales	Error	(Error)^2	PEt	ABS PEt
79	33835	33900	65	4225	0.19%	0.19%
80	34525	32400	-2125	4515625	-6.56%	6.56%
81	35761	38300	2539	6446521	6.63%	6.63%
82	37905	36000	-1905	3629025	-5.29%	5.29%
83	39094	39300	206	42436	0.52%	0.52%
84	38200	39300	1100	1210000	2.80%	2.80%
			MSE	2641305.33	**MAPE**	**3.67%**
			RMSE	1625.21		

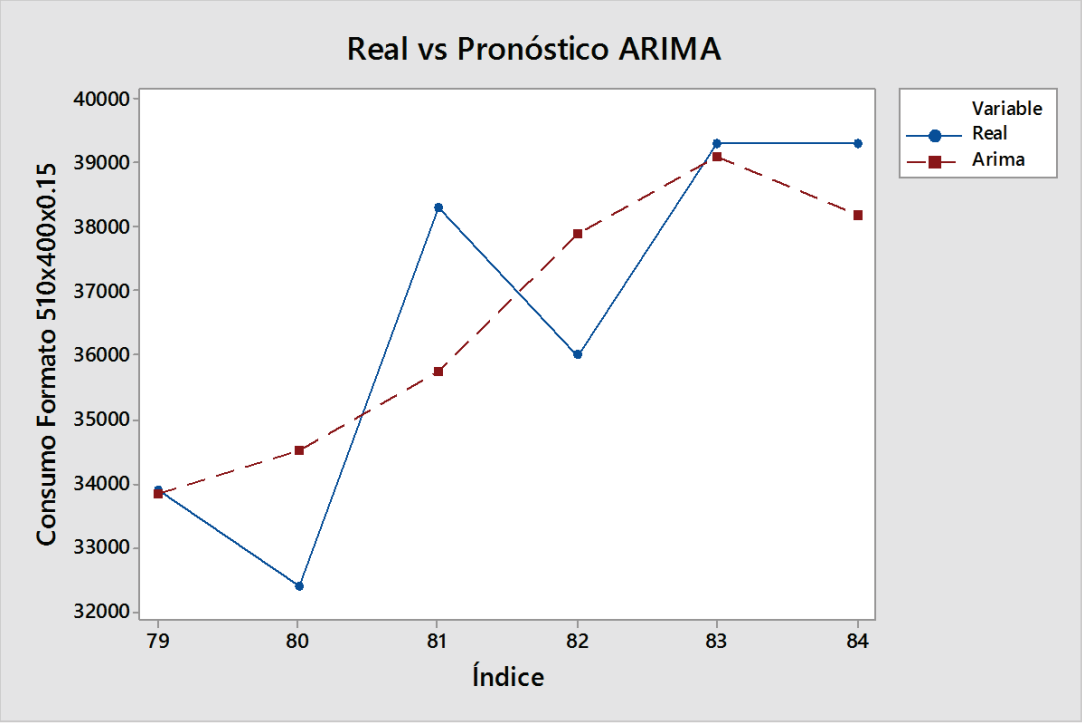


Figura 7.2 Pronóstico con la metodología ARIMA vs Real

(Gráfico Obtenido con el programa Minitab ver 17)

7.1.3 METODOLOGÍA DE REDES NEURONALES

Tabla 7.3 Mejor Predicción con el Método de Redes Neuronales

Mes	**Pronóstico Narnet**	Valores Reales	Error	(Error)^2	PEt	ABS PEt
79	35614	33900	-1714	2937796	-5.06%	5.06%
80	36378	32400	-3978	15824484	-12.28%	12.28%
81	37795	38300	505	255025	1.32%	1.32%
82	36974	36000	-974	948676	-2.71%	2.71%
83	40336	39300	-1036	1073296	-2.64%	2.64%
84	39008	39300	292	85264	0.74%	0.74%
			MSE	3520756.83	**MAPE**	**4.12%**
			RMSE	1876.37		

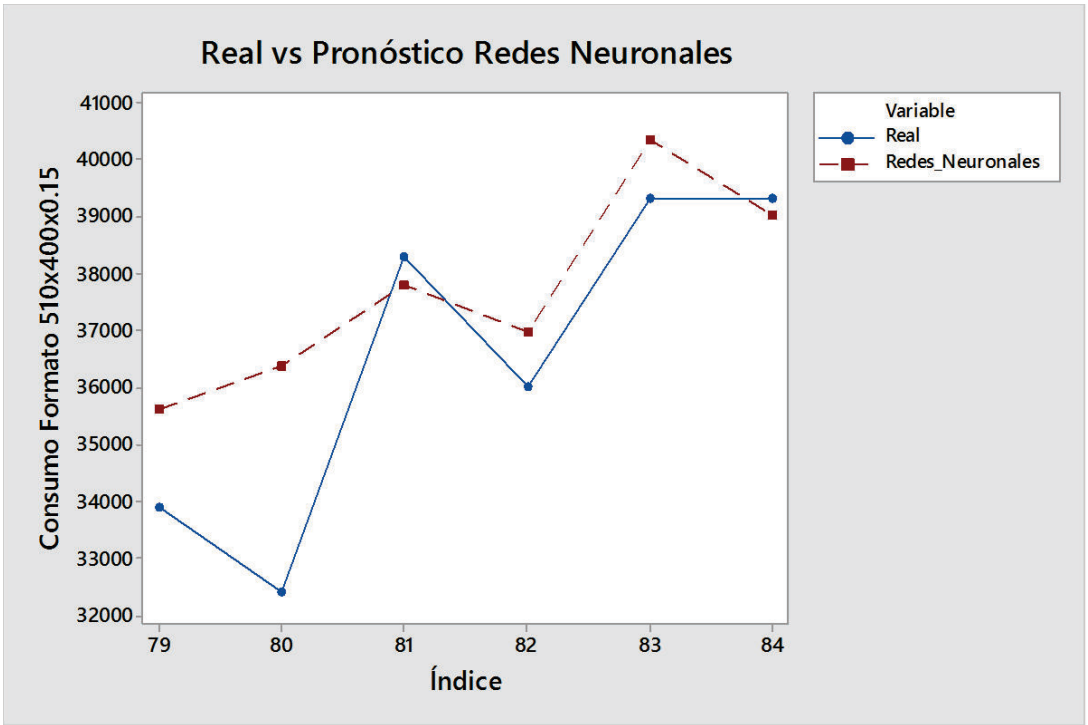


Figura 7.3 Pronóstico con Redes Neuronales vs Real

(Gráfico Obtenido con el programa Minitab ver 17)

En las tablas 7.1, 7.2 y 7.3 se resumen los mejores resultados de cada uno de los tres métodos de pronósticos tratados en esta investigación y en las figuras 7.1, 7.2 y 7.3 sus respectivos gráficos. Finalmente en la figura 7.4 se superponen los tres métodos de pronósticos versus los valores reales.

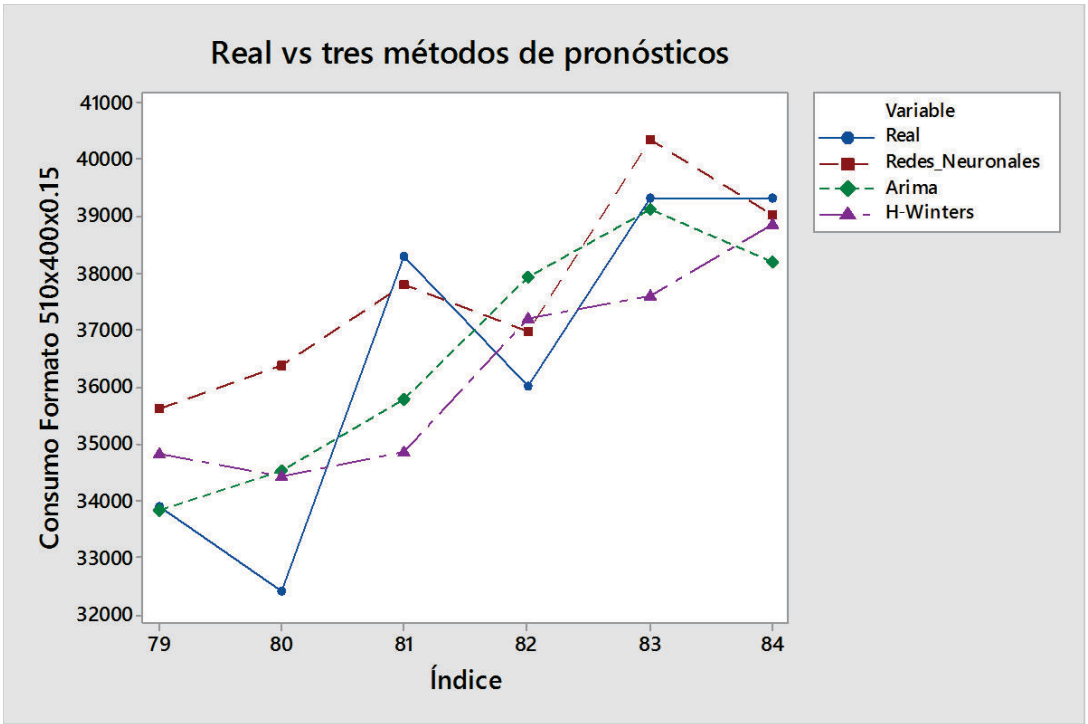


Figura 7.4 Pronóstico con los tres métodos vs Real

(Gráfico Obtenido con el programa Minitab ver 17)

El método que menor error ha producido es el de Box – Jenkins con un MAPE del 3.67%, por lo tanto para esta aplicación en particular, es el mejor método de pronóstico.

Después de analizar los errores MAPE y RMSE de los tres métodos de pronósticos, se puede garantizar que la predicción con cualquiera de ellos es muy confiable.

7.2 PREDICCIÓN DEL RESTO DE FORMATOS DE PLACAS DIGITALES CON LA METODOLOGÍA DE BOX – JENKINS.

Ya que la metodología ARIMA produjo el menor error de pronóstico MAPE, se realizará la predicción del resto de los formatos de placas con esta metodología.

7.2.1 PREDICCIÓN DEL FORMATO 525X450X0.15 (SM52)

En la tabla 7.4 se muestra la información del consumo de placas digitales formato 525x459x0.15 (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo SM_52, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utilizará el paquete estadístico Minitab ver 17, que ayudará en la identificación, estimación y diagnóstico de los modelos mediante la metodología ARIMA, con el modelo ARIMA adecuado se realizará el pronóstico respectivo, para luego calcular el error de pronóstico MAPE que ayudará a verificar la precisión del modelo.

Tabla 7.4 Consumo de Placas Formato 525x459x0.15 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	5400	4050	4950	3300	3500	3900	5700
2	3900	1950	12750	4850	3500	6500	7400
3	7850	6650	6550	5450	4200	6540	11400
4	2050	4450	2350	4000	4600	7099	6900
5	4250	5550	3000	5500	4600	6900	8700
6	2250	6260	5000	4950	4070	7099	7150
7	3900	6050	5350	3200	5100	5898	7400
8	1750	8008	8100	5400	4188	5292	7800
9	5550	6150	3400	4500	4698	6200	7200
10	3100	6900	5750	5300	5200	6700	8000
11	5950	7300	5300	4800	6400	6572	8500
12	8350	11700	3750	4900	5800	7200	7200

En la figura 7.5 se muestra el gráfico de la demanda de placas digitales (SM 52) desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Se modelará la serie con los datos hasta el primer semestre del año 2015 (78 observaciones) y se reservarán los datos del segundo semestre del año 2015 (6 observaciones) para poder verificar el error del pronóstico contra los valores reales.
A simple vista se puede ver que esta serie es No estacionaria en su media, ya que se observa un componente de tendencia no muy marcado pero un poco irregular.
Además no se puede observar ninguna estacionalidad marcada.
Cabe indicar que en la etapa posterior de estimación del modelo, también se pueden revelar ciertas características no tan claras en la etapa de identificación.

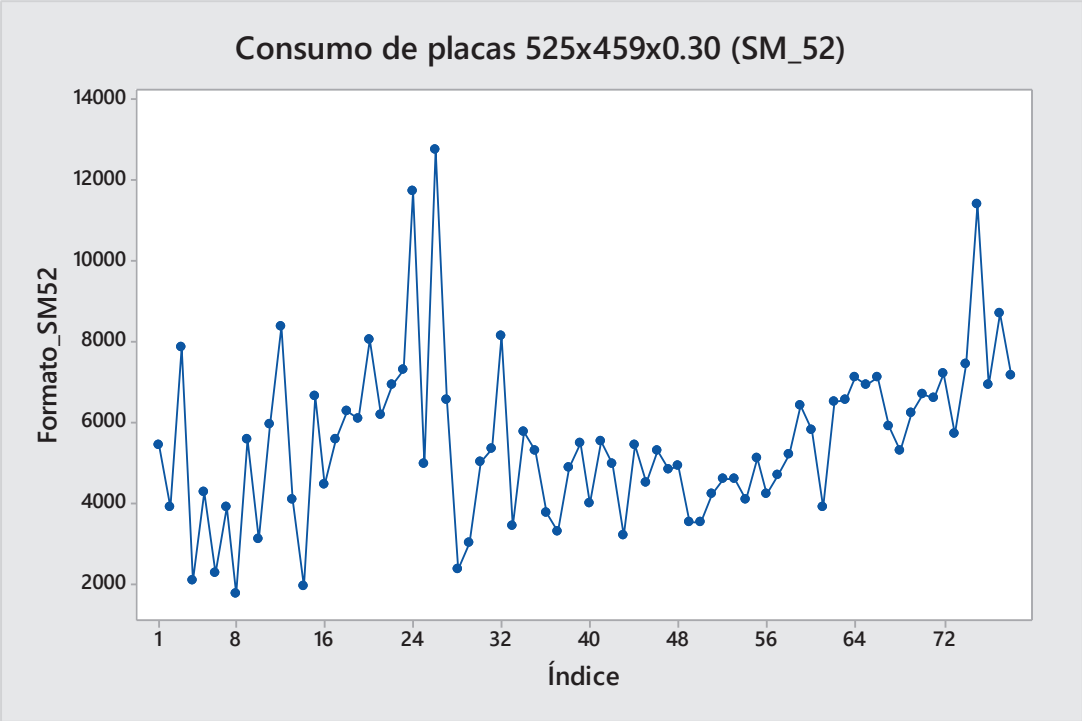


Figura 7.5 Consumo de placas digitales formato 525x459x0.15 (SM_52) (2009 – 2015)

Para verificar si la serie es No estacionaria, se va a correr la prueba de Dickey Fuller Aumentada (DFA), mediante el programa EViews 9. Los resultados se muestran en la tabla 7.5.
El estadístico t de la prueba de DFA se encuentra en la zona de No rechazo de la Hipótesis Nula, por lo tanto aceptamos la hipótesis nula que existe una raíz unitaria, que comprueba a su vez que la serie es no estacionaria, cosa que a la vista se

verifica claramente, la etapa de estimación confirmará esta suposición de no estacionariedad..

Las figuras 7.6 y 7.7 son gráficos de las funciones de autocorrelación y autocorrelación parcial (sacf y spacf).

Solamente como ejemplo, se supone que no detectamos la necesidad de aplicar una diferencia regular, entonces se tratará de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas sacf y spacf, para luego verificar lo que pasa en la etapa de identificación.

Las dos funciones sacf y spacf tienen picos significativos (sobrepasan los límites de confianza), por lo tanto iniciaremos con un modelo ARMA(1,1).

Tabla 7.5 Prueba DFA del Consumo de Placas Formato 525x459x0.15

Null Hypothesis: SM_52 has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 1 (Fixed)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-3.937435	0.0151
Test critical values:	1% level	-4.083355	
	5% level	-3.470032	
	10% level	-3.161982	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(SM_52)
Method: Least Squares
Date: 10/24/16 Time: 22:18
Sample (adjusted): 3 78
Included observations: 76 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
SM_52(-1)	-0.563333	0.143071	-3.937435	0.0002
D(SM_52(-1))	-0.282911	0.113979	-2.482139	0.0154
C	2608.636	802.3156	3.251384	0.0017
@TREND("1")	13.82367	10.42306	1.326258	0.1889

R-squared	0.442102	Mean dependent var	42.76316
Adjusted R-squared	0.418856	S.D. dependent var	2500.124
S.E. of regression	1905.914	Akaike info criterion	17.99451
Sum squared resid	2.62E+08	Schwarz criterion	18.11718
Log likelihood	-679.7913	Hannan-Quinn criter.	18.04353
F-statistic	19.01860	Durbin-Watson stat	2.000487
Prob(F-statistic)	0.000000		

(Gráfico Obtenido con el programa Eviews ver. 9)

En las figura 7.8 y 7.9 se muestran las funciones sacf y spacf de la serie después de aplicar un modelo ARMA(1,1).

Las dos funciones son satisfactorias, se podría afirmar que prácticamente se tiene ruido blanco en el residuo.

En la etapa de estimación se produjeron los resultados que se muestran en la tabla 7.6, mediante el programa Minitab ver 17.

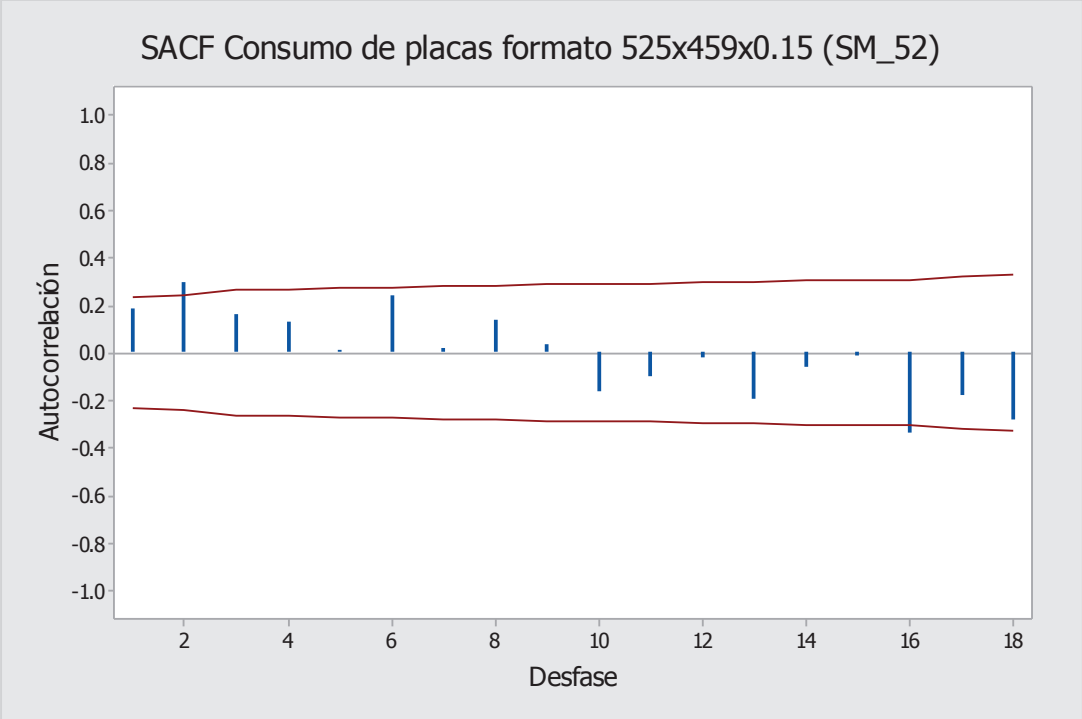


Figura 7.6 Autocorrelación Muestral del Consumo de placas digitales formato SM_52

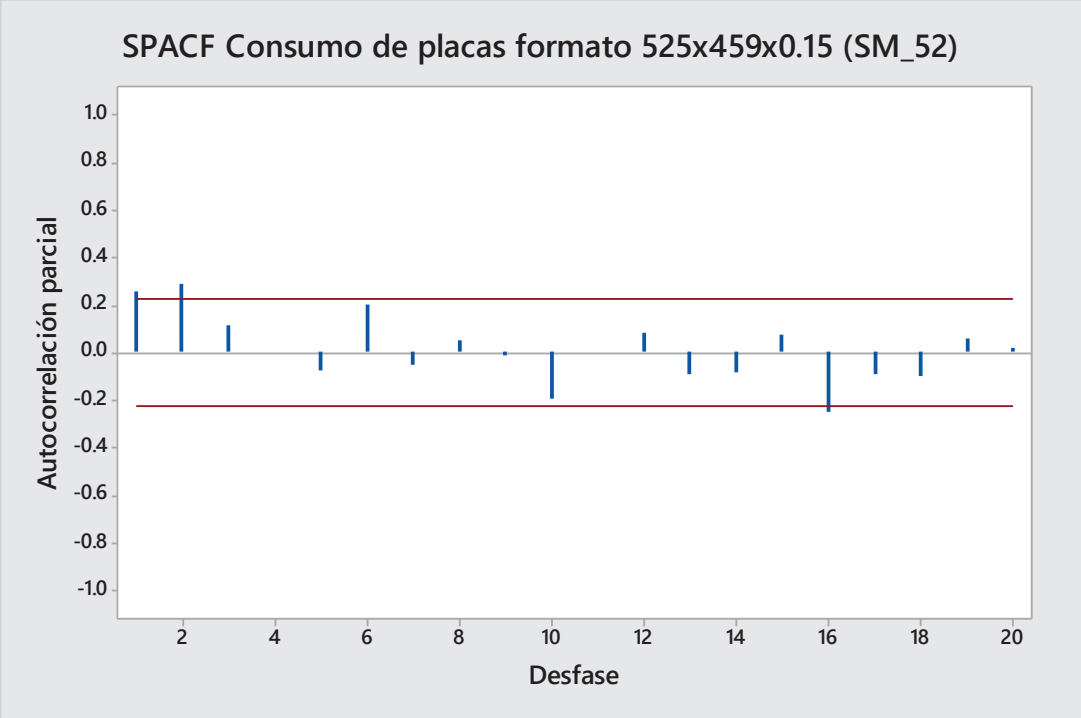


Figura 7.7 Autocorrelación Parcial del Consumo de placas digitales formato SM_52

Tabla 7.6 Estimación modelo ARMA (1,1) Formato 525x459x0.15 (SM_52)

```
Estimaciones finales de los parámetros

Tipo      Coef  SE Coef       T      P
AR   1  1.0013   0.0091  109.51  0.000
MA   1  0.7793   0.0758   10.28  0.000


Número de observaciones:  78
Residuos:    SC =  277697260 (se excluyeron pronósticos retrospectivos)
              MC =  3653911  GL = 76
```

En esta tabla se puede detectar un problema en el coeficiente $\emptyset_1$ (AR(1)) es mayor a uno, no cumple la condición de estacionariedad. Por lo tanto este modelo no es adecuado.

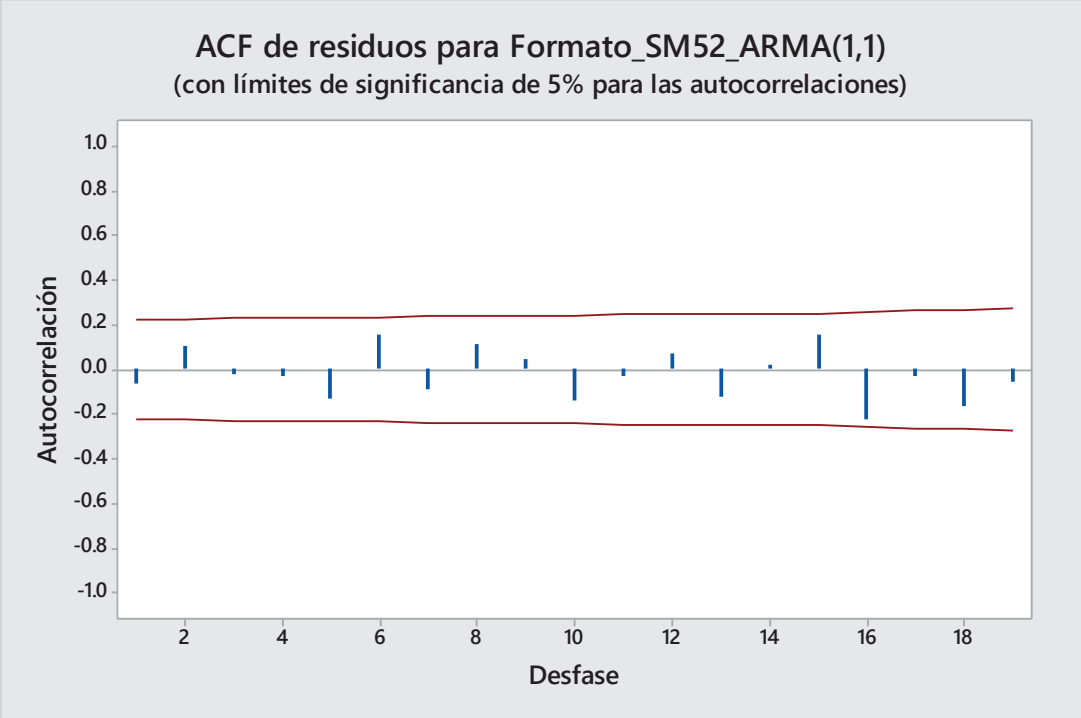


Figura 7.8 Autocorrelación residuos modelo ARMA(1,1), formato SM_52

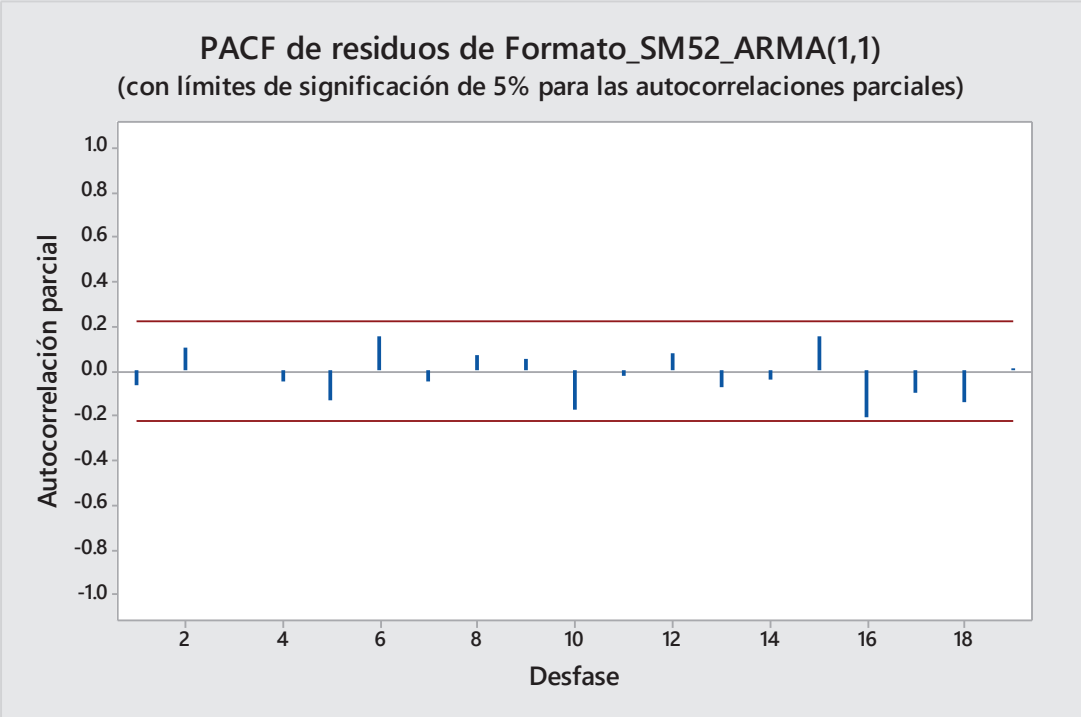


Figura 7.9 Autocorrelación parcial residuos modelo ARMA(1,1), formato SM_52

Además la tabla 7.6 sugiere la aplicación de una diferencia regular a la serie ya que el coeficiente autoregresivo AR(1) es prácticamente uno (1.0013).
En las figura 7.10 y 7.11 se muestran las funciones sacf y spacf de la serie después de aplicar un modelo ARIMA(0,1,1), es decir con una diferencia regular.
Las dos funciones son satisfactorias, se podría afirmar que prácticamente se tiene ruido blanco en el residuo.
En la etapa de estimación se produjeron los resultados que se muestran en la tabla 7.7, mediante el programa Minitab ver 17.

Tabla 7.7 Estimación modelo ARIMA (0,1,1) Formato 525x459x0.15 (SM_52)

```
Tipo      Coef  SE Coef      T      P
MA   1  0.7754   0.0731  10.61  0.000


Diferenciación: 1 Diferencia regular
Número de observaciones:  Serie original 78, después de diferenciar 77
Residuos:    SC =  277547943 (se excluyeron pronósticos retrospectivos)
             MC =  3651947  GL = 76
```

La tabla 7.7 ratifica que le modelo ARIMA (0,1,1) es adecuado, es decir es invertible $MA(1)$ menor a uno y significativo $(t > 2)$.

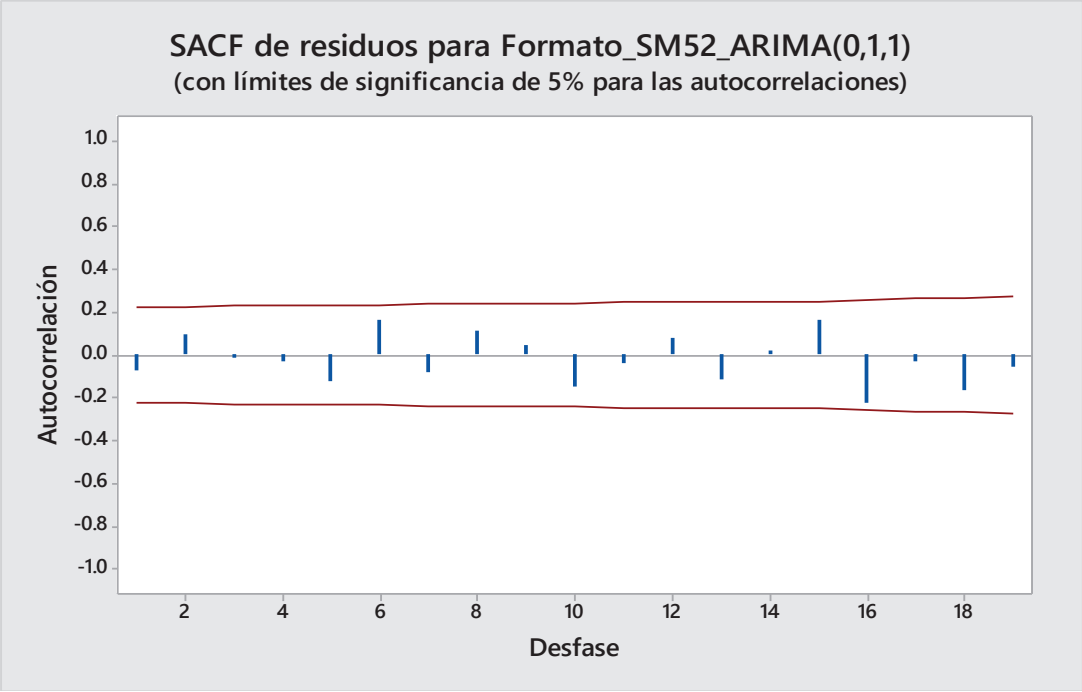


Figura 7.10 Autocorrelación residuos modelo ARIMA(0,1,1), formato SM_52

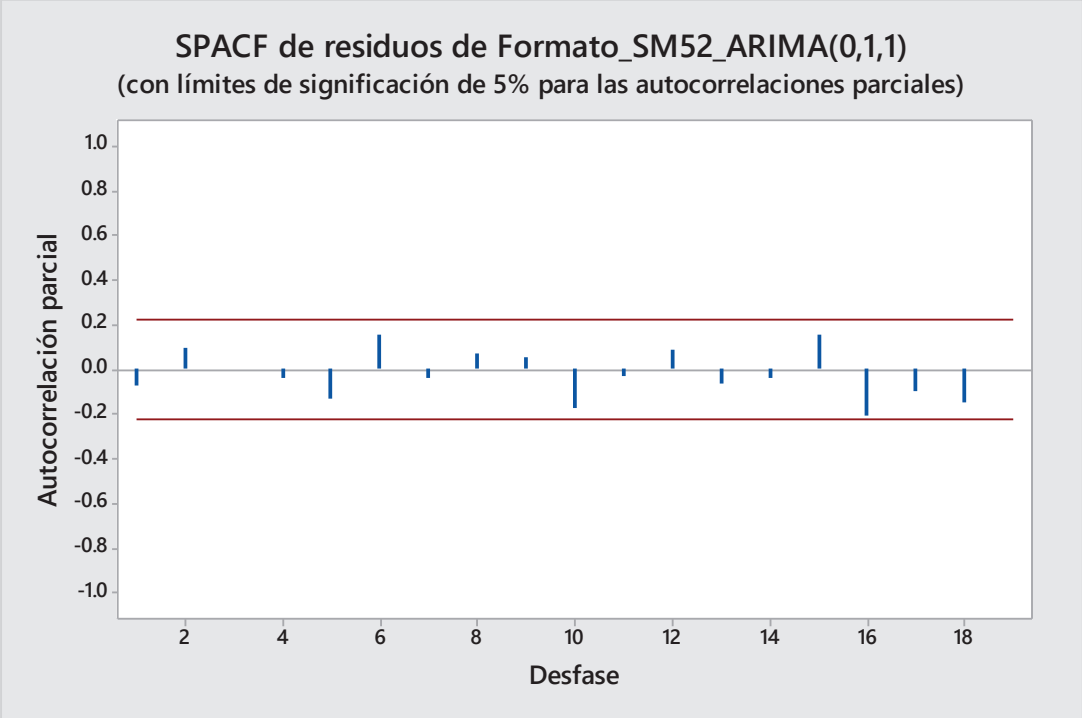


Figura 7.11 Autocorrelación parcial residuos modelo ARIMA(0,1,1), formato SM_52

Además en la tabla 7.8 se pueden ver los valores del estadístico Q^* de Ljung – Box, todos ellos se han chequeado con la tabla C2 (Distribución χ^2) del Anexo C y son adecuados (menor que los valores al 0.05).

Tabla 7.8 Estadístico de Ljung – Box Q^* para el Modelo ARIMA(0,1,1) SM_52

Estadística Chi-cuadrada modificada de Box-Pierce (Ljung-Box)				
Desfase	12	24	36	48
Chi-cuadrada	9.4	26.1	31.4	36.6
GL	11	23	35	47
Valor p	0.586	0.297	0.644	0.862

Después de chequear todos los criterios anteriores se puede concluir que el Modelo ARIMA(0,1,1) es estadísticamente adecuado.

El Modelo ARIMA (0,1,1) para el formato SM 52 quedaría:

$$(1-B)\tilde{Z}_t = (1-0.7754B)a_t \qquad (7.1)$$

Su pronóstico se presenta en la tabla 7.9.

Tabla 7.9 Pronósticos y errores Modelo ARIMA (0,1,1) SM_52

Mes	**Pronóstico Minitab ARIMA (0,1,1) Sin C**	Reales	Error	(Error)^2	PEt	ABS PEt
79	7618	7400	-218	47578.4456	-2.95%	2.95%
80	7618	7800	181.87516	33078.574	2.33%	2.33%
81	7618	7200	-418.12484	174828.381	-5.81%	5.81%
82	7618	8000	381.87516	145828.638	4.77%	4.77%
83	7618	8500	881.87516	777703.799	10.38%	10.38%
84	7618	7200	-418.12484	174828.381	-5.81%	5.81%
			MSE	**225641.037**	**MAPE**	**5.34%**
			RMSE	**475.02**		

La tabla 7.9 indica que el error MAPE no es considerable, por lo tanto se puede concluir que el pronóstico es bastante bueno.

En la figura 7.12 se muestra el gráfico de los pronósticos y su intervalo de predicción con este modelo.

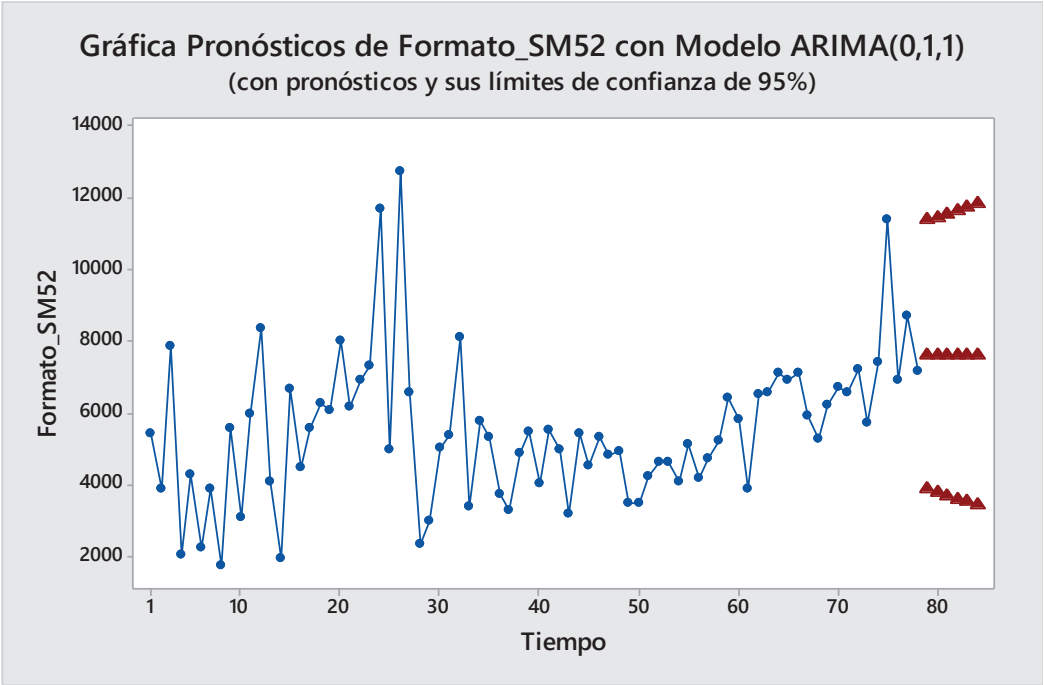


Figura 7.12 Pronósticos con modelo ARIMA(0,1,1), formato SM_52

(Gráfico Obtenido con el programa Minitab ver 17

7.2.2 PREDICCIÓN FORMATO 650X550X0.30 (KORD-MO)

En la tabla 7.10 se muestra la información del consumo de placas digitales formato 650x550x0.30 (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo KORD o MO, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utilizará el paquete estadístico Eviews ver 9, que ayudará en la identificación, estimación y diagnóstico de los modelos mediante la metodología ARIMA, con el modelo ARIMA adecuado se realizará el pronóstico respectivo, para luego calcular el error de pronóstico MAPE que ayudará a verificar la precisión del modelo.

En la figura 7.13 se muestra el gráfico de la demanda de placas digitales (KORD MO) desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Tabla 7.10 Consumo de Placas Formato 650x550x0.30 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	3990	5250	2250	4900	8750	14800	2650
2	3210	5580	3150	6260	8800	11730	8700
3	5880	7400	3240	8580	11050	8250	7680
4	5580	6600	2910	8500	14406	9620	5500
5	4200	6180	5340	10150	9850	10900	8900
6	3720	5580	4860	5580	8405	8449	7100
7	6960	6000	3140	12540	11619	7795	7890
8	5400	6877	2400	10650	7447	9698	9350
9	5700	4680	3930	7350	9448	8550	9050
10	5790	4590	3030	8900	8425	7700	8650
11	5910	3390	2470	11300	15098	10646	9150
12	10020	4080	3480	9450	16750	8800	9283

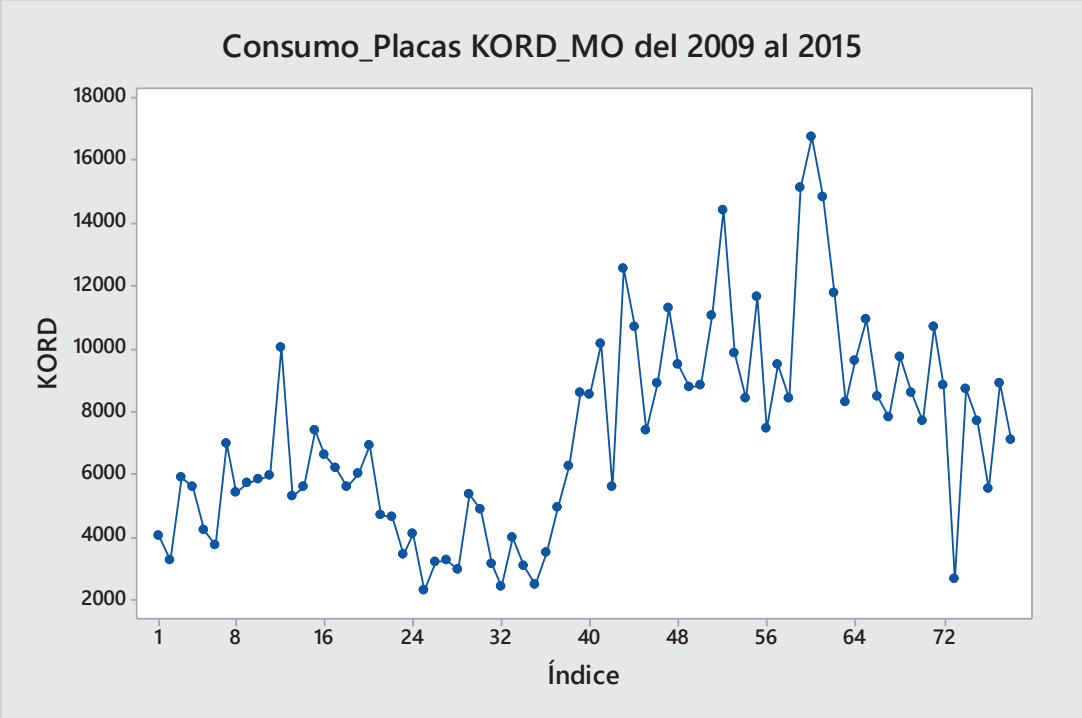


Figura 7.13 Consumo de placas formato 650x550x0.30 (KORD_MO) (2009 – 2015)

Se seguirá el mismo procedimiento seguido en formatos anteriores, es decir se apartarán las últimas 6 observaciones, para verificar el error del pronóstico contra los valores reales.

A simple vista parece que esta serie es No estacionaria en su media, ya que se observa un componente de tendencia irregular. No se detecta mayor cambio en la varianza.

Para comprobar si la serie es no estacionaria, se corre la prueba de Dickey Fuller Aumentada (DFA), mediante el programa EViews 9. Los resultados se muestran en la tabla 7.11.

El estadístico t de la prueba de DFA (-3.2075) se encuentra en la zona de no rechazo de la Hipótesis Nula (con un nivel de significancia del 5%), por lo tanto no se rechaza la hipótesis nula (existe una raíz unitaria), que comprueba a su vez que la serie es no estacionaria, cosa que a simple vista no se ve claramente, debido a la presencia del patrón estacional.

Tabla 7.11 Prueba DFA del Consumo de Placas Digitales Formato 650x550x0.30

Null Hypothesis: KORD_MO has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 1 (Fixed)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-3.207595	0.0907
Test critical values:	1% level	-4.083355	
	5% level	-3.470032	
	10% level	-3.161982	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(KORD_MO)
Method: Least Squares
Date: 10/25/16 Time: 23:44
Sample (adjusted): 3 78
Included observations: 76 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
KORD_MO(-1)	-0.341564	0.106486	-3.207595	0.0020
D(KORD_MO(-1))	-0.157084	0.117277	-1.339422	0.1846
C	1664.783	686.8510	2.423790	0.0179
@TREND("1")	22.36615	14.53924	1.538329	0.1284

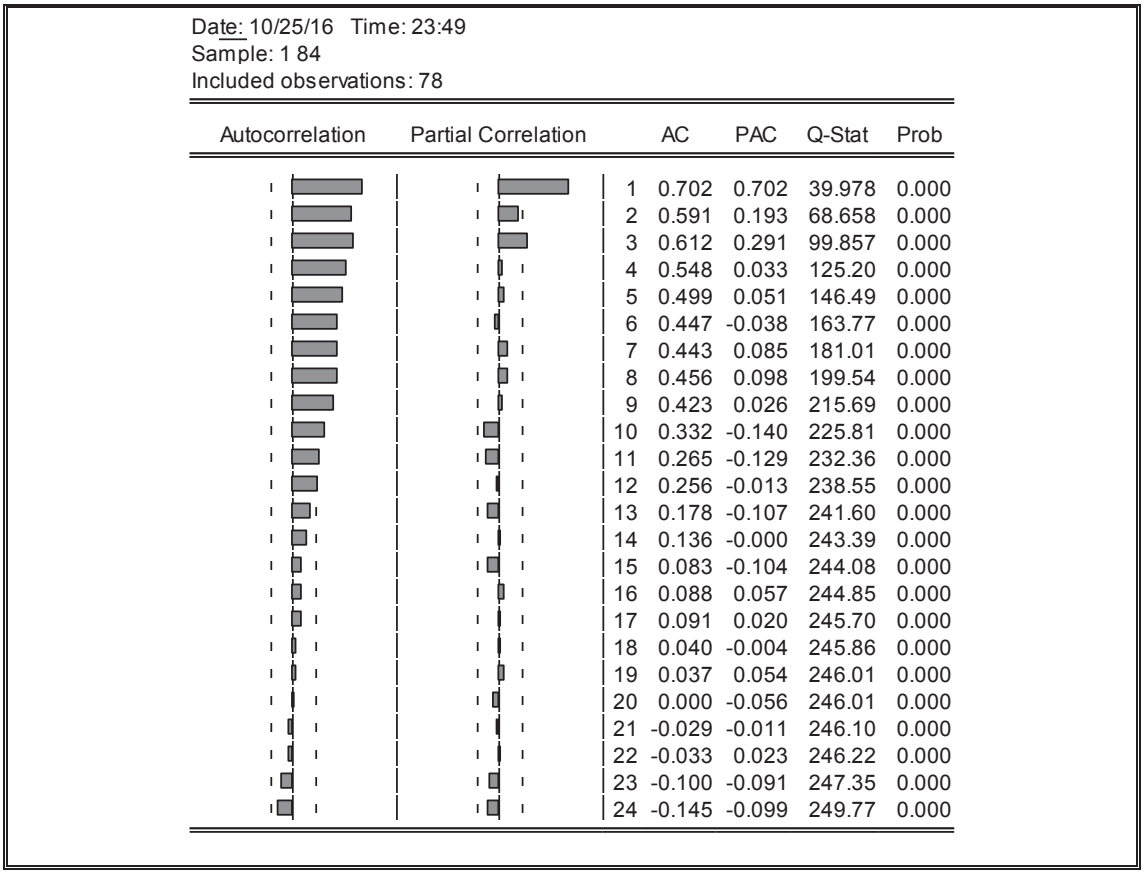
R-squared	0.220280	Mean dependent var	51.18421
Adjusted R-squared	0.187792	S.D. dependent var	2497.069
S.E. of regression	2250.424	Akaike info criterion	18.32682
Sum squared resid	3.65E+08	Schwarz criterion	18.44949
Log likelihood	-692.4192	Hannan-Quinn criter.	18.37585
F-statistic	6.780281	Durbin-Watson stat	2.049422
Prob(F-statistic)	0.000432		

(Tabla Obtenida con el programa Eviews ver 9

En la figura 7.14 se muestra el correlograma de la serie del consumo de placas digitales KORD_MO desde el año 2009 al 2015.

La función sacf tienen picos significativos en los rezagos 4,8,12 que advierten la presencia de un patrón estacional, por esta razón se aplicará una diferencia estacional, para después verificar si es necesaria una diferencia regular.

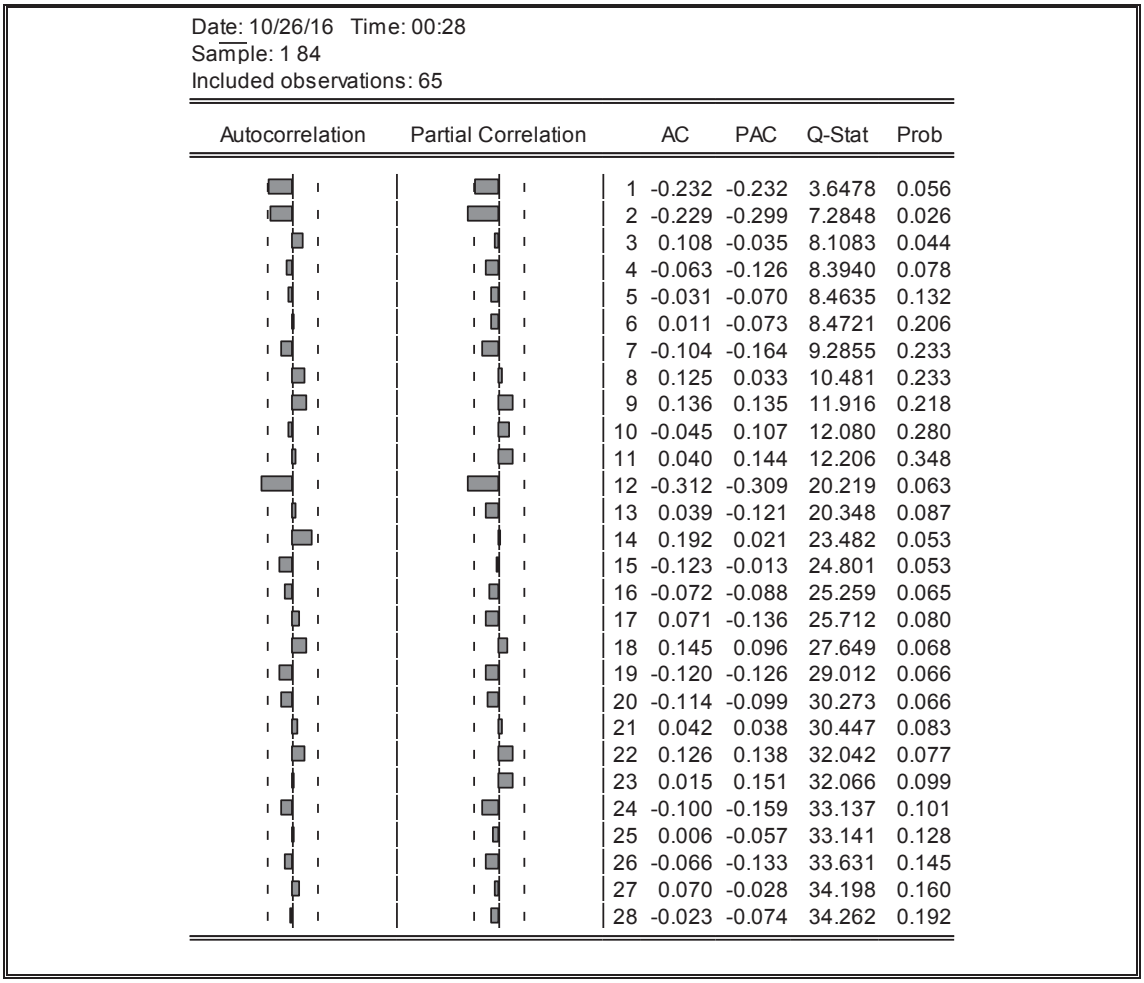
A continuación se va a realizar una diferencia estacional a la serie para ver si se aclara el correlograma y tratar de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas sacf y spacf.



	AC	PAC	Q-Stat	Prob
1	0.702	0.702	39.978	0.000
2	0.591	0.193	68.658	0.000
3	0.612	0.291	99.857	0.000
4	0.548	0.033	125.20	0.000
5	0.499	0.051	146.49	0.000
6	0.447	-0.038	163.77	0.000
7	0.443	0.085	181.01	0.000
8	0.456	0.098	199.54	0.000
9	0.423	0.026	215.69	0.000
10	0.332	-0.140	225.81	0.000
11	0.265	-0.129	232.36	0.000
12	0.256	-0.013	238.55	0.000
13	0.178	-0.107	241.60	0.000
14	0.136	-0.000	243.39	0.000
15	0.083	-0.104	244.08	0.000
16	0.088	0.057	244.85	0.000
17	0.091	0.020	245.70	0.000
18	0.040	-0.004	245.86	0.000
19	0.037	0.054	246.01	0.000
20	0.000	-0.056	246.01	0.000
21	-0.029	-0.011	246.10	0.000
22	-0.033	0.023	246.22	0.000
23	-0.100	-0.091	247.35	0.000
24	-0.145	-0.099	249.77	0.000

Figura 7.14 Correlograma del Consumo de placas digitales formato KORD_MO

La figura 7.15 muestra el correlograma de la primera diferencia estacional. Se puede notar que la serie ya es estacionaria (la función sacf decrece rápidamente a cero). Se puede observar un patrón auto regresivo (AR) en la parte no estacional, es decir existen dos picos en la función spacf y la función sacf decrece con un patrón sinusoidal, mientras la parte estacional presenta un patrón de medias móviles (MA), ya que existe un pico en el rezago 12 de la función sacf y en la función spacf el pico del rezago 12 va decreciendo en el rezago 24.



	AC	PAC	Q-Stat	Prob
1	-0.232	-0.232	3.6478	0.056
2	-0.229	-0.299	7.2848	0.026
3	0.108	-0.035	8.1083	0.044
4	-0.063	-0.126	8.3940	0.078
5	-0.031	-0.070	8.4635	0.132
6	0.011	-0.073	8.4721	0.206
7	-0.104	-0.164	9.2855	0.233
8	0.125	0.033	10.481	0.233
9	0.136	0.135	11.916	0.218
10	-0.045	0.107	12.080	0.280
11	0.040	0.144	12.206	0.348
12	-0.312	-0.309	20.219	0.063
13	0.039	-0.121	20.348	0.087
14	0.192	0.021	23.482	0.053
15	-0.123	-0.013	24.801	0.053
16	-0.072	-0.088	25.259	0.065
17	0.071	-0.136	25.712	0.080
18	0.145	0.096	27.649	0.068
19	-0.120	-0.126	29.012	0.066
20	-0.114	-0.099	30.273	0.066
21	0.042	0.038	30.447	0.083
22	0.126	0.138	32.042	0.077
23	0.015	0.151	32.066	0.099
24	-0.100	-0.159	33.137	0.101
25	0.006	-0.057	33.141	0.128
26	-0.066	-0.133	33.631	0.145
27	0.070	-0.028	34.198	0.160
28	-0.023	-0.074	34.262	0.192

Figura 7.15 Correlograma diferencia estacional del Consumo del formato KORD_MO

Después de lo dicho iniciaremos con un modelo estacional: $ARIMA(1,0,0)(0,1,1)_{12}$ que se lo denominará modelo1.

En la etapa de estimación se produjeron los resultados que se muestran en la tabla 7.12, mediante el programa Eviews ver 9. Se puede observar que los coeficientes del modelo 1 cumplen con las condiciones de estacionariedad e invertibilidad además son significativos. El correlograma del modelo 1 se muestra en la figura 7.16. En esta figura se puede notar un pico casi saliendo de los límites de confianza en el rezago 3, además en la tabla 7.12 el estadístico e Durbin Watson (DW) sale de los límites permitidos $(1.85 < DW < 2.15)$.

Tabla 7.12 Estimación del modelo 1

Dependent Variable: D(KORD_MO,0,12)
Method: ARMA Maximum Likelihood (BFGS)
Date: 10/25/16 Time: 22:15
Sample: 13 78
Included observations: 66
Convergence achieved after 12 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.745620	0.081198	9.182751	0.0000
MA(12)	-0.803945	0.223917	-3.590378	0.0006
SIGMASQ	5652733.	1196132.	4.725845	0.0000

R-squared	0.649482	Mean dependent var	414.5303
Adjusted R-squared	0.638354	S.D. dependent var	4046.589
S.E. of regression	2433.498	Akaike info criterion	18.66574
Sum squared resid	3.73E+08	Schwarz criterion	18.76527
Log likelihood	-612.9693	Hannan-Quinn criter.	18.70506
Durbin-Watson stat	2.257518		

Inverted AR Roots	.75			
Inverted MA Roots	.98	.85+.49i	.85-.49i	.49-.85i
	.49+.85i	.00+.98i	-.00-.98i	-.49-.85i
	-.49+.85i	-.85+.49i	-.85-.49i	-.98

(Tabla Obtenida con el programa Eviews ver 9

Date: 10/26/16 Time: 00:56
Sample: 1 84
Included observations: 66
Q-statistic probabilities adjusted for 2 ARMA terms

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.164	-0.164	1.8511	
		2	-0.112	-0.143	2.7332	
		3	0.233	0.198	6.6012	0.010
		4	-0.021	0.040	6.6331	0.036
		5	-0.007	0.046	6.6373	0.084
		6	0.110	0.077	7.5371	0.110
		7	-0.075	-0.052	7.9644	0.158
		8	0.074	0.069	8.3827	0.211
		9	0.208	0.201	11.791	0.108
		10	0.017	0.137	11.813	0.160
		11	-0.045	-0.010	11.978	0.215
		12	-0.030	-0.137	12.053	0.281
		13	-0.010	-0.097	12.062	0.359
		14	0.060	0.025	12.378	0.416
		15	-0.082	-0.063	12.976	0.450
		16	-0.119	-0.140	14.255	0.431
		17	0.021	-0.112	14.295	0.503
		18	0.051	-0.016	14.536	0.559
		19	-0.054	-0.021	14.818	0.609
		20	-0.071	-0.045	15.304	0.641
		21	-0.068	-0.061	15.771	0.672
		22	0.123	0.155	17.313	0.633
		23	-0.066	0.006	17.767	0.664
		24	-0.174	-0.117	20.993	0.521

Figura 7.16 Correlograma Modelo 1

Esto significa que podemos mejorar el modelo 1. Para esto aumentamos un coeficiente MA(3), a este nuevo modelo se lo llamará modelo 2.
En la tabla 7.13 se muestran los resultados de la estimación con el modelo 2 y en la figura 7.17 su correlograma.

Tabla 7.13 Estimación modelo 2

Dependent Variable: D(KORD_MO,0,12)
Method: ARMA Maximum Likelihood (BFGS)
Date: 10/25/16 Time: 23:17
Sample: 13 78
Included observations: 66
Convergence achieved after 7 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	0.704526	0.088308	7.978025	0.0000
MA(3)	0.161063	0.148235	1.086540	0.2814
MA(12)	-0.747631	0.182299	-4.101123	0.0001
SIGMASQ	5614039.	1034505.	5.426786	0.0000

R-squared	0.651881	Mean dependent var	414.5303
Adjusted R-squared	0.635036	S.D. dependent var	4046.589
S.E. of regression	2444.634	Akaike info criterion	18.66707
Sum squared resid	3.71E+08	Schwarz criterion	18.79978
Log likelihood	-612.0134	Hannan-Quinn criter.	18.71951
Durbin-Watson stat	2.141540		

Inverted AR Roots	.70			
Inverted MA Roots	.96	.84+.50i	.84-.50i	.50-.86i
	.50+.86i	.01+.98i	.01-.98i	-.48-.83i
	-.48+.83i	-.85-.48i	-.85+.48i	-.99

El coeficiente de MA(3) no es significativo, pero se lo dejará como parte del modelo, ya que mejora la correlación de los residuos (DW=2.14) y además el error de pronóstico, mismo que se mostrará más adelante.
En el correlograma se puede ver un pico en el rezago 9 que casi sale de los límites de confianza, después de varias simulaciones aumentando un coeficiente MA(9), una constante al modelo, se puede concluir que el modelo que menor error de pronóstico produce es el modelo 2.
En la figura 7.18 se muestra las raíces inversas del modelo 2, todas están dentro del círculo unitario, por lo tanto se cumplen las condiciones de estacionariedad e invertibilidad del modelo 2.

Date: 10/25/16 Time: 23:18
Sample: 1 84
Included observations: 66
Q-statistic probabilities adjusted for 3 ARMA terms

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.095	-0.095	0.6247	
		2	-0.074	-0.083	1.0048	
		3	0.122	0.109	2.0718	
		4	0.030	0.048	2.1370	0.144
		5	0.005	0.030	2.1389	0.343
		6	0.066	0.063	2.4631	0.482
		7	-0.086	-0.083	3.0198	0.555
		8	0.097	0.086	3.7414	0.587
		9	0.246	0.247	8.5078	0.203
		10	0.017	0.103	8.5312	0.288
		11	-0.040	-0.011	8.6634	0.371
		12	-0.059	-0.139	8.9547	0.441
		13	-0.003	-0.070	8.9558	0.536
		14	0.071	0.043	9.3949	0.585
		15	-0.149	-0.135	11.348	0.499
		16	-0.126	-0.143	12.783	0.465
		17	0.010	-0.115	12.793	0.543
		18	0.067	-0.004	13.217	0.586
		19	-0.077	-0.067	13.783	0.615
		20	-0.077	-0.058	14.360	0.642
		21	-0.053	-0.013	14.641	0.686
		22	0.110	0.144	15.873	0.666
		23	-0.046	0.014	16.090	0.711
		24	-0.206	-0.130	20.645	0.481

Figura 7.17 Correlograma Modelo 2

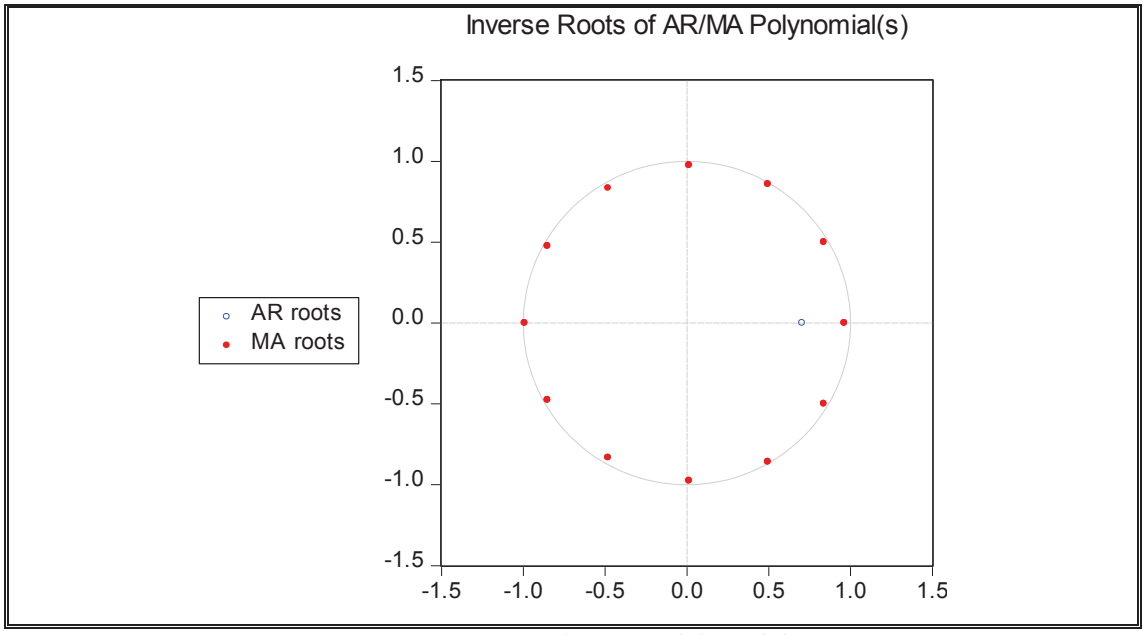


Figura 7.18 Raíces inversas del modelo 2

Después de chequear todos los criterios anteriores se puede concluir que el Modelo 2 es estadísticamente adecuado.

El Modelo 2 para el formato MO_KORD quedaría:

$$(1 - 0.7045B)(1 - B^{12})\tilde{Z}_t = (1 - 0.1610B^3 - 0.7476B^{12})a_t \qquad (7.5)$$

Su pronóstico se presenta en la tabla 7.14.

La tabla 7.14 indica que el error MAPE es aceptable, por lo tanto se puede concluir que el pronóstico es bastante bueno.

Tabla 7.14 Pronósticos y errores modelo 2

Mes	**Pronóstico Eviews ARIMA Modelo 2**	Reales	Error	(Error)^2	PEt	ABS PEt
79	8648.35861	7890	-758	575107.779	-9.61%	9.61%
80	8111.52574	9350	1238.47426	1533818.49	13.25%	13.25%
81	7646.36545	9050	1403.63455	1970189.94	15.51%	15.51%
82	7084.63065	8650	1565.36935	2450381.21	18.10%	18.10%
83	9602.60156	9150	-452.601564	204848.176	-4.95%	4.95%
84	9870.57376	9283	-587.573755	345242.918	-6.33%	6.33%
			MSE	**1179931.42**	**MAPE**	**11.29%**
			RMSE	**1086.25**		

7.2.3 PREDICCIÓN FORMATO 754X605X0.30 (PM74)

En la tabla 7.15 se muestra la información del consumo de placas digitales formato 745x605x0.30, correspondiente a una prensa modelo PM_74, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utilizará el paquete estadístico Eviews ver 9, que ayudará en la identificación, estimación y diagnóstico de los modelos mediante la metodología ARIMA, con el modelo ARIMA adecuado se realizará el pronóstico respectivo, para luego calcular el error de pronóstico MAPE que ayudará a verificar la precisión del modelo.

Tabla 7.15 Consumo de Placas Formato 745x605x0.30 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	2130	750	4680	1550	3650	4650	6350
2	1710	2160	7920	4060	4200	4200	8500
3	3240	2460	4860	5420	4040	3400	9950
4	840	2820	4830	4600	4500	5047	8240
5	750	3150	3180	4950	3950	6950	6440
6	3600	3900	3240	3350	3350	5680	7690
7	3540	3180	4230	6160	2770	5000	8700
8	3510	3500	9810	5000	4050	8050	7200
9	7530	3360	4940	3950	3950	5830	9450
10	3180	5190	4170	1700	3050	5485	8150
11	6360	3660	4950	5000	3350	7471	9800
12	9390	7950	3560	3460	4250	7600	7550

En la figura 7.19 se muestran el gráfico de la demanda de placas digitales (PM 74) desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Se modelará la serie con los datos hasta el primer semestre del año 2015 (78 observaciones) y se reservarán los datos del segundo semestre del año 2015 (6 observaciones) para poder verificar el error del pronóstico contra los valores reales.

A simple vista se puede ver que esta serie es no estacionaria en su media, ya que se observa un componente de tendencia marcado e irregular. También se puede observar que la varianza tiene una variación algo pronunciada. No se puede observar ninguna estacionalidad marcada.

Para verificar si la serie es no estacionaria, se va a correr la prueba de Dickey Fuller Aumentada (DFA), mediante el programa EViews 9. Los resultados se muestran en la tabla 7.16.

El estadístico t de la prueba de DFA se encuentra en la zona de no rechazo de la Hipótesis Nula, es decir aceptamos la hipótesis nula que existe una raíz unitaria, por lo tanto la serie es no estacionaria, cosa que a la vista se verifica claramente.

En la figura 7.20 se muestra el correlograma de la serie (funciones de autocorrelación y autocorrelación parcial (sacf y spacf)). Este gráfico ratifica la no estacionariedad de la serie en su media, es decir la función sacf decae lentamente a niveles no significativos.

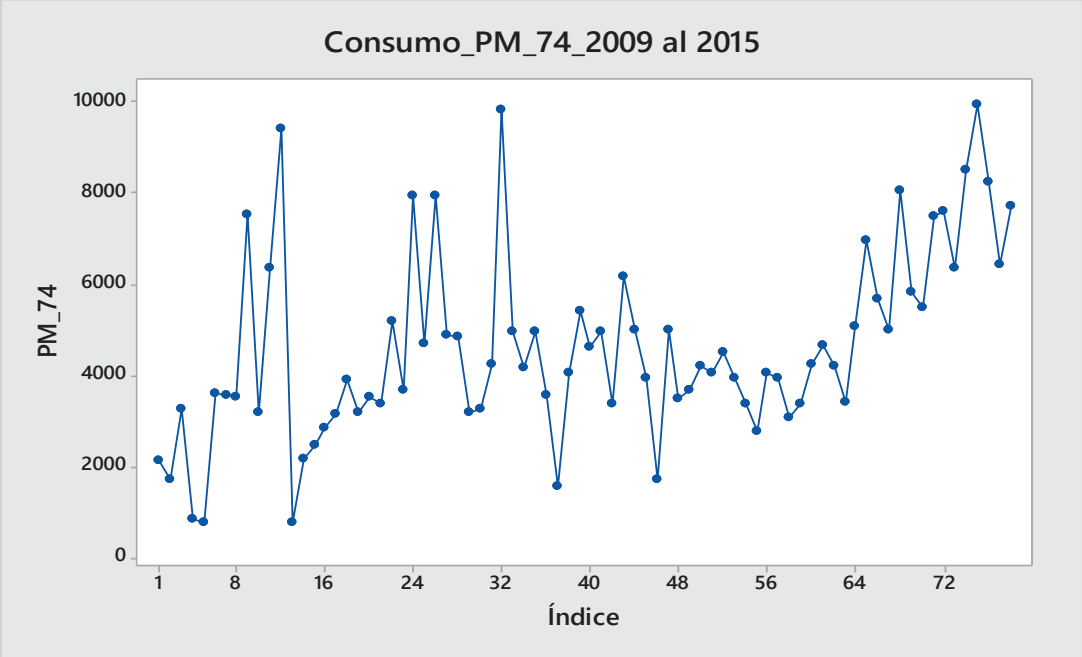


Figura 7.19 Consumo de placas formato 745x605x0.30 (PM_74) (2009 – 2015)

(Gráfico Obtenido con el programa Minitab ver 17)

Date: 10/24/16 Time: 21:20
Sample: 1 84
Included observations: 78

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.428	0.428	14.860	0.000
		2	0.341	0.193	24.397	0.000
		3	0.360	0.203	35.196	0.000
		4	0.194	-0.060	38.372	0.000
		5	0.041	-0.157	38.513	0.000
		6	0.169	0.148	41.000	0.000
		7	0.051	-0.047	41.231	0.000
		8	0.117	0.146	42.448	0.000
		9	0.105	-0.025	43.448	0.000
		10	0.027	-0.087	43.514	0.000
		11	-0.019	-0.064	43.548	0.000
		12	0.085	0.097	44.236	0.000
		13	0.001	0.021	44.236	0.000
		14	-0.010	-0.033	44.246	0.000
		15	0.055	0.030	44.547	0.000
		16	-0.057	-0.151	44.877	0.000
		17	-0.021	0.076	44.922	0.000
		18	-0.117	-0.187	46.343	0.000
		19	-0.172	-0.039	49.459	0.000
		20	-0.098	0.050	50.483	0.000
		21	-0.052	0.041	50.775	0.000
		22	-0.187	-0.106	54.656	0.000
		23	-0.055	0.009	54.999	0.000
		24	-0.079	-0.033	55.718	0.000

Figura 7.20 Correlograma del Consumo del formato PM_74 del 2009 al 2015

(Gráfico Obtenido con el programa Eviews ver 9

Tabla 7.16 Prueba DFA del Consumo de Placas Digitales Formato 745x605x0.30

Null Hypothesis: PM_74 has a unit root
Exógenous: Constant, Linear Trend
Lag Length: 2 (Fixed)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-3.076570	0.1195
Test critical values:	1% level	-4.085092	
	5% level	-3.470851	
	10% level	-3.162458	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(PM_74)
Method: Least Squares
Date: 10/24/16 Time: 22:02
Sample (adjusted): 4 78
Included observations: 75 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
PM_74(-1)	-0.476751	0.154962	-3.076570	0.0030
D(PM_74(-1))	-0.282418	0.144956	-1.948305	0.0554
D(PM_74(-2))	-0.194513	0.118238	-1.645095	0.1044
C	1531.241	642.3878	2.383671	0.0199
@TREND("1")	19.27423	11.07688	1.740042	0.0862

R-squared	0.380437	Mean dependent var	59.33333
Adjusted R-squared	0.345033	S.D. dependent var	2191.510
S.E. of regression	1773.589	Akaike info criterion	17.86374
Sum squared resid	2.20E+08	Schwarz criterion	18.01824
Log likelihood	-664.8902	Hannan-Quinn criter.	17.92543
F-statistic	10.74570	Durbin-Watson stat	1.959523
Prob(F-statistic)	0.000001		

(Tabla Obtenida con el programa Eviews ver 9

En la figura 7.21 se muestra el gráfico de la serie después de la primera diferencia. La serie ya es estacionaria alrededor de su media, no hace falta otra diferencia.

Pero la varianza que se puede observar en la fig. 7.21, de la serie con diferencia regular, es bastante pronunciada, que podría ocasionar un problema de no estacionariedad en la varianza. Para evitar esto se tomará el logaritmo natural de la serie, cuyo gráfico se muestra en la fig.7.22.

Con la varianza más estable, a continuación se tratará de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas sacf y spacf, después de la diferencia regular y aplicación del logaritmo natural ya se puede asumir estacionariedad en la serie.

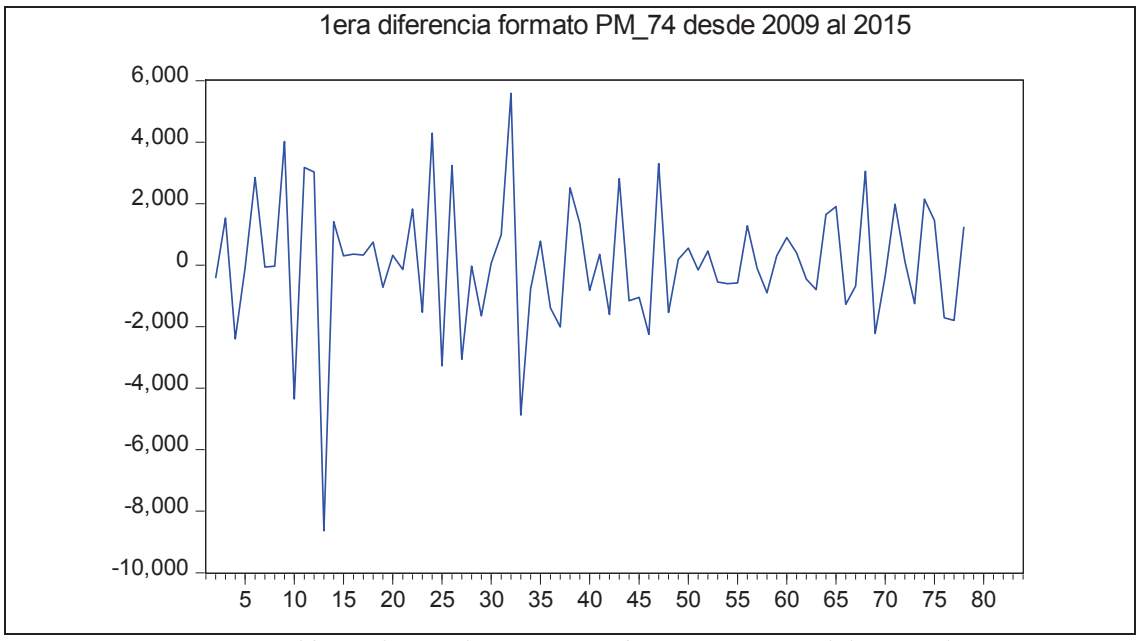


Figura 7.21 Diferencia regular consumo formato PM_74 del 2009 al 2015

(Gráfico Obtenido con el programa Eviews ver 9

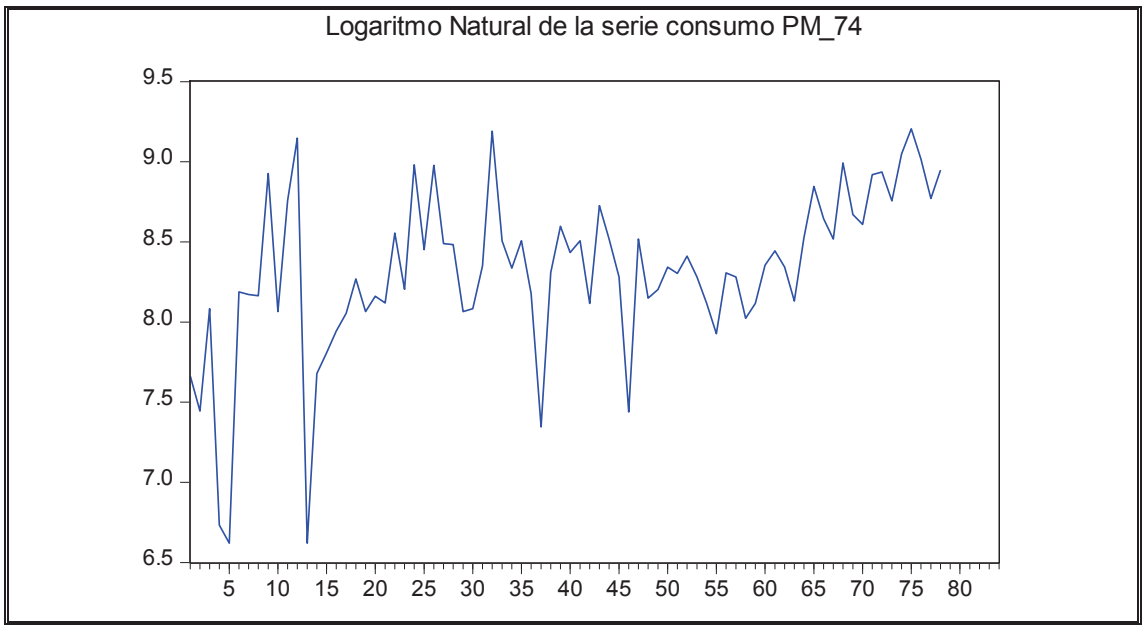


Figura 7.22 Logaritmo Natural consumo formato PM_74 del 2009 al 2015

(Gráfico Obtenido con el programa Eviews ver 9

El correlograma de la serie logarítmica se muestra en la Fig.23, a partir de esta figura y tomando en cuenta en principio de parsimonia, se puede suponer un modelo ARIMA (0,1,1).

En las figura 7.24 se muestra el correlograma del residuo de la serie después de aplicar un modelo ARIMA(0,1,1).

El correlograma es satisfactorio, se podría afirmar que prácticamente se tiene ruido blanco en el residuo.

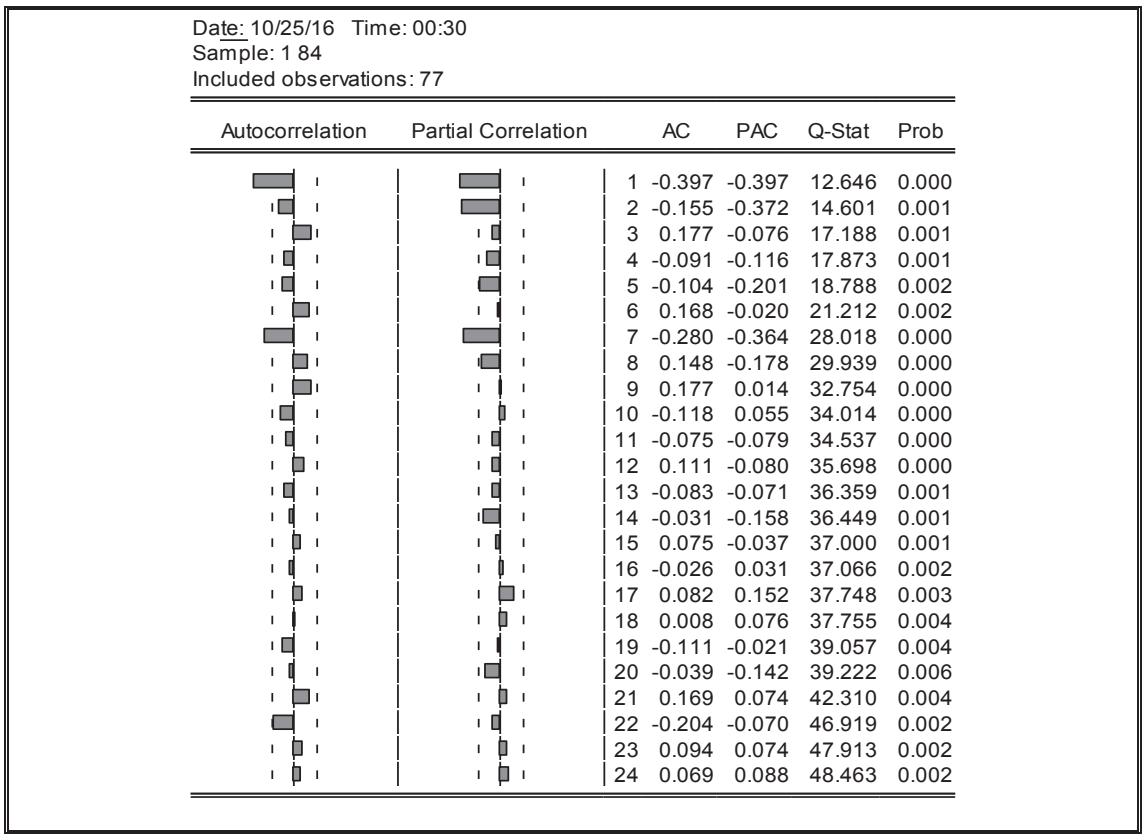
En la etapa de estimación se produjeron los resultados que se muestran en la tabla 7.17.

Se puede notar que el modelo es adecuado, ya que es invertible $|MA(1)| < 1$ y significativo, estadístico $t > 2$.

El Modelo ARIMA (0,1,1) para el formato PM 74 quedaría:

$$(1 - B)\tilde{Z}_t = 0.017 + (1 - 0.8190B)a_t \qquad (7.3)$$

Su pronóstico se presenta en la tabla 7.18.



	AC	PAC	Q-Stat	Prob
1	-0.397	-0.397	12.646	0.000
2	-0.155	-0.372	14.601	0.001
3	0.177	-0.076	17.188	0.001
4	-0.091	-0.116	17.873	0.001
5	-0.104	-0.201	18.788	0.002
6	0.168	-0.020	21.212	0.002
7	-0.280	-0.364	28.018	0.000
8	0.148	-0.178	29.939	0.000
9	0.177	0.014	32.754	0.000
10	-0.118	0.055	34.014	0.000
11	-0.075	-0.079	34.537	0.000
12	0.111	-0.080	35.698	0.000
13	-0.083	-0.071	36.359	0.001
14	-0.031	-0.158	36.449	0.001
15	0.075	-0.037	37.000	0.001
16	-0.026	0.031	37.066	0.002
17	0.082	0.152	37.748	0.003
18	0.008	0.076	37.755	0.004
19	-0.111	-0.021	39.057	0.004
20	-0.039	-0.142	39.222	0.006
21	0.169	0.074	42.310	0.004
22	-0.204	-0.070	46.919	0.002
23	0.094	0.074	47.913	0.002
24	0.069	0.088	48.463	0.002

Figura 7.23 Correlograma del logaritmo del consumo del formato PM_74

(Gráfico Obtenido con el programa Eviews ver 9

Tabla 7.17 Estimación modelo ARIMA (0,1,1) Formato PM_74

Dependent Variable: D(LOG(PM_74),1,0)
Method: ARMA Maximum Likelihood (BFGS)
Date: 10/24/16 Time: 20:17
Sample: 2 78
Included observations: 77
Convergence achieved after 7 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.017033	0.012758	1.335100	0.1859
MA(1)	-0.819068	0.053520	-15.30407	0.0000
SIGMASQ	0.205181	0.024948	8.224243	0.0000

R-squared	0.327148	Mean dependent var	0.016673
Adjusted R-squared	0.308963	S.D. dependent var	0.555837
S.E. of regression	0.462060	Akaike info criterion	1.346370
Sum squared resid	15.79895	Schwarz criterion	1.437687
Log likelihood	-48.83524	Hannan-Quinn criter.	1.382896
F-statistic	17.98982	Durbin-Watson stat	1.762590
Prob(F-statistic)	0.000000		

Inverted MA Roots	.82

Date: 10/25/16 Time: 00:24
Sample: 1 84
Included observations: 77
Q-statistic probabilities adjusted for 1 ARMA term

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.118	0.118	1.1053	
		2	-0.043	-0.057	1.2544	0.263
		3	0.079	0.092	1.7632	0.414
		4	-0.117	-0.145	2.9126	0.405
		5	-0.163	-0.124	5.1702	0.270
		6	-0.025	-0.012	5.2235	0.389
		7	-0.198	-0.200	8.6372	0.195
		8	0.123	0.200	9.9813	0.190
		9	0.202	0.114	13.636	0.092
		10	-0.036	-0.055	13.756	0.131
		11	-0.071	-0.124	14.215	0.163
		12	0.039	-0.002	14.360	0.214
		13	-0.056	0.016	14.656	0.261
		14	-0.003	0.030	14.657	0.329
		15	0.106	0.157	15.763	0.328
		16	0.084	0.089	16.472	0.351
		17	0.111	0.033	17.725	0.340
		18	-0.007	-0.125	17.730	0.406
		19	-0.140	-0.091	19.782	0.345
		20	-0.079	-0.009	20.455	0.368
		21	0.048	0.140	20.702	0.415
		22	-0.129	-0.063	22.544	0.369
		23	0.058	0.085	22.917	0.406
		24	0.081	-0.060	23.666	0.422

Figura 7.24 Correlograma residuos modelo Log. ARIMA(0,1,1), formato PM_74

(Gráfico Obtenido con el programa Eviews ver 9

Tabla 7.18 Pronósticos y errores Modelo ARIMA (0,1,1) para formato PM_74

Mes	**Pronóstico Eviews Log ARIMA (0,1,1)+C**	Reales	Error	(Error)^2	PEt	ABS PEt
79	7852	8700	848	719690.936	9.75%	9.75%
80	7987	7200	-786.537	618640.452	-10.92%	10.92%
81	8124	9450	1326.262	1758970.89	14.03%	14.03%
82	8263	8150	-113.295	12835.757	-1.39%	1.39%
83	8405	9800	1394.75	1945327.56	14.23%	14.23%
84	8550	7550	-999.644	999288.127	-13.24%	13.24%
			MSE	**1009125.62**	**MAPE**	**10.60%**
			RMSE	**1004.55**		

El programa Eviews nos permite generar los pronósticos directamente a partir de la serie logarítmica, se puede notar que el error MAPE es aceptable, por lo tanto se garantiza un buen pronóstico.

Como práctica se realizó el mismo pronóstico pero sin tomar el logaritmo de la serie original. En otras palabras se aplicó un modelo ARIMA (0,1,1) a la serie original, cuyo correlograma de los residuos se muestra en la figura 7.25.

En la tabla 7.19 se muestra la estimación del modelo ARIMA(0,1,1).

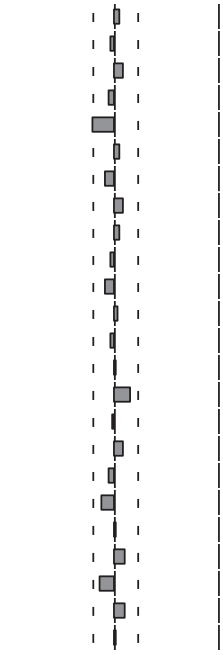
El Modelo ARIMA (0,1,1) para el formato PM 74 en este caso quedaría:

$$(1-B)\tilde{Z}_t = (1-0.7564B)a_t \qquad (7.4)$$

Se mantendrá la constante C, ya que mejora el pronóstico.

Su pronóstico se presenta en la tabla 7.20. En este caso los dos pronósticos obtenidos con y sin transformación logarítmica son muy similares.

Date: 10/25/16 Time: 01:11
Sample: 1 84
Included observations: 77
Q-statistic probabilities adjusted for 1 ARMA term



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.049	0.049	0.1902	
		2	-0.037	-0.039	0.2989	0.585
		3	0.083	0.087	0.8608	0.650
		4	-0.061	-0.072	1.1690	0.760
		5	-0.210	-0.199	4.9006	0.298
		6	0.062	0.075	5.2298	0.388
		7	-0.085	-0.102	5.8626	0.439
		8	0.089	0.144	6.5613	0.476
		9	0.047	-0.013	6.7608	0.563
		10	-0.046	-0.064	6.9520	0.642
		11	-0.092	-0.089	7.7293	0.655
		12	0.036	0.008	7.8517	0.727
		13	-0.044	0.022	8.0350	0.782
		14	0.012	0.012	8.0481	0.840
		15	0.170	0.168	10.887	0.695
		16	-0.013	-0.085	10.903	0.759
		17	0.082	0.123	11.592	0.772
		18	-0.057	-0.146	11.924	0.805
		19	-0.122	-0.041	13.484	0.762
		20	0.016	0.071	13.511	0.811
		21	0.100	0.079	14.605	0.799
		22	-0.143	-0.082	16.883	0.718
		23	0.106	0.024	18.153	0.697
		24	0.003	-0.046	18.154	0.749

Figura 7.25 Correlograma residuos modelo ARIMA(0,1,1), formato PM_74

(Gráfico Obtenido con el programa Eviews ver 9

Tabla 7.19 Estimación Modelo ARIMA (0,1,1) PM_74

Dependent Variable: D(PM_74)
Method: ARMA Maximum Likelihood (BFGS)
Date: 10/24/16 Time: 00:19
Sample: 2 78
Included observations: 77
Convergence achieved after 5 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	71.37782	59.46460	1.200341	0.2338
MA(1)	-0.756426	0.063094	-11.98894	0.0000
SIGMASQ	3098370.	425720.0	7.277953	0.0000

R-squared	0.333183	Mean dependent var	72.20779
Adjusted R-squared	0.315160	S.D. dependent var	2169.710
S.E. of regression	1795.544	Akaike info criterion	17.87321
Sum squared resid	2.39E+08	Schwarz criterion	17.96453
Log likelihood	-685.1187	Hannan-Quinn criter.	17.90974
F-statistic	18.48745	Durbin-Watson stat	1.901696
Prob(F-statistic)	0.000000		

Inverted MA Roots	.76

Tabla 7.20 Pronósticos y errores Modelo ARIMA (0,1,1) PM_74

Mes	**Pronóstico Eviews ARIMA (0,1,1)+C**	Reales	Error	(Error)^2	PEt	ABS PEt
79	7827.327	8700	873	761558.165	10.03%	10.03%
80	7898.705	7200	-698.705	488188.677	-9.70%	9.70%
81	7970.083	9450	1479.917	2190154.33	15.66%	15.66%
82	8041.461	8150	108.539	11780.7145	1.33%	1.33%
83	8112.839	9800	1687.161	2846512.24	17.22%	17.22%
84	8184.217	7550	-634.217	402231.203	-8.40%	8.40%
			MSE	**1116737.55**	**MAPE**	**10.39%**
			RMSE	**1056.76**		

7.2.4 PREDICCIÓN FORMATO 1030X790X0.30 (PM102)

En la tabla 7.21 se muestra la información del consumo de placas digitales formato 1030x790x0.30 (cuyas dimensiones son ancho, largo y espesor de la placa en mm), correspondiente a una prensa modelo SM_102, en la ciudad de Quito desde el año 2009 hasta el año 2015.

En esta sección se utilizará el paquete estadístico Minitab ver 17, que ayudará en la identificación, estimación y diagnóstico de los modelos mediante la metodología ARIMA, con el modelo ARIMA adecuado se realizará el pronóstico respectivo, para luego calcular el error de pronóstico MAPE que ayudará a verificar la precisión del modelo.

Tabla 7.21 Consumo de Placas Formato 1030x790x0.30 desde el 2009 hasta 2015

Mes	2009	2010	2011	2012	2013	2014	2015
1	6630	2250	2640	266	2510	3800	4280
2	90	570	3300	7174	2880	4720	4480
3	630	1680	3360	6440	5320	5080	6080
4	1380	2610	450	240	7660	8199	6392
5	60	3330	2160	1560	7040	8860	6000
6	2070	180	2700	640	6840	6720	5080
7	1920	5310	2880	4520	7180	6120	5040
8	3720	2760	2790	4680	6280	5971	6200
9	270	4830	2040	3200	4526	5120	4800
10	3840	2730	1980	2080	4880	6120	5360
11	2250	4110	3150	4920	4560	6160	5680
12	4650	3000	60	3670	4480	6020	5960

En la figura 7.26 se muestra el gráfico de la demanda de placas digitales (SM 102) desde el año 2009 hasta el año 2015 en el mercado gráfico quiteño.

Se seguirá el mismo procedimiento seguido en formatos anteriores, es decir se separarán las últimas 6 observaciones, para verificar el error del pronóstico contra los valores reales.

A simple vista parece que esta serie es no estacionaria en su media, ya que se observa un componente de tendencia y ciertos picos estacionales. Debido a la presencia de un patrón estacional, no se puede asegurar que se necesita una diferencia regular, además no se detecta mayor cambio en la varianza.

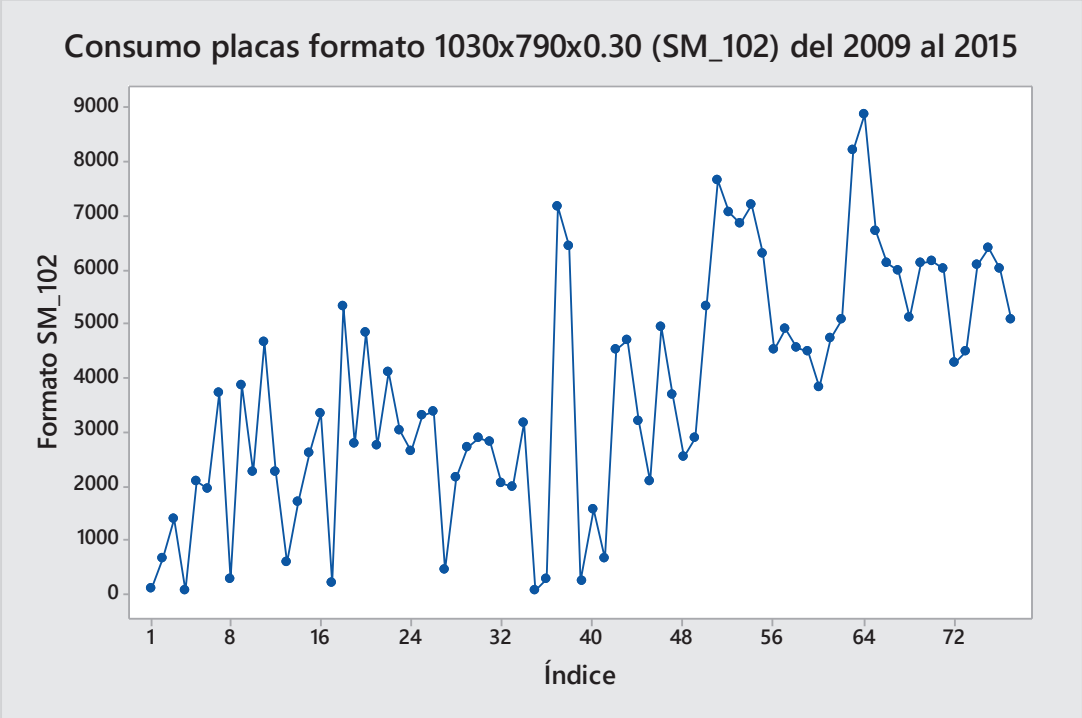


Figura 7.26 Consumo de placas formato 1030x790x0.30 (SM_102) (2009 – 2015)

Para comprobar si la serie es no estacionaria, se corre la prueba de Dickey Fuller Aumentada (DFA), mediante el programa EViews 9. Los resultados se muestran en la tabla 7.22.

El estadístico t de la prueba de DFA (-6.4332) se encuentra en la zona de rechazo de la Hipótesis Nula (valor menor a los tres valores críticos), por lo tanto no aceptamos la hipótesis nula que existe una raíz unitaria, que comprueba a su vez que la serie es estacionaria, cosa que a simple vista no se ve claramente.

En las figuras 7.27 y 7.28 se muestran los gráficos de las funciones de autocorrelación y autocorrelación parcial (sacf y spacf).

A continuación se tratará de identificar uno o más modelos cuyas acfs y pacfs teóricas sean similares a las estimadas sacf y spacf, ya que según la prueba de DFA podemos asumir estacionariedad.

La función sacf tienen picos significativos en los rezagos 4,8,12 que advierten la presencia de un patrón estacional.

Tabla 7.22 Prueba DFA del Consumo de Placas Digitales Formato 1030x790x0.30

Null Hypothesis: DFA_SM_102 has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 0 (Automatic - based on SIC, maxlag=11)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-6.433212	0.0000
Test critical values:	1% level	-4.083355	
	5% level	-3.470032	
	10% level	-3.161982	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DFA_SM_102)
Method: Least Squares
Date: 10/25/16 Time: 20:01
Sample (adjusted): 2 77
Included observations: 76 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DFA_SM_102(-1)	-0.723959	0.112535	-6.433212	0.0000
C	995.1813	392.1482	2.537768	0.0133
@TREND("1")	46.47545	11.30188	4.112186	0.0001

R-squared	0.362311	Mean dependent var	65.65789
Adjusted R-squared	0.344840	S.D. dependent var	1973.645
S.E. of regression	1597.507	Akaike info criterion	17.62895
Sum squared resid	1.86E+08	Schwarz criterion	17.72095
Log likelihood	-666.9001	Hannan-Quinn criter.	17.66572
F-statistic	20.73791	Durbin-Watson stat	1.966321
Prob(F-statistic)	0.000000		

(Tabla Obtenida con el programa Eviews ver 9

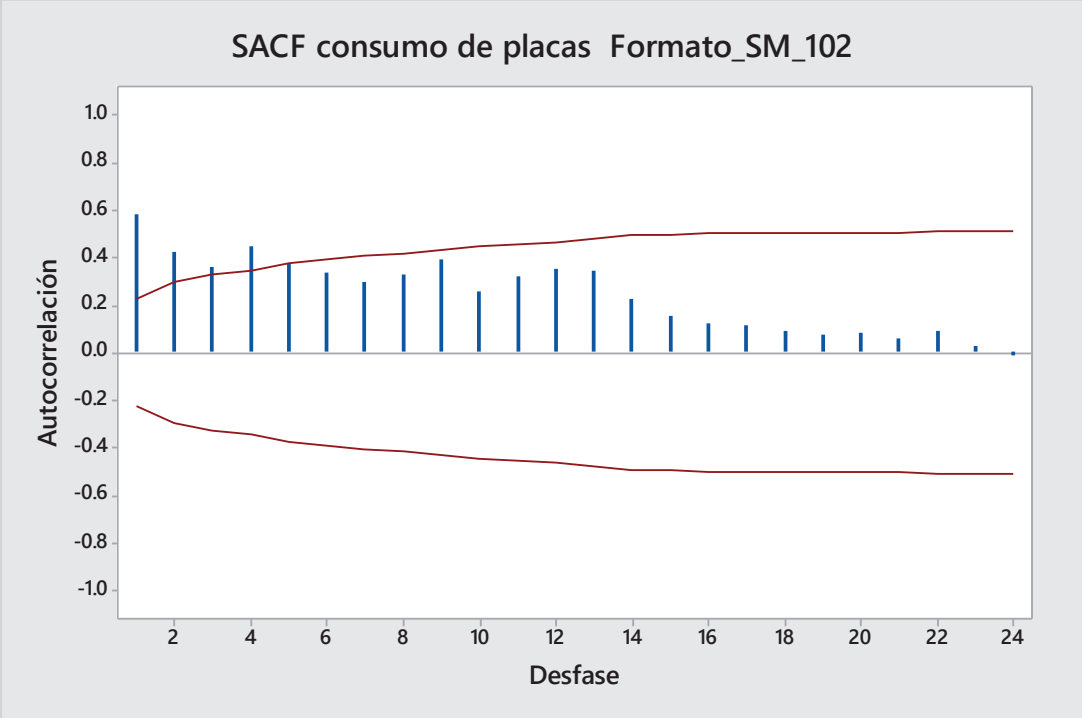


Figura 7.27 Autocorrelación del Consumo de placas digitales formato SM_102

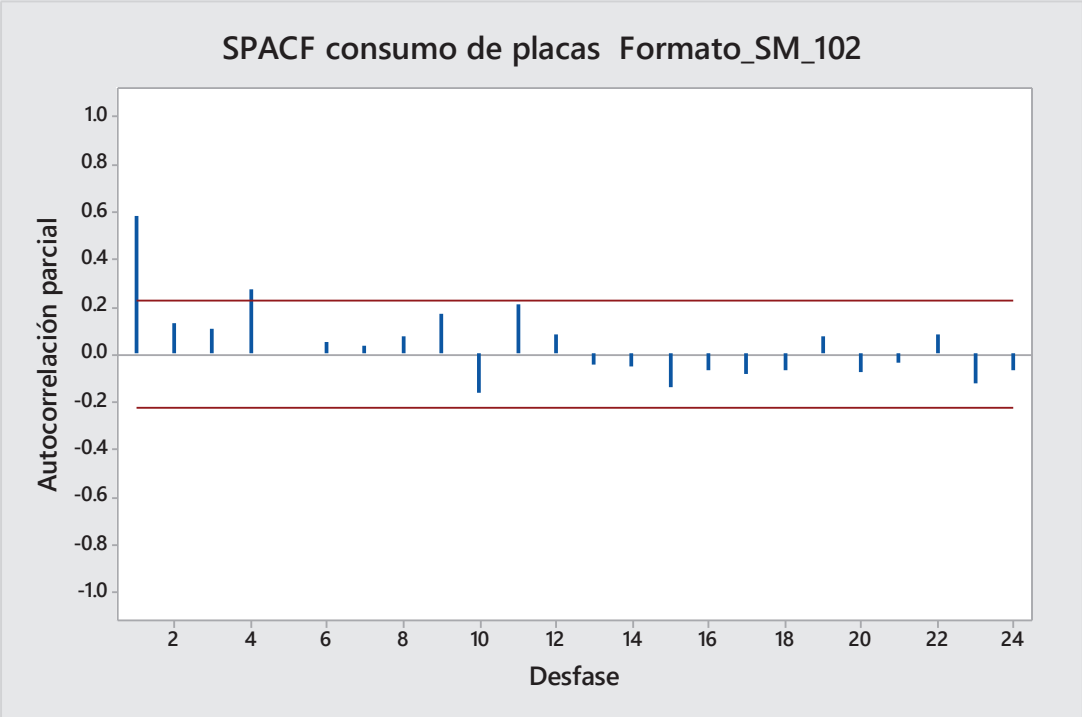


Figura 7.28 Autocorrelación Parcial del Consumo de placas digitales formato SM_102

La figura 7.29 muestra el gráfico de la serie después de aplicar una diferencia estacional, el patrón estacional ya no existe, por lo tanto no se necesita otra diferencia estacional.

En las figura 7.30 y 7.31 se muestran las funciones sacf y spacf de la serie después de aplicar una diferencia estacional.

Después de varias simulaciones y de acuerdo con las funciones sacf y spacf, el modelo que mejor se desempeña es el siguiente: $ARIMA(1,0,0)(1,1,0)_{12}$

En las figuras 7.32 y 7.33 se muestran las funciones sacf y spacf de los residuos de la serie después de aplicar el modelo. $ARIMA(1,0,0)(1,1,0)_{12}$, las dos funciones son satisfactorias.

En la etapa de estimación se produjeron los resultados que se muestran en la tabla 7.23, mediante el programa Minitab ver 17. Se puede observar que los coeficientes cumplen con la condición de estacionariedad y son significativos.

Además en la tabla 7.24 se pueden ver los valores del estadístico Q^* de Ljung – Box, todos ellos se han chequeado con la tabla C2 (Distribución χ^2) del Anexo C y son adecuados (menor que los valores al 0.05).

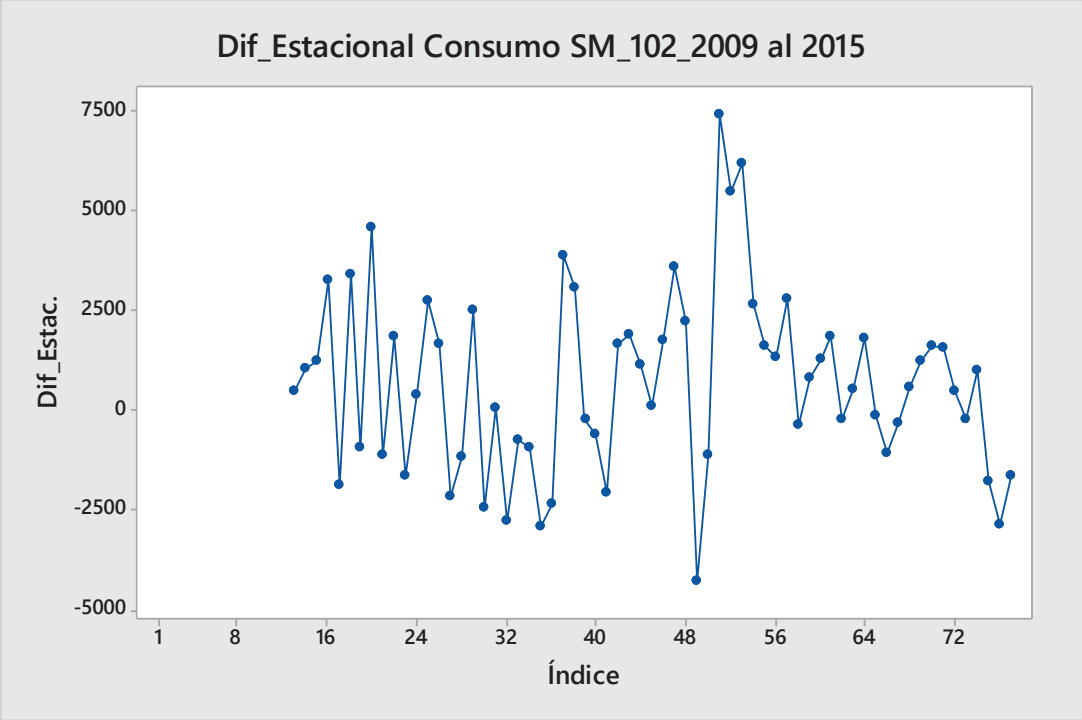


Figura 7.29 Gráfico de la serie después de una diferencia estacional formato SM_102

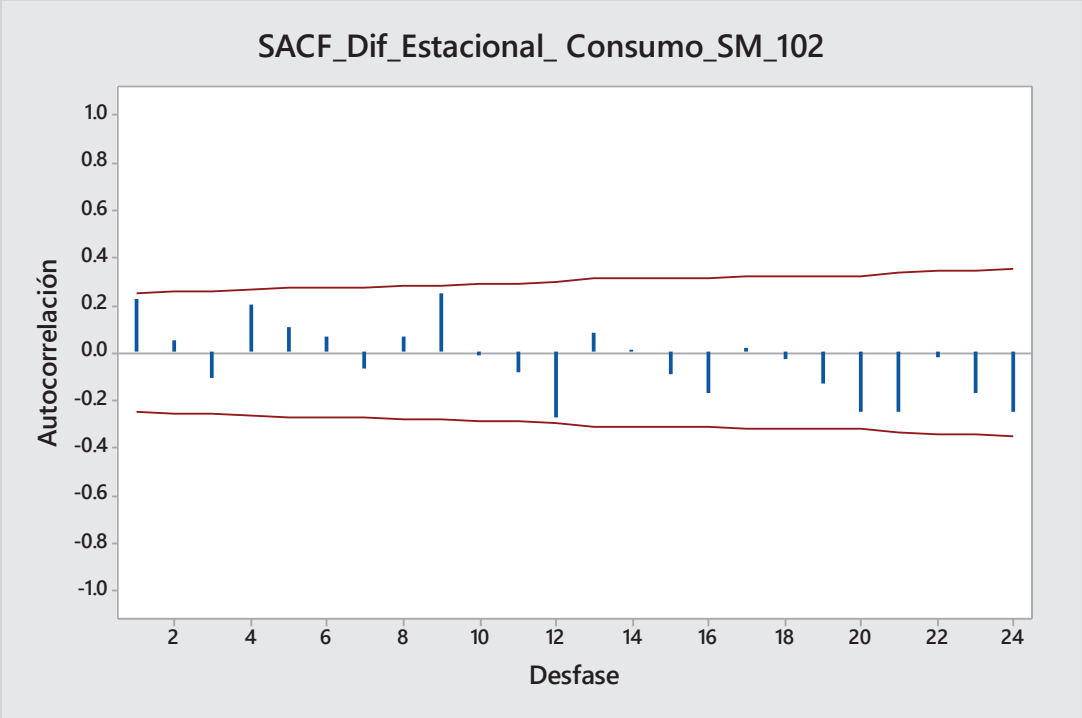


Figura 7.30 Autocorrelación de la serie con diferencia estacional formato SM_102

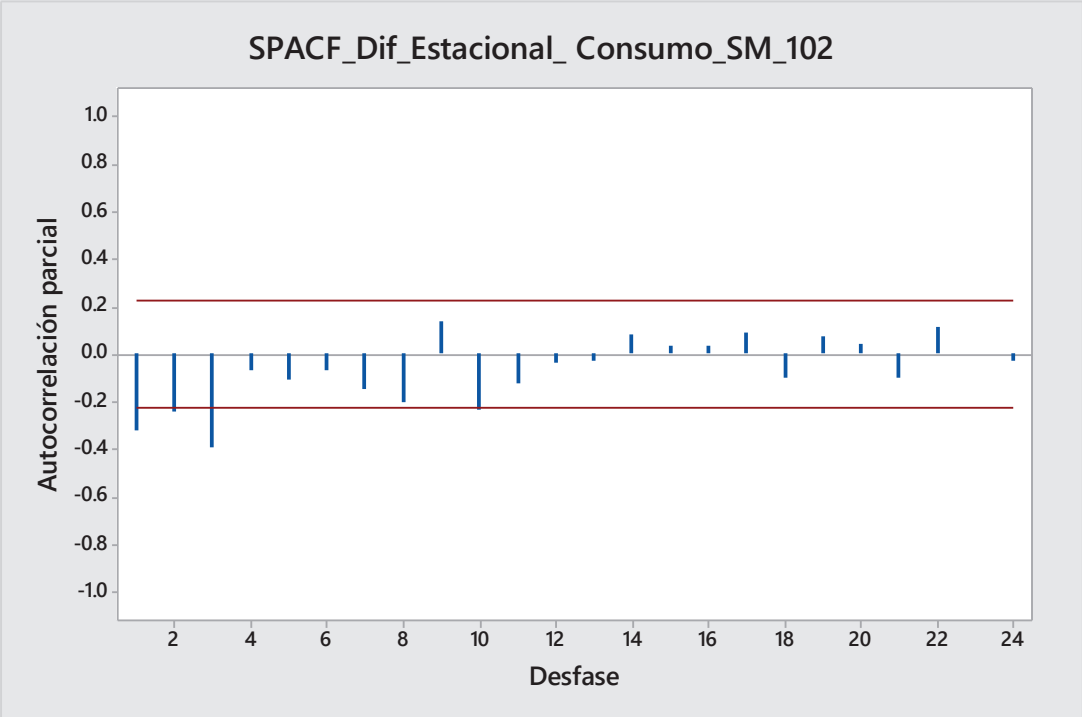


Figura 7.31 Autocorrelación parcial de la serie con diferencia estacional SM_102

Tabla 7.23 Estimación modelo $ARIMA(1,0,0)(1,1,0)_{12}$Formato SM_102

```
Estimaciones finales de los parámetros

Tipo        Coef  SE Coef      T      P
AR   1    0.4231   0.1137   3.72  0.000
SAR  12  -0.4150   0.1159  -3.58  0.001


Diferenciación: 0 regular, 1 estacional de orden 12
Número de observaciones:  Serie original 77, después de diferenciar 65
Estimaciones finales de los parámetros
```

Tabla 7.24 Estadístico de Ljung – Box Q^* $ARIMA(1,0,0)(1,1,0)_{12}$Formato SM_102

Estadística Chi-cuadrada modificada de Box-Pierce (Ljung-Box)				
Desfase	12	24	36	48
Chi-cuadrada	14.0	35.6	49.0	53.8
GL	10	22	34	46
Valor p	0.175	0.034	0.046	0.201

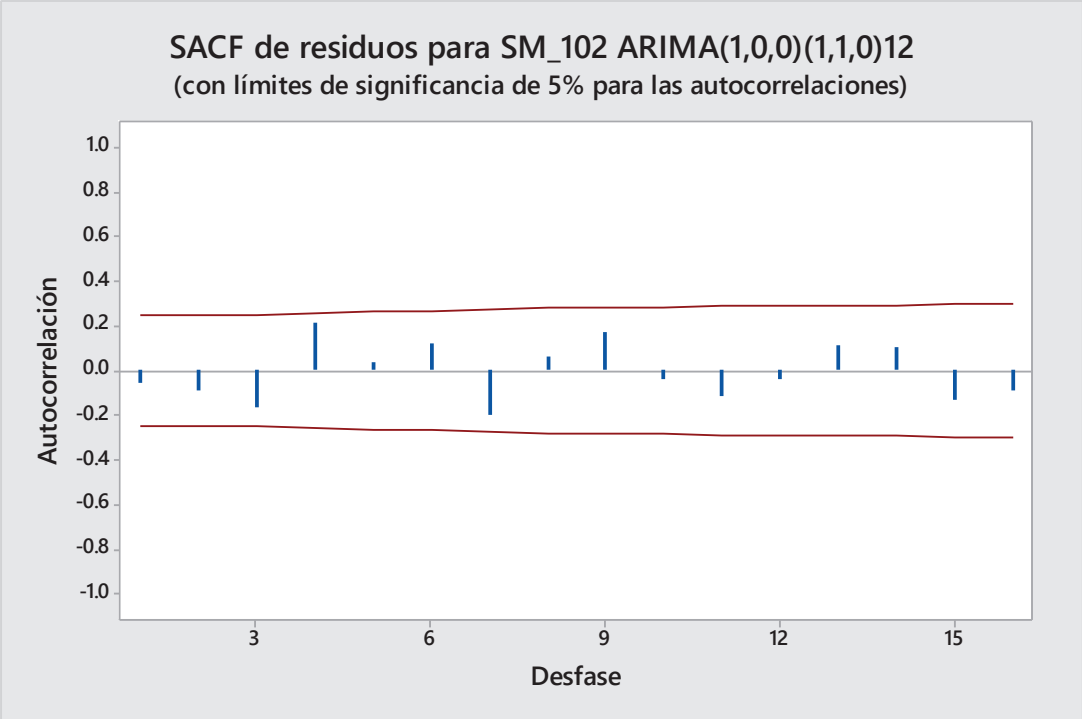


Figura 7.32 Autocorrelación residuos $ARIMA(1,0,0)(1,1,0)_{12}$ Formato SM_102

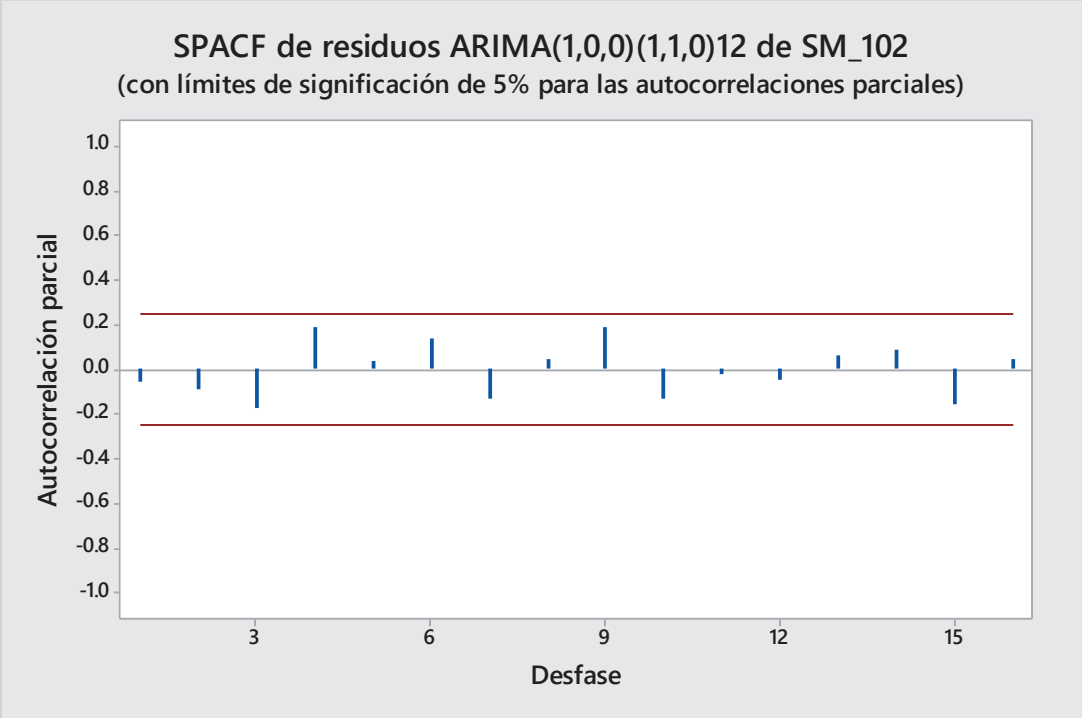


Figura 7.33 Autocorrelación parcial residuos $ARIMA(1,0,0)(1,1,0)_{12}$ SM_102

Después de chequear todos los criterios anteriores se puede concluir que el Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ es estadísticamente adecuado.

El Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ para el formato SM 102 quedaría:

$$(1 - 0.4231B)(1 + 0.415B^{12})(1 - B^{12})\tilde{Z}_t = a_t \qquad (7.4)$$

Su pronóstico se presenta en la tabla 7.25.

La tabla 7.25 indica que el error MAPE no es considerable, por lo tanto se puede concluir que el pronóstico es bastante bueno.

Tabla 7.25 Pronósticos y errores del Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ SM_102

Mes	Pronóstico Minitab ARIMA (1,0,0)(1,1,0)12	Reales	Error	(Error)^2	PEt	ABS PEt
79	5845	5040	-805	647978.127	-15.97%	15.97%
80	5797	6200	403.249365	162610.05	6.50%	6.50%
81	4745	4800	54.5002724	2970.27969	1.14%	1.14%
82	5551	5360	-191.233159	36570.1212	-3.57%	3.57%
83	5473	5680	206.933893	42821.6362	3.64%	3.64%
84	5371	5960	588.816215	346704.536	9.88%	9.88%
			MSE	**206609.125**	**MAPE**	**6.78%**
			RMSE	**454.54**		

En la figura 7.34 se muestra el gráfico de los pronósticos con su respectivo intervalo de predicción al 95 % de confianza.

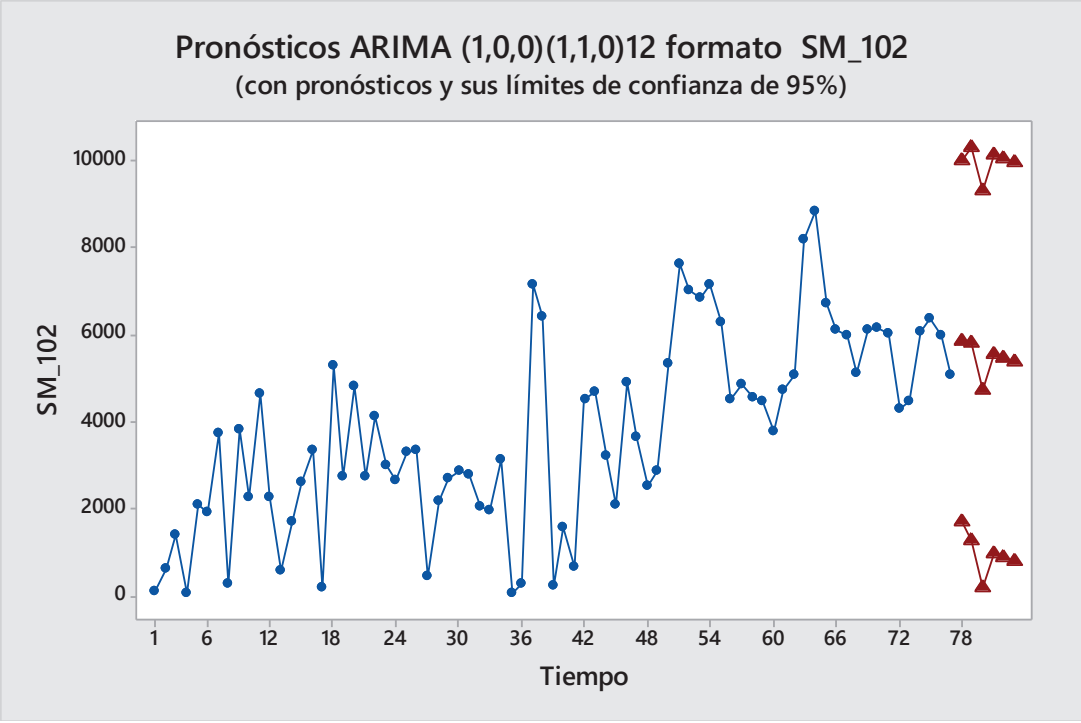


Figura 7.34 Gráfico Pronósticos Modelo $ARIMA(1,0,0)(1,1,0)_{12}$ Formato SM_102

8 CONCLUSIONES Y RECOMENDACIONES

Al terminar esta investigación se puede concluir y recomendar lo siguiente:

8.1 CONCLUSIONES

-El mejor pronóstico con el método de Holt – Winters, lo produjo un modelo Holt-Winters Multiplicativo con los parámetros $\alpha = 0.4$ $\beta = 0.2$ y $\gamma = 0.4$ corrido con el programa Minitab Ver 17, dando un error MAPE del 4.46%

-La metodología de Box-Jenkins con un modelo $ARIMA(2,1,2)(0,0,1)_{12}$ descrito por la Ec: (5.104):

$(1 - 0.8343B + 0.6395B^2)(1 - B)\tilde{Z}_t = (\ 1 + 0.34B^{12})(1 - 1.2233B + 0.7796B^2)a_t$

Simulado por el programa Minitab ver. 17, produjo el menor error MAPE=3.67%.

-Con el método de redes neuronales, una red dinámica tipo Narnet (2-10-1), es decir, con 2 retardos en la línea de entrada, 10 neuronas en la capa oculta, 1 neurona en la capa de salida, con una función de transferencia en la capa oculta tipo $tan - sigmoid$ y una función de transferencia $lineal$ en la capa de salida, entrenada con el algoritmo de Levenberg-Marquardt y simulada en la plataforma Matlab ver 15 mediante el programa *Prog_Narnet_Tesis_OK.m*, fue la que mejor pronóstico produjo con un MAPE del 4.12%.

-La metodología de Box-Jenkins produce los mejores pronósticos a corto plazo (6 pasos adelante) como se ha podido observar en la sección 5.4 del presente trabajo, en cambio las redes neuronales dinámicas producen excelentes pronósticos a largo plazo como se analizó en la sección 6.3 (200 pasos adelante).

-No se puede garantizar que un método de pronóstico sea mejor que otro, ya que este depende de algunos factores de las series de tiempo como son: cantidad de datos disponibles para el pronóstico, horizonte de tiempo a predecir, forma de la serie de tiempo (que tan irregular es la serie en el tiempo), nivel de estacionalidad, etc.

-Los métodos de Holt-Winters y Box-Jenkins permiten conocer mejor los datos, ya que de acuerdo a ellos se deben escoger sus parámetros y valores iniciales. Se pueden detectar estacionalidades, tendencias, que son muy importantes en las decisiones gerenciales. En cambio en el método de redes neuronales no se necesita un conocimiento profundo de los datos, simplemente los datos son procesados y optimizados para obtener el mejor pronóstico, sin importar la forma de los mismos.

-Las Redes Neuronales son una alternativa relativamente nueva y promisoria para problemas de predicción, pero se debe elegir muy bien su estructura,
(número de capas, número de neuronas en cada capa, funciones de transferencias, etc) y el método de entrenamiento, ya que estos factores tienen un gran impacto en la velocidad y precisión del pronóstico.

-La función de autocorrelación de los residuos es una herramienta muy útil para decidir si un pronóstico es bueno, mientras más se asemeje a ruido blanco, mejor será el pronóstico.

8.2 RECOMENDACIONES

-Se recomienda antes de cualquier acción en un pronóstico, hacer un gráfico de la serie de tiempo, este nos permitirá detectar linealidad, tendencias, estacionalidades e irregularidades, para de acuerdo a estas características escoger el método de pronóstico adecuado.

-Para la metodología ARIMA, a veces hay patrones estacionales escondidos en las series de tiempo, por lo tanto se recomienda tomar la primera diferencia regular $(d = 1)$, ya que la función acf de esta serie tendrá al patrón estacional más claro que la serie original.

-Para la metodología de Box-Jenkins, a veces hay series de tiempo con patrones estacionales pronunciados, se recomienda primero identificar y estimar el patrón estacional, para luego examinar la función acf residual para el patrón no estacional.

-A veces en ciertas series es difícil identificar un modelo mixto ARIMA. Es útil es estos casos primero estimar un modelo AR puro, para luego aumentar coeficientes de acuerdo a la función acf residual.

-Para mejorar la precisión del pronóstico se recomienda volver a calcular los parámetros o volver a entrenar la red neuronal, a medida que vayan apareciendo nuevos valores reales. Es decir hacer la predicción un paso adelante.

-En las redes neuronales para mejorar el pronóstico se recomienda seguir el siguiente orden:

- Re inicializar la red varias veces (Se atrapa en mínimos locales).
- Incrementar el número de neuronas en la capa oculta (gradualmente).
- Intente con una función de entrenamiento diferente Ejemplo: trainlm.
- Finalmente, utilice datos adicionales para el entrenamiento, con más datos mejorará la generalización de la red.

-En las redes neuronales dinámicas Narnet la función de autocorrelación del error nos da pistas acerca del valor de la longitud del retardo en las líneas de entrada, si existe correlación en el error de predicción se debería incrementar la longitud del retardo. En redes dinámicas tipo Narx la función de autocorrelación cruzada cumple la misma función, si existe correlación entre el error de predicción y la secuencia de entrada se debería incrementar el retardo en las líneas entre la entrada y la realimentación de la red.

-De acuerdo a la cantidad de datos que la serie de tiempo tiene disponibles para el pronóstico, se debe escoger el método adecuado así: Si la cantidad de datos es muy limitada (menor a 50), se debería utilizar el método de Holt-Winters, ya que Box y Jenkins sugieren utilizar su método con al menos 50 valores, si los valores son limitados pero mayores a 50, se debería utilizar la metodología ARIMA o Holt-Winters, pero para cantidades de datos muy grandes se recomienda utilizar el método de redes neuronales.

-Si los datos de una serie de tiempo son lineales y limitados, se recomienda utilizar la tecnología de Box-Jenkins como primera opción, en cambio si los datos son no lineales y en gran cantidad, se recomienda utilizar el método de redes neuronales como primera opción.

REFERENCIAS

Arteaga E. (2010). *Construcción de un Modelo Econométrico para estimar las ventas mensuales de las cuatro marcas principales de bebidas gaseosas de la empresa Ecuador Bottling Company Corp.* Quito: EPN - Facultad de Ciencias.

Azoff, M. (1994). *Neural Network time series forecasting of financial markets.* Chichester: John Wiley & Sons.

Beale M., H. M. (2015). *Neural Network Toolbox User Guide* . Middletown, DE.

Box G., J. G. (2008 4th Edition). *Time Series Analysis Forecasting and Control.* New Jersey: John Wiley & Sons, Inc., Publication.

Brockwell, R. (2009). *Introduction to Time Series and Forecasting.* New York: Springer - Verlag.

Capa, H. (2007). *Modelación de Series Temporales.* Quito: Designio.

Capa, H. (2008). *Un Primer Curso en Series Temporales.* Quito: Designio.

Chatfield C., M. Y. (1988). Holt - Winters Forecasting: Some Practical Issues. *The Statistician*, 129 - 140.

Chatfield, C. (2012 Sixth Edition). *The Analysis of Time Series An Introduction* . Boca Raton , Florida 33431: Chapman & Hall/CRC.

Cryer J. (2009). *Time Series Analysis With Applications in R.* New York: Springer.

Del Brío, B. (2007). *Redes Neuronales y Sistemas Borrosos 3era. Ed.* México D.F.: Alfaomega Grupo Editor.

Du K., S. M. (2014). *Neural Networks and Statistical Learning.* London: Springer - Verlag.

Fausett, L. (1994). *Fundamentals of Neural Networks.* New Jersey: Prentice-Hall.

Gujarati Damodar N, P. D. (2010 Quinta Edición). *Econometría.* México: McGrawHill Education.

Hagan M., D. H. (2015). *Neural Network Design, 2nd Edition.* Middletown.

Hamilton, J. (1994). *Time Series Analysis.* Princeton, NJ: Princeton University Press.

Harvey, A. (1990). *Forecasting, Structural Time Series Models and the Kalman Filter.* Cambridge: Cambridge University Press.

Harvey, A. (1991). *The Econometric Analysis of Time Series 2nd. Ed.* Cambridge, MA: The MIT Press.

Haykin, S. (1999). *Neural Networks a Comprehensive Foundation.* New Jersey: Prentice Hall.

Heath, G. (2016, 06 15). *MathWorks*. Retrieved from MathLab Answers: https://www.mathworks.com/matlabcentral/answers/

Hecht Nielsen, R. (1991). *Neurocomputing.* New York: Addison-Wesley Publishing Company.

HEIZER, J., & RENDER, B. (2004). Principios de Administración de Operaciones. In J. HEIZER, & B. RENDER, *Principios de Administración de Operaciones* (p. 13). Mexico: Pearson Education.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction toThe Theory of Neural Computation.* Redwood: Addison-Wesley Publishing Company.

Hofacker, A. (2008). *Rapid lean construction - quality rating model.* Manchester: s.n.

Hyndman R., A. G. (2014). *Forecasting Principles and Practice.* Middletown, DE: Otext.

Koskela, L. (1992). *Application of the new production philosophy to construction.* Finland: VTT Building Technology.

Levin, R. R. (2010). *Estadística para Administración y Economía.* México D.F.: Pearson Education de México, S.A. de C.V.

Lisboa, P., Edisbury, B., & Vellido, A. (2000). *Business Applications of Neural Networks.* New Jersey: World Scientific.

Makridakis S., W. S. (1998). *Forecasting Methods and Applications.* USA: John Wiley & Sons, Inc.

McNelis, P. (2005). *Neural Networks in Finance.* New York: Elsevier Academic Press.

Mills, T. (1992). *Time Series techniques for economists.* Cambridge, GB: Cambridge University Press.

Pankratz, A. (1983). *Forecasting With Univariate Box-Jenkins Models.* Toronto: John Wiley & Sons.

Pankratz, A. (1991). *Forecasting with Dynamic Regression Models.* Canada: John Wiley & Sons.

Principe, J. (2000). *Neural and Adaptive Systems. Fundamentals through Simulations.* New York: John Wiley&Sons.

Ramón y Cajal, S. (1990). *New Ideas on the Structure of the Nervous System in Man and Vertebrates.* Cambridge, MA: The MIT Press.

Ramón y Cajal, S. (1995). *Histology of the Nervous System Vol. I.* Oxford: Oxford University Press, Inc.

Rogers, R., & Vemuri, R. (1994). *Artificial Neural Networks Forecasting Time Series.* Los Alamitos CA: IEEE Computer Society Press.

Rumelhart, D. E., McClelland, J. L., & Group, t. P. (1987). *Parallel Distributed Processing - 2 Vol. Set: Explorations in the Microstructure of Cognition.* Masachusets: The MIT Press.

Schroeder R., G. S. (2011). *Administración de Operaciones.* México: Mc Graw Hill Educación.

Shukla, P. (2010). *Levenberg-Marquardt Algorithms for Nonlinear Equations, Multi-objective Optimization, and Complementarity Problems.* Shaker Verlag.

Shumway R. (2011). *Time Series Analysis and Its Applications.* London: Springer.

Terence, M. (1990). *Time Series techniques for economists.* Cambridge: Cambridge University Press.

Tiao G., T. R. (2001). *A Course in Time Series Analysis.* New York: John Wiley & Sons, Inc.

Tsay, R. (2010 3rd Edition). *Financial Time Series.* New Jersey: John Wiley $ Sons, Inc.

Valverde A., P. L. (2012). *Evaluación de Modelos Econométricos Alternativos de Series de Tiempo para el Pronóstico de la Inflación en el Ecuador en el cotro plazo: Período 2000-2010.* Cuenca: Universidad de Cuenca.

Winston, W. L. (2005). *Investigación de Operaciones 4ta. Ed.* México D.F.: International Thomson Editores S.A. de C.V.

Wooldridge, J. (2010). *Introducción a la Econometría Un enfoque Moderno 4ta. Ed.* México D.F.: Cengage Learning.

Yaffee, R. (2000). *Introduction to Time Series Analisys and Forecasting with applications of SAS and SPSS.* USA: Academic Press, Inc.

Zavaleta E., C. E. (2010). *Sistema de pronóstico de la demanda de productos farmaceúticos basado en redes neuronales.* Lima: Universidad San Marcos - Facultad de Sistemas.

Zhang, P. (2004). *Neural Networks in Business Forecasting.* London: Idea Group Inc.

ANEXOS

ANEXO A - Modelo de la orden de encuadernación

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS ADMINISTRATIVAS

ORDEN DE ENCUADERNACIÓN

De acuerdo con lo estipulado en el Art. 17 del instructivo para la Aplicación del Reglamento del Sistema de Estudios, dictado por la Comisión de Docencia y Bienestar Estudiantil el 9 de agosto del 2000, y una vez comprobado que se han realizado las correcciones, modificaciones y más sugerencias realizadas por los miembros del Tribunal Examinador al informe del proyecto de titulación {ó tesis de grado} presentado por JHIMY XAVIER PONCE JARRíN.

Se emite la presente orden de empastado, con fecha mes día del año.

Para constancia firman los miembros del Tribunal Examinador:

NOMBRE	FUNCIÓN	FIRMA
Ing. Alex Davila Frías	Director	
	Examinador	
	Examinador	

_____

Mat. Nelson Alomoto

DECANO

ANEXO B – Información del consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2105.

Tabla B.1 – Consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2015 formato 510x400x0.15 (GTO_52)

Mes	2009	2010	2011	2012	2013	2014	2015
1	16600	16750	15300	17400	32390	45300	33700
2	16600	18000	18750	20350	30400	39672	30660
3	26850	26500	17750	24950	36300	34000	37900
4	16250	22800	14400	23800	37200	36862	35885
5	18150	23000	16300	37600	37956	42900	36400
6	18550	22150	19100	25323	32044	39233	36190
7	22200	20650	16100	29050	37200	38025	33900
8	21250	22265	16400	35600	34526	36500	32400
9	19300	25230	17700	30350	38880	36800	38300
10	21400	23100	18200	37100	41982	38500	36000
11	19950	18700	15150	42260	47700	43200	39300
12	28850	22800	19000	40900	48464	44300	39300

Tabla B.2 – Consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2015 formato 525x459x0.15 (SM_52)

Mes	2009	2010	2011	2012	2013	2014	2015
1	5400	4050	4950	3300	3500	3900	5700
2	3900	1950	12750	4850	3500	6500	7400
3	7850	6650	6550	5450	4200	6540	11400
4	2050	4450	2350	4000	4600	7099	6900
5	4250	5550	3000	5500	4600	6900	8700
6	2250	6260	5000	4950	4070	7099	7150
7	3900	6050	5350	3200	5100	5898	7400
8	1750	8008	8100	5400	4188	5292	7800
9	5550	6150	3400	4500	4698	6200	7200
10	3100	6900	5750	5300	5200	6700	8000
11	5950	7300	5300	4800	6400	6572	8500
12	8350	11700	3750	4900	5800	7200	7200

Tabla B.3 – Consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2015 formato 650x550x0.30 (KORD_MO)

Mes	2009	2010	2011	2012	2013	2014	2015
1	3990	5250	2250	4900	8750	14800	2650
2	3210	5580	3150	6260	8800	11730	8700
3	5880	7400	3240	8580	11050	8250	7680
4	5580	6600	2910	8500	14406	9620	5500
5	4200	6180	5340	10150	9850	10900	8900
6	3720	5580	4860	5580	8405	8449	7100
7	6960	6000	3140	12540	11619	7795	7890
8	5400	6877	2400	10650	7447	9698	9350
9	5700	4680	3930	7350	9448	8550	9050
10	5790	4590	3030	8900	8425	7700	8650
11	5910	3390	2470	11300	15098	10646	9150
12	10020	4080	3480	9450	16750	8800	9283

Tabla B.4 – Consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2015 formato 745x605x0.30 (PM_74)

Mes	2009	2010	2011	2012	2013	2014	2015
1	2130	750	4680	1550	3650	4650	6350
2	1710	2160	7920	4060	4200	4200	8500
3	3240	2460	4860	5420	4040	3400	9950
4	840	2820	4830	4600	4500	5047	8240
5	750	3150	3180	4950	3950	6950	6440
6	3600	3900	3240	3350	3350	5680	7690
7	3540	3180	4230	6160	2770	5000	8700
8	3510	3500	9810	5000	4050	8050	7200
9	7530	3360	4940	3950	3950	5830	9450
10	3180	5190	4170	1700	3050	5485	8150
11	6360	3660	4950	5000	3350	7471	9800
12	9390	7950	3560	3460	4250	7600	7550

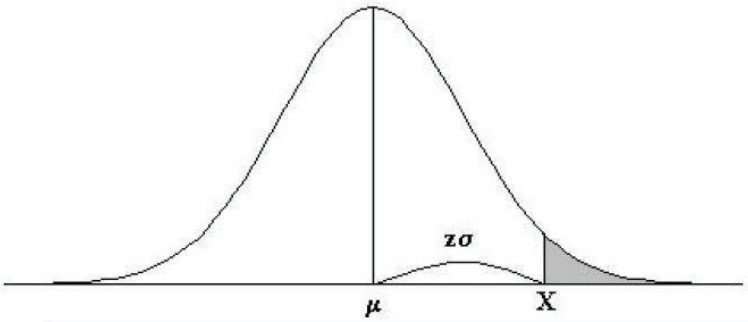
Tabla B.5 – Consumo de placas digitales en el mercado gráfico quiteño desde el año 2009 al 2015 formato 1030x790x0.30 (SM_102)

Mes	2009	2010	2011	2012	2013	2014	2015
1	6630	2250	2640	266	2510	3800	4280
2	90	570	3300	7174	2880	4720	4480
3	630	1680	3360	6440	5320	5080	6080
4	1380	2610	450	240	7660	8199	6392
5	60	3330	2160	1560	7040	8860	6000
6	2070	180	2700	640	6840	6720	5080
7	1920	5310	2880	4520	7180	6120	5040
8	3720	2760	2790	4680	6280	5971	6200
9	270	4830	2040	3200	4526	5120	4800
10	3840	2730	1980	2080	4880	6120	5360
11	2250	4110	3150	4920	4560	6160	5680
12	4650	3000	60	3670	4480	6020	5960

ANEXO – C Tablas de Distribución de Probabilidad

Tabla C.1 – Tabla de Distribución Normal

Áreas bajo la curva normal



Ejemplo:

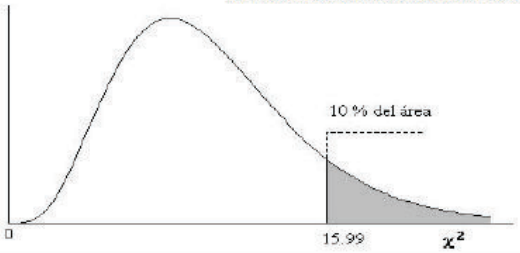
$$Z = \frac{X - \mu}{\sigma}$$

P [Z > 1] = 0.1587
P [Z > 1.96] = 0.0250

Desv. normal x	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010

Tabla C.2 – Tabla de Distribución χ^2(ji Cuadrada)

Puntos de porcentaje de la distribución χ^2



Ejemplo:
Para $\phi = 10$ grados de libertad

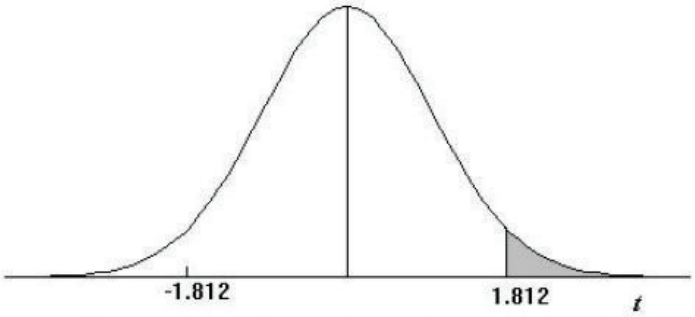
$P[\chi^2 > 15.99] = 0.10$

ϕ \ α	0.995	0.99	0.975	0.95	0.9	0.75	0.5	0.25	0.1	0.05	0.025	0.01	0.005	ϕ \ α
1	3.93E-05	1.57E-04	9.82E-04	3.93E-03	1.58E-02	0.102	0.455	1.323	2.71	3.84	5.02	6.63	7.88	1
2	1.00E-02	2.01E-02	5.06E-02	0.103	0.211	0.575	1.386	2.77	4.61	5.99	7.38	9.21	10.60	2
3	7.17E-02	0.115	0.216	0.352	0.584	1.213	2.37	4.11	6.25	7.81	9.35	11.34	12.84	3
4	0.207	0.297	0.484	0.711	1.064	1.923	3.36	5.39	7.78	9.49	11.14	13.28	14.86	4
5	0.412	0.554	0.831	1.145	1.610	2.67	4.35	6.63	9.24	11.07	12.83	15.09	16.75	5
6	0.676	0.872	1.237	1.635	2.20	3.45	5.35	7.84	10.64	12.59	14.45	16.81	18.55	6
7	0.989	1.239	1.690	2.17	2.83	4.25	6.35	9.04	12.02	14.07	16.01	18.48	20.3	7
8	1.344	1.647	2.18	2.73	3.49	5.07	7.34	10.22	13.36	15.51	17.53	20.1	22.0	8
9	1.735	2.09	2.70	3.33	4.17	5.90	8.34	11.39	14.68	16.92	19.02	21.7	23.6	9
10	2.16	2.56	3.25	3.94	4.87	6.74	9.34	12.55	15.99	18.31	20.5	23.2	25.2	10
11	2.60	3.05	3.82	4.57	5.58	7.58	10.34	13.70	17.28	19.68	21.9	24.7	26.8	11
12	3.07	3.57	4.40	5.23	6.30	8.44	11.34	14.85	18.55	21.0	23.3	26.2	28.3	12
13	3.57	4.11	5.01	5.89	7.04	9.30	12.34	15.98	19.81	22.4	24.7	27.7	29.8	13
14	4.07	4.66	5.63	6.57	7.79	10.17	13.34	17.12	21.1	23.7	26.1	29.1	31.3	14
15	4.60	5.23	6.26	7.26	8.55	11.04	14.34	18.25	22.3	25.0	27.5	30.6	32.8	15
16	5.14	5.81	6.91	7.96	9.31	11.91	15.34	19.37	23.5	26.3	28.8	32.0	34.3	16
17	5.70	6.41	7.56	8.67	10.09	12.79	16.34	20.5	24.8	27.6	30.2	33.4	35.7	17
18	6.26	7.01	8.23	9.39	10.86	13.68	17.34	21.6	26.0	28.9	31.5	34.8	37.2	18
19	6.84	7.63	8.91	10.12	11.65	14.56	18.34	22.7	27.2	30.1	32.9	36.2	38.6	19
20	7.43	8.26	9.59	10.85	12.44	15.45	19.34	23.8	28.4	31.4	34.2	37.6	40.0	20
21	8.03	8.90	10.28	11.59	13.24	16.34	20.3	24.9	29.6	32.7	35.5	38.9	41.4	21
22	8.64	9.54	10.98	12.34	14.04	17.24	21.3	26.0	30.8	33.9	36.8	40.3	42.8	22
23	9.26	10.20	11.69	13.09	14.85	18.14	22.3	27.1	32.0	35.2	38.1	41.6	44.2	23
24	9.89	10.86	12.40	13.85	15.66	19.04	23.3	28.2	33.2	36.4	39.4	43.0	45.6	24
25	10.52	11.52	13.12	14.61	16.47	19.94	24.3	29.3	34.4	37.7	40.6	44.3	46.9	25
26	11.16	12.20	13.84	15.38	17.29	20.8	25.3	30.4	35.6	38.9	41.9	45.6	48.3	26
27	11.81	12.88	14.57	16.15	18.11	21.7	26.3	31.5	36.7	40.1	43.2	47.0	49.6	27
28	12.46	13.56	15.31	16.93	18.94	22.7	27.3	32.6	37.9	41.3	44.5	48.3	51.0	28
29	13.12	14.26	16.05	17.71	19.77	23.6	28.3	33.7	39.1	42.6	45.7	49.6	52.3	29
30	13.79	14.95	16.79	18.49	20.6	24.5	29.3	34.8	40.3	43.8	47.0	50.9	53.7	30
40	20.7	22.2	24.4	26.5	29.1	33.7	39.3	45.6	51.8	55.8	59.3	63.7	66.8	40
50	28.0	29.7	32.4	34.8	37.7	42.9	49.3	56.3	63.2	67.5	71.4	76.2	79.5	50
60	35.5	37.5	40.5	43.2	46.5	52.3	59.3	67.0	74.4	79.1	83.3	88.4	92.0	60
70	43.3	45.4	48.8	51.7	55.3	61.7	69.3	77.6	85.5	90.5	95.0	100.4	104.2	70
80	51.2	53.5	57.2	60.4	64.3	71.1	79.3	88.1	96.6	101.9	106.6	112.3	116.3	80
90	59.2	61.8	65.6	69.1	73.3	80.6	89.3	98.6	107.6	113.1	118.1	124.1	128.3	90
100	67.3	70.1	74.2	77.9	82.4	90.1	99.3	109.1	118.5	124.3	129.6	135.8	140.2	100
Z_α	-2.58	-2.33	-1.96	-1.64	-1.28	-0.674	0.000	0.674	1.282	1.645	1.96	2.33	2.58	Z_α

Para $\phi > 100$ tómese $\chi^2 = \frac{1}{2}\left(Z_\alpha + \sqrt{2\phi - 1}\right)^2$. Z_α es la desviación normal estandarizada correspondiente al nivel de significancia y se muestra en la parte superior de la tabla.

Tabla C.3 – Tabla de Distribución t de Student

Puntos de porcentaje de la distribución t



Ejemplo

Para $\phi = 10$ grados de libertad:

$P[t > 1.812] = 0.05$
$P[t < -1.812] = 0.05$

r \ α	0,25	0,2	0,15	0,1	0,05	0,025	0,01	0,005	0,0005
1	1,000	1,376	1,963	3,078	6,314	12,706	31,821	63,656	636,578
2	0,816	1,061	1,386	1,886	2,920	4,303	6,965	9,925	31,600
3	0,765	0,978	1,250	1,638	2,353	3,182	4,541	5,841	12,924
4	0,741	0,941	1,190	1,533	2,132	2,776	3,747	4,604	8,610
5	0,727	0,920	1,156	1,476	2,015	2,571	3,365	4,032	6,869
6	0,718	0,906	1,134	1,440	1,943	2,447	3,143	3,707	5,959
7	0,711	0,896	1,119	1,415	1,895	2,365	2,998	3,499	5,408
8	0,706	0,889	1,108	1,397	1,860	2,306	2,896	3,355	5,041
9	0,703	0,883	1,100	1,383	1,833	2,262	2,821	3,250	4,781
10	0,700	0,879	1,093	1,372	1,812	2,228	2,764	3,169	4,587
11	0,697	0,876	1,088	1,363	1,796	2,201	2,718	3,106	4,437
12	0,695	0,873	1,083	1,356	1,782	2,179	2,681	3,055	4,318
13	0,694	0,870	1,079	1,350	1,771	2,160	2,650	3,012	4,221
14	0,692	0,868	1,076	1,345	1,761	2,145	2,624	2,977	4,140
15	0,691	0,866	1,074	1,341	1,753	2,131	2,602	2,947	4,073
16	0,690	0,865	1,071	1,337	1,746	2,120	2,583	2,921	4,015
17	0,689	0,863	1,069	1,333	1,740	2,110	2,567	2,898	3,965
18	0,688	0,862	1,067	1,330	1,734	2,101	2,552	2,878	3,922
19	0,688	0,861	1,066	1,328	1,729	2,093	2,539	2,861	3,883
20	0,687	0,860	1,064	1,325	1,725	2,086	2,528	2,845	3,850
21	0,686	0,859	1,063	1,323	1,721	2,080	2,518	2,831	3,819
22	0,686	0,858	1,061	1,321	1,717	2,074	2,508	2,819	3,792
23	0,685	0,858	1,060	1,319	1,714	2,069	2,500	2,807	3,768
24	0,685	0,857	1,059	1,318	1,711	2,064	2,492	2,797	3,745
25	0,684	0,856	1,058	1,316	1,708	2,060	2,485	2,787	3,725
26	0,684	0,856	1,058	1,315	1,706	2,056	2,479	2,779	3,707
27	0,684	0,855	1,057	1,314	1,703	2,052	2,473	2,771	3,689
28	0,683	0,855	1,056	1,313	1,701	2,048	2,467	2,763	3,674
29	0,683	0,854	1,055	1,311	1,699	2,045	2,462	2,756	3,660
30	0,683	0,854	1,055	1,310	1,697	2,042	2,457	2,750	3,646
40	0,681	0,851	1,050	1,303	1,684	2,021	2,423	2,704	3,551
60	0,679	0,848	1,045	1,296	1,671	2,000	2,390	2,660	3,460
120	0,677	0,845	1,041	1,289	1,658	1,980	2,358	2,617	3,373
∞	0,674	0,842	1,036	1,282	1,645	1,960	2,326	2,576	3,290

Tabla C.4 – Tabla de Distribución de Durbin Watson. Nivel de Significancia del 5%

n	$k' = 1$		$k' = 2$		$k' = 3$		$k' = 4$		$k' = 5$	
	d_i	d_s	d_i	d_s	d_i	d_s	d_i	d_s	d_i	d_s
15	1,08	1,36	0,95	1,54	0,82	1,75	0,69	1,97	0,56	2,21
16	1,10	1,37	0,98	1,54	0,86	1,73	0,74	1,93	0,62	2,15
17	1,13	1,38	1,02	1,54	0,90	1,71	0,78	1,90	0,67	2,10
18	1,16	1,39	1,05	1,53	0,93	1,69	0,82	1,87	0,71	2,06
19	1,18	1,40	1,08	1,53	0,97	1,68	0,86	1,85	0,75	2,02
20	1,20	1,41	1,10	1,54	1,00	1,68	0,90	1,83	0,79	1,99
21	1,22	1,42	1,13	1,54	1,03	1,67	0,93	1,81	0,83	1,96
22	1,24	1,43	1,15	1,54	1,05	1,66	0,96	1,80	0,86	1,94
23	1,26	1,44	1,17	1,54	1,08	1,66	0,99	1,79	0,90	1,92
24	1,27	1,45	1,19	1,55	1,10	1,66	1,01	1,78	0,93	1,90
25	1,29	1,45	1,21	1,55	1,12	1,66	1,04	1,77	0,95	1,89
26	1,30	1,46	1,22	1,55	1,14	1,65	1,06	1,76	0,98	1,88
27	1,32	1,47	1,24	1,56	1,16	1,65	1,08	1,76	1,01	1,86
28	1,33	1,48	1,26	1,56	1,18	1,65	1,10	1,75	1,03	1,85
29	1,34	1,48	1,27	1,56	1,20	1,65	1,12	1,74	1,05	1,84
30	1,35	1,49	1,28	1,57	1,21	1,65	1,14	1,74	1,07	1,83
31	1,36	1,50	1,30	1,57	1,23	1,65	1,16	1,74	1,09	1,83
32	1,37	1,50	1,31	1,57	1,24	1,65	1,18	1,73	1,11	1,82
33	1,38	1,51	1,32	1,58	1,26	1,65	1,19	1,73	1,13	1,81
34	1,39	1,51	1,33	1,58	1,27	1,65	1,21	1,73	1,15	1,81
35	1,40	1,52	1,34	1,58	1,28	1,65	1,22	1,73	1,16	1,80
36	1,41	1,52	1,35	1,59	1,29	1,65	1,24	1,73	1,18	1,80
37	1,42	1,53	1,36	1,59	1,31	1,66	1,25	1,72	1,19	1,80
38	1,43	1,54	1,37	1,59	1,32	1,66	1,26	1,72	1,21	1,79
39	1,43	1,54	1,38	1,60	1,33	1,66	1,27	1,72	1,22	1,79
40	1,44	1,54	1,39	1,60	1,34	1,66	1,29	1,72	1,23	1,79
45	1,48	1,57	1,43	1,62	1,38	1,67	1,34	1,72	1,29	1,78
50	1,50	1,59	1,46	1,63	1,42	1,67	1,38	1,72	1,34	1,77
55	1,53	1,60	1,49	1,64	1,45	1,68	1,41	1,72	1,38	1,77
60	1,55	1,62	1,51	1,65	1,48	1,69	1,44	1,73	1,41	1,77
65	1,57	1,63	1,54	1,66	1,50	1,70	1,47	1,73	1,44	1,77
70	1,58	1,64	1,55	1,67	1,52	1,70	1,49	1,74	1,46	1,77
75	1,60	1,65	1,57	1,68	1,54	1,71	1,51	1,74	1,49	1,77
80	1,61	1,66	1,59	1,69	1,56	1,72	1,53	1,74	1,51	1,77
85	1,62	1,67	1,60	1,70	1,57	1,72	1,55	1,75	1,52	1,77
90	1,63	1,68	1,61	1,70	1,59	1,73	1,57	1,75	1,54	1,78
95	1,64	1,69	1,62	1,71	1,60	1,73	1,58	1,75	1,56	1,78
100	1,65	1,69	1,63	1,72	1,61	1,74	1,59	1,76	1,57	1,78

n = número de observaciones.
k' = número de variables explicativas, excluyendo el término constante.

Tabla C.5 – Tabla de Distribución de Durbin Watson. Nivel de Significancia del 1%

n	$k' = 1$		$k' = 2$		$k' = 3$		$k' = 4$		$k' = 5$	
	d_i	d_s	d_i	d_s	d_i	d_s	d_i	d_s	d_i	d_s
15	0,81	1,07	0,70	1,25	0,59	1,46	0,49	1,70	0,39	1,96
16	0,84	1,09	0,74	1,25	0,63	1,44	0,53	1,66	0,44	1,90
17	0,87	1,10	0,77	1,25	0,67	1,43	0,57	1,63	0,48	1,85
18	0,90	1,12	0,80	1,26	0,71	1,42	0,61	1,60	0,52	1,80
19	0,93	1,13	0,83	1,26	0,74	1,41	0,65	1,58	0,56	1,77
20	0,95	1,15	0,86	1,27	0,77	1,41	0,68	1,57	0,60	1,74
21	0,97	1,16	0,89	1,27	0,80	1,41	0,72	1,55	0,63	1,71
22	1,00	1,17	0,91	1,28	0,83	1,40	0,75	1,54	0,66	1,69
23	1,02	1,19	0,94	1,29	0,86	1,40	0,77	1,53	0,70	1,67
24	1,04	1,20	0,96	1,30	0,88	1,41	0,80	1,53	0,72	1,66
25	1,05	1,21	0,98	1,30	0,90	1,41	0,83	1,52	0,75	1,65
26	1,07	1,22	1,00	1,31	0,93	1,41	0,85	1,52	0,78	1,64
27	1,09	1,23	1,02	1,32	0,95	1,41	0,88	1,51	0,81	1,63
28	1,10	1,24	1,04	1,32	0,97	1,41	0,90	1,51	0,83	1,62
29	1,12	1,25	1,05	1,33	0,99	1,42	0,92	1,51	0,85	1,61
30	1,13	1,26	1,07	1,34	1,01	1,42	0,94	1,51	0,88	1,61
31	1,15	1,27	1,08	1,34	1,02	1,42	0,96	1,51	0,90	1,60
32	1,16	1,28	1,10	1,35	1,04	1,43	0,98	1,51	0,92	1,60
33	1,17	1,29	1,11	1,36	1,05	1,43	1,00	1,51	0,94	1,59
34	1,18	1,30	1,13	1,36	1,07	1,43	1,01	1,51	0,95	1,59
35	1,19	1,31	1,14	1,37	1,08	1,44	1,03	1,51	0,97	1,59
36	1,21	1,32	1,15	1,38	1,10	1,44	1,04	1,51	0,99	1,59
37	1,22	1,32	1,16	1,38	1,11	1,45	1,06	1,51	1,00	1,59
38	1,23	1,33	1,18	1,39	1,12	1,45	1,07	1,52	1,02	1,58
39	1,24	1,34	1,19	1,39	1,14	1,45	1,09	1,52	1,03	1,58
40	1,25	1,34	1,20	1,40	1,15	1,46	1,10	1,52	1,05	1,58
45	1,29	1,38	1,24	1,42	1,20	1,48	1,16	1,53	1,11	1,58
50	1,32	1,40	1,28	1,45	1,24	1,49	1,20	1,54	1,16	1,59
55	1,36	1,43	1,32	1,47	1,28	1,51	1,25	1,55	1,21	1,59
60	1,38	1,45	1,35	1,48	1,32	1,52	1,28	1,56	1,25	1,60
65	1,41	1,47	1,38	1,50	1,35	1,53	1,31	1,57	1,28	1,61
70	1,43	1,49	1,40	1,52	1,37	1,55	1,34	1,58	1,31	1,61
75	1,45	1,50	1,42	1,53	1,39	1,56	1,37	1,59	1,34	1,62
80	1,47	1,52	1,44	1,54	1,42	1,57	1,39	1,60	1,36	1,62
85	1,48	1,53	1,46	1,55	1,43	1,58	1,41	1,60	1,39	1,63
90	1,50	1,54	1,47	1,56	1,45	1,59	1,43	1,61	1,41	1,64
95	1,51	1,55	1,49	1,57	1,47	1,60	1,45	1,62	1,42	1,64
100	1,52	1,56	1,50	1,58	1,48	1,60	1,46	1,63	1,44	1,65

n = número de observaciones.
k' = número de variables explicativas, excluyendo el término constante.

ANEXO – D Programa: Test_Mag_Data_NARX.m (Redes Neuronales)

```
cont=1;                          % Inicializa contador para repetir el programa
while cont==1
   close,clear,clc,plt = 0;      % Encera y borra parámetros
   % 1.- Lectura de datos
   S=load('magdata');
   T = con2seq(S.y);             % Conversión a vector secuencial para la red
   X = con2seq(S.u);
   % 2.- Preparación de datos
   N=input('Ingrese número de Pasos a Predecir:   ');
   % X,T son divididos en dos grupos: el 1ero para entrenamiento
   inputSeries= X(1:end-N);
   targetSeries= T(1:end-N);
   % El 2do grupo se utilizará para validar la red
   inputSeriesVal = X(end-N+1:end);
   targetSeriesVal = T(end-N+1:end);
   % 3.- Arquitectura de la Red
   delay = input('Ingrese número de retardos:   ');
   neurons=input('Ingrese número de Neuronas en la capa oculta:   ');
   iterac=input('Ingrese número de Iteraciones para el entrenamiento:  ');
   % 4.- Creación de la Red
   trainFcn = 'trainbr';         % Entrenamiento con Reg. Bayesiana
   % trainFcn = 'trainlm';       % Algoritmo Levenberg-Marquardt
   net = narxnet(1:delay,1:delay,neurons,'open',trainFcn); % Creación de la red
   view(net)                     % Verificación de la Red en Pantalla
   %net.divideFcn = 'dividetrain';  % Todos los datos para entrenamiento
   net.divideFcn = 'dividerand';   % Se divide los datos aleatoriamente, activa
   net.trainParam.min_grad = 1e-6; % validación
   % 5.- Entrenamiento de la red
   net.trainParam.epochs=iterac;   % Número de iteraciones para terminar entrenamiento
   [Xs,Xi,Ai,Ts] = preparets(net,inputSeries,{},targetSeries);
   net = init(net);              % Inicializa pesos y bías de la red
   net = train(net,Xs,Ts,Xi,Ai);    % Entrenamiento de la Red
   Y = net(Xs,Xi,Ai);            % Salida de red entrenada en vector Y
   % Transformación a red en lazo cerrado
   % Se utilizará esta red para hacer predicciones a futuro
   netc = closeloop(net);        % Realiza la realimentación desde la salida
   view(netc)                    % Verifica si la red está en lazo cerrado
   [Xc,Xic,Aic,Tc] = preparets(netc,inputSeries,{},targetSeries);
   Yc = netc(Xc,Xic,Aic);        % Salida de la red en lazo cerrado en vector Yc
   % 6.- Predicción varios pasos adelante
   inputSeriesPred = [inputSeries(end-delay+1:end),inputSeriesVal];
                                 % Toma los retardos (end-delay+1)
   targetSeriesPred = [targetSeries(end-delay+1:end),con2seq(nan(1,N))];
                                 % Toma Retardos y adiciona celdas en blanco NaN
   [Xsp,Xip,Aip,Tp]= preparets(netc,inputSeriesPred,{},targetSeriesPred);
                                 % Perpara datos para predicción a futuro
   yPred=netc(Xsp,Xip,Aip);       % Pronóstico mediante red en lazo cerrado

plot([cell2mat(targetSeries),nan(1,N);nan(1,length(targetSeries)),cell2mat(yPred);nan(1,length(targetSeries)),
cell2mat(targetSeriesVal)]');
   legend('Destinos Originales','Predicción de la Red','Salida Esperada');
   cont=input('Desea realizar una nueva iteración SI=1; NO=0  :');
end
disp('El programa ha terminado');
```

ANEXO –E Programa: Prog_Narnet_Tesis_OK.m (Redes Neuronales)

```
cont=1;                                  % Inicializa contador para repetir el programa
while cont==1                            % para asegurar mínimo global
   close,clear,clc,plt = 0;              % Encera y borra los parámetros
   % 1.- Lectura de Datos
   [nombre, path]=uigetfile('*.*','Seleccione archivo de datos a Pronosticar');
   nombrearchivo=strcat(path,nombre);    % une directorio y nombre del archivo
   if nombre==0                          % Si presiona cancel, finaliza
   return
   end
   yi = xlsread(nombrearchivo);          % Carga datos de archivo Excel en yi
   T = con2seq(yi);                      % Conversión a vector secuencial para la red
   NT = length(T);                       % Número total de datos en NT
   N=input('Ingrese número de Pasos para Predecir a Futuro:   ');
   % 2.- Preparación de datos
   % T es dividido en dos grupos: el 1ero para entrenamiento
   targetSeries = T(1:end-N);
   % El 2do grupo se utilizará para validar la red
   targetSeriesVal = T(end-N+1:end);     % Estos datos normalmente no están disponibles
   % 3.- Arquitectura de la Red
   delay = input('Ingrese número de retardos:   ');
   neurons=input('Ingrese número de neuronas en la capa culta:   ');
   iterac=input('Ingrese número de iteraciones para el entrenamiento:  ');
   % 4.- Creación de la Red Narnet
   trainFcn = 'trainbr';                 % Entrenamiento Bayesiano
   %trainFcn = 'trainlm';                % Algortimo Levenberg-Marquardt
   net = narnet(1:delay,neurons,'open',trainFcn);
   view(net)                             % Esquema de red en pantalla
   net.trainParam.epochs=iterac;         % Número de Iteraciones para finalizar entrenamiento
   %net.divideFcn = 'dividerand';        % Divide datos aleatoriamente
   net.divideFcn = 'dividetrain';        % Toma todos los datos para entrenamiento
   net.trainParam.min_grad = 1e+1;       %Si gradiente menor a 1e+1 se detiene
   % 5.- Entrenamiento de la red
    [Xs,Xi,Ai,Ts] = preparets(net,{},{},targetSeries);          % Desplaza datos para entrenamiento
    net = init(net);                     %Inicializa Pesos y Bías de la Red
    net = train(net,Xs,Ts,Xi,Ai);        % Entrenamiento de la Red
    Y = net(Xs,Xi,Ai);                   % Salida de la Red Entrenada
    % Red de lazo Cerrado
    netc = closeloop(net);               % Se utilizará esta red para hacer predicciones a futuro
    view(netc)                           % Esquema de Red para verificar si se cerró el lazo
    [Xc,Xic,Aic,Tc] = preparets(netc,{},{},targetSeries);       % Prepara datos para red en lazo cerrado
    Yc = netc(Xc,Xic,Aic);               % Salida de la red de lazo cerrado
    % 6.- Predicción varios pasos adelante
    targetSeriesPred = [targetSeries(end-delay+1:end),con2seq(nan(1,N))];
                                         % Toma Retardos y adiciona celdas en blanco (NaN)
    [Xsp,Xip,Aip,Tp]= preparets(netc,{},{},targetSeriesPred);
                                         % Preapra datos para predicción varios pasos adelante
    yPred=netc(Xsp,Xip,Aip);             % Predicción mediante la red en lazo cerrado
    % 7.- Gráficos de la predicción varios pasos adelante
plot([cell2mat(targetSeries),nan(1,N);nan(1,length(targetSeries)),cell2mat(yPred);nan(1,length(targetSeries)),
cell2mat(targetSeriesVal)]');
    legend('Destinos Originales','Predicción de la Red','Salida Esperada');
    title( 'Resultados Red Lazo Cerrado' );
    cont=input('Desea realizar una nueva iteración SI=1; NO=0  :');
end
disp('El programa ha terminado');
```

APÉNDICES