REDES NEURONALES Y ALGORITMOS GENETICOS

J. Alberto Delgado R.

Profesor Asistente
Departamento de Ingeniería Eléctrica
Universidad Nacional de Colombia

ABSTRACT

This paper presents the training of a neural network with a genetical algorithm. The effects due to the size of chromosomes and gens are discussed for the learning of the XOR problem.

RESUMEN

Este artículo presenta el entrenamiento de una red neuronal con un algoritmo genético. Se analizan los efectos debidos al número de cromosomas y al tamaño de los genes durante el aprendizaje de la función lógica XOR.

Los algoritmos genéticos pueden ser una alternativa para entrenar redes neuronales con función de activación tipo limitador duro y pesos discretos.

INTRODUCCION

El algoritmo de propagación inversa (BP) se ha estudiado [1] para una red neuronal con pesos discretos durante el aprendizaje de la función lógica XOR. Este tipo de red es consecuencia natural del desarrollo digital VLSI donde las conexiones toman valores binarios.

Los resultados experimentales [1] muestran que para lograr 60% a 70% de convergencia, la red requiere más de 1000 niveles discretos, lo cual equivale a 10 bits de almacenamiento por peso de interconexión. Si el número de niveles discretos es menor al umbral (300) la red no aprende, debido a que la corrección en los pesos no es lo suficientemente grande para compensar el efecto de la discretización.

Los algoritmos genéticos son una alternativa de entrenamiento para redes neuronales con pesos discretos, puesto que se logra el aprendizaje con pocos niveles. En el caso de la función lógica XOR son suficientes cuatro niveles discretos por peso, lo cual equivale a dos bits.

NEURONA ARTIFICIAL

Un modelo simple de la dinámica Entrada/Salida (E/S) de la neurona natural se observa en la figura 1.

La salida de este modelo satisface la ecuación,

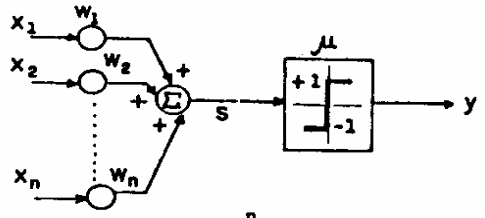
$$y=f(\sum_{i=1}^{n} \omega_i . x_i - \mu) \quad (1)$$

donde,

$$f(a) = \begin{cases} +1, & \text{si } a \geq 0 \\ -1, & \text{si } a < 0 \end{cases}$$

μ : Umbral
xi : Señales de Entrada
wi : Pesos de cada Conexión

En palabras, la salida de la neurona toma el valor constante de +1 cuando la suma ponderada de las entradas excede el umbral μ. En caso contrario, y = -1.



$$y = \theta\left(\sum_{K=1}^{n} W_K . X_K - \mu \right)$$

Fig.1.- Modelo E/S de la neurona natural.

La ecuación (1) se puede escribir como,

$$y=f(\sum_{i=0}^{n} \omega_i . x_i) \quad (2)$$

donde,

x0 : + 1
w0 = − μ : Peso Umbral

RED NEURONAL [2], [3]

La figura 2 muestra una red neuronal multicapa, en notación simplificada es una red 2-2-1.

La capa de entrada no realiza procesamiento, simplemente recibe las entradas y las distribuye a la capa oculta.

La capa oculta y la capa de salida realizan procesamiento y contienen neuronas como la señalada en la figura 1.

Las capas de entrada y salida se conectan al exterior, mientras la capa del medio no. Por ello se denomina a esta última, capa oculta.

Durante el entrenamiento de la red, los pesos se modifican para satisfacer la relación: entradas → salida, (X1, X2) → (Y). En el aprendizaje de la función lógica XOR hay cuatro relaciones o patrones.

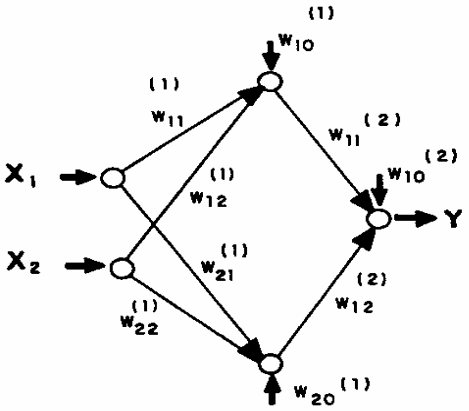


Fig.2.- Red neuronal multicapa 2-2-1.

El peso,

$$\omega_{ij}^{(l)}$$

se interpreta como el valor de la conexión que llega a la neurona i de la capa l, desde la neurona j de la capa l-1. Los pesos con el índice j=0 corresponden a pesos umbral, y conectan la neurona a una entrada con valor constante de +1.

ALGORITMOS GENETICOS [4], [5]

Los algoritmos genéticos (AGs) se basan en la mecánica de la selección natural. Utilizan la supervivencia de las cadenas mejor adaptadas para buscar un óptimo.

Las diferencias entre los algoritmos genéticos y otros métodos de optimización son:

- Los AGs trabajan con un código correspondiente a los parámetros, no con los parámetros. Esto es, los parámetros se codifican como una cadena de longitud finita sobre un alfabeto finito.

- Los AGs realizan la búsqueda desde varios puntos del espacio y no desde una condición inicial.

- Los AGs explotan sólamente la información de la función objetivo.

- Los AGs utilizan reglas de transición probabilísticas.

Para entrenar la red neuronal, el conjunto de pesos se codifica como una cadena binaria llamada cromosoma. Cada cromosoma produce un error sobre todos los patrones de entrenamiento, el objetivo es minimizar este error o maximizar su recíproco.

La función de bondad a maximizar es,

$$g=\frac{1}{\sum_{i=1}^{np}(y_i-d_i)^2} \quad (3)$$

donde,

np : Número de patrones
y_i : Salida de la red para cada patrón
d_i : Salida deseada de la red

El proceso inicia con una población aleatoria de cadenas y se construyen generaciones sucesivas usando los operadores genéticos de reproducción y cruce. Las cadenas mejores, según (3), tienen mayor probabilidad de sobrevivir y participar en cruces. El cruce combina parte de una cadena con otra y puede reunir buenos segmentos hallados al azar en la población.

Cada cromosoma es una cadena de bits correspondiente a los pesos, según (4). Cada peso se representa en binario complemento a dos y tiene su equivalente decimal.

$$\omega_{10}^{(1)}\omega_{11}^{(1)}\omega_{12}^{(1)}\omega_{20}^{(1)}\omega_{21}^{(1)}\omega_{22}^{(1)}\omega_{10}^{(2)}\omega_{11}^{(2)}\omega_{12}^{(2)} \quad (4)$$

El entrenamiento se realiza en función del tamaño de la población (número de cadenas: nc) y del tamaño de los genes (número de bits por peso: nb).

ENTRENAMIENTO

La Tabla 1 es la función lógica XOR que debe aprender la red neuronal de la figura 2. Note el uso de valores bipolares {-1,+1} en lugar de {0,+1}.

Tabla 1.- Función Lógica XOR.

X1	X2	Y
- 1	- 1	- 1
- 1	+ 1	+ 1
+ 1	- 1	+ 1
+ 1	+ 1	- 1

Se considera que no hay convergencia durante el entrenamiento cuando el número de iteraciones es mayor que diez. Cada iteración corresponde a un recorrido por la Tabla 1.

La figura 3 muestra el porcentaje de convergencias como función del número de cromosomas (nc) y del número de bits por peso (nb). Note que el número de aprendizajes aumenta con el tamaño de la población, esto es, a mayor diversidad hay más oportunidad de hallar el óptimo.

La figura 4 presenta el promedio de iteraciones por convergencia como función del tamaño de la población y de los genes. El comportamiento es bastante irregular, se observa para ocho cromosomas que las iteraciones promedio disminuyen con el tamaño de los genes. Es decir, un gene de mayor tamaño equivale a mayor número de niveles discretos para cada peso y por lo tanto hay más posibilidad de aprendizaje.

El punto correspondiente a diez cromosomas y seis bits por gene produce el menor número promedio de iteraciones, ello indica un rápido aprendizaje.

En la figura 5 se observa el tiempo promedio por convergencia como función del tamaño de la población y de los genes. Es claro, el incremento en tiempo de CPU con el aumento de la población y de los niveles discretos por peso (tamaño genes).

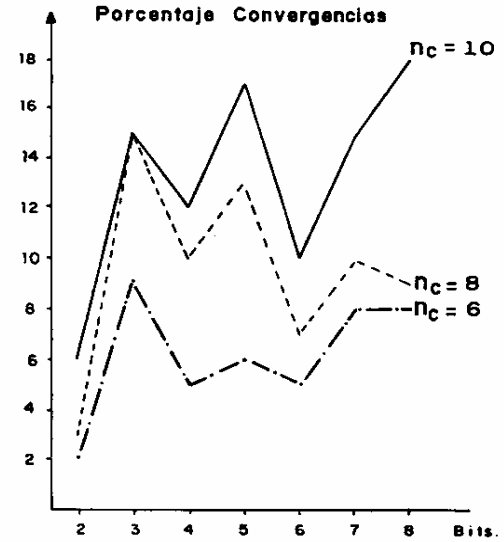


Fig.3.- Porcentaje de Convergencias como función del tamaño de la población y de los genes.

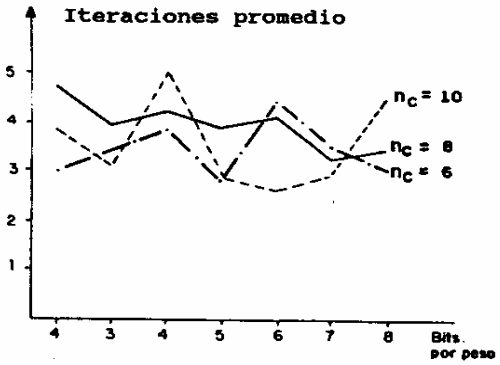


Fig.4.- Iteraciones promedio por convergencia como función del tamaño de la población y de los genes.

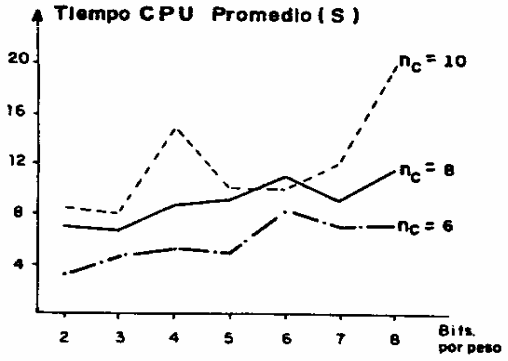


Fig.5.- Tiempo promedio por iteración como función del tamaño de la población y de los genes.

La figura 6 corresponde a seis redes neuronales multicapa 2-2-1 entrenadas con un algoritmo genético, la población es de diez cadenas y el tamaño de los genes es de dos bits.

CONCLUSIONES

Se ha mostrado la posibilidad de entrenar una red neuronal con un algoritmo genético. La ventaja es la minimización de la función objetivo a pesar del ruido de la superficie de los pesos.

Con el algoritmo genético es posible entrenar la red con 4 niveles discretos en contraste con 300 niveles como mínimo para el algoritmo de propagación inversa (BP).

Un inconveniente del algoritmo genético es el crecimiento del tiempo de aprendizaje con el aumento de la diversidad y del tamaño de los genes, lo cual podría ser una limitación para redes más grandes.

REFERENCIAS

[1] Von Lehmen, A., Paek, E.G., Liao, P.F., Marrakchi, A. y Patel, J.S.: "Factors Influencing Learning by Backpropagation", IEEE - ICNN, Vol. I, pp. 335-341, 1988.

[2] Widrow, B. y Lehr M.: "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", Proc. IEEE, Vol. 78, No.9, pp. 1415-1441, Septiembre 1985.

[3] Lippmann, R. P.: "An Introduction to Computing with Neural Nets", IEEE Acoustics Speech and Sinal Processing, pp. 4-22, Abril 1987.

[4] Goldberg, D. E.: "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, M.A., Addison-Wesley, 1988.

[5] Wayner, P.: "Genetic Algorithms", BYTE, pp. 361-368, Enero 1991.

BIOGRAFIA

J. Alberto Delgado R. es Ingeniero Eléctrico de la Universidad de los Andes y Magister en Control y Tratamiento de Señales de la misma Universidad.

Actualmente es Profesor Asistente en el Departamento de Ingeniería Eléctrica de la Universidad Nacional de Colombia. Sus áreas de interés son las redes neuronales y los algoritmos genéticos aplicados a la Ingeniería Eléctrica y Electrónica.

Dirección para correspondencia : A.A. No. 25268 Santafé de Bogotá, Colombia.

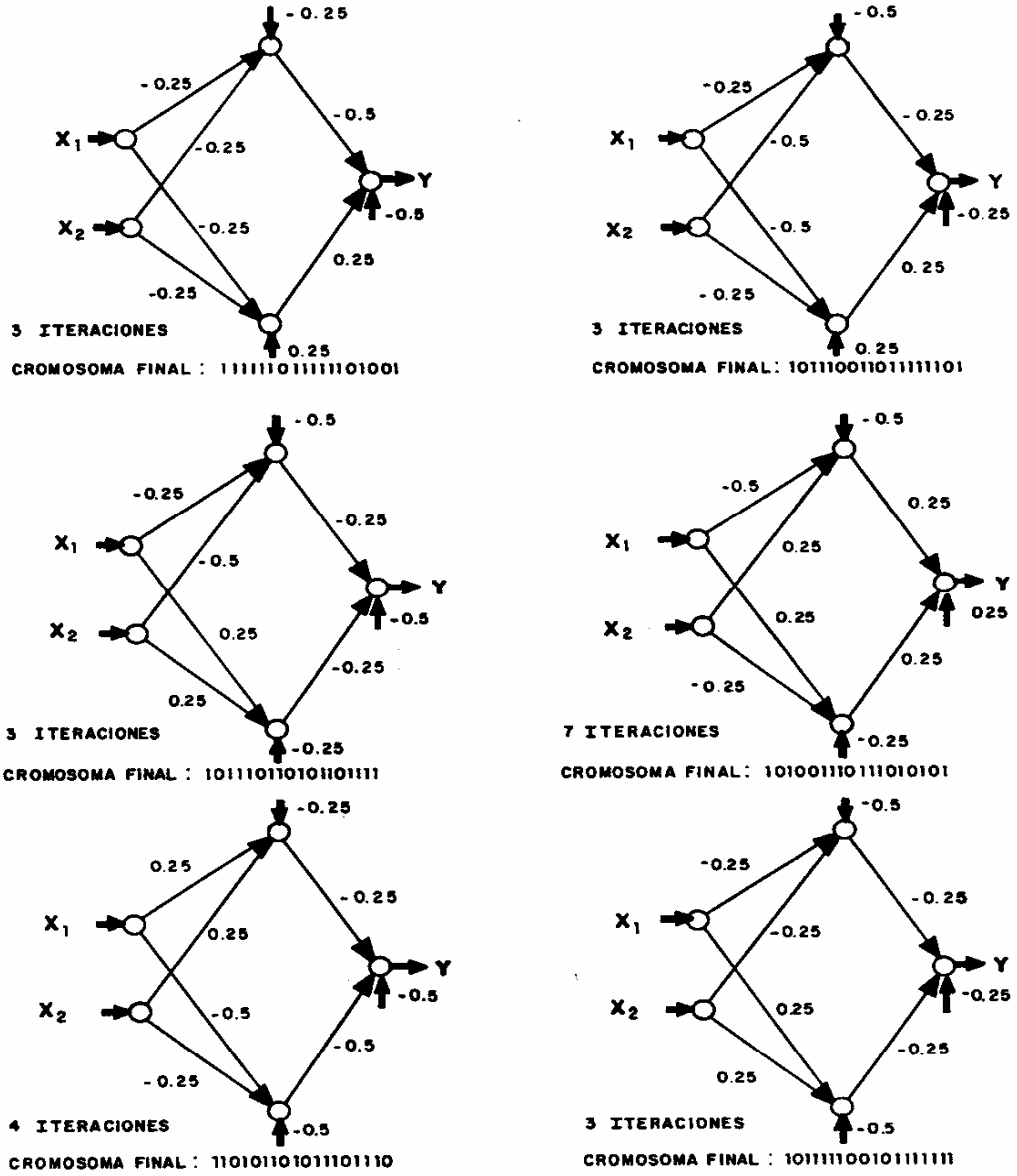


Fig.6.- Redes neuronales multicapa 2-2-1, entrenadas con un algoritmo genético para el problema del XOR.