Entrenamiento del Perceptrón

Perceptron Training

Autor: Cristian Camilo Manzano Calvo

Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira, Pereira, Colombia
Correo-e: ccmanzano@utp.edu.co

***Resumen*—** La modificación de los pesos y bias con un algoritmo de entrenamiento de un perceptrón multicapa es un problema clásico de programación no lineal irrestricto. El presente trabajo muestra el desempeño del "Simulated Annealing" como algoritmo de entrenamiento de esta red neuronal resolviendo dos problemas clásicos en redes neuronales, el problema de la "Codificación" y el problema de la "Doble Espiral". Resultados de buena calidad son obtenidos cuando se compara esta propuesta frente a algoritmos clásicos de entrenamiento.

***Palabras clave*—** Perceptrón, Inferencia, Red Neuronal, Aprendizaje, Inteligencia Artificial, Algoritmo, Neurona, Entrada, Salida.

***Abstract*—** This paper uses a Simulated Annealing algorithm for training an Artificial Neural Network. The efficiency of this algorithm is testing using two well know benchmark problems. Results are compared with BackPropagation algorithm and they show how this technique can resolve no-linear problem with this technique.

***Key Word* —** Perceptron, Inference, Neuronal Network, Learning, Artificial Intelligence, Algorithm, Neuron, Input, Output.

I. INTRODUCCIÓN

Una de las principales características de las redes neuronales es su capacidad para aprender a partir de alguna fuente de información interactuando con su entorno. El psicólogo Frank Rosenblat desarrolló un modelo simple de neurona basado en el modelo de McCulloch y Pitts y en una regla de aprendizaje basada en la corrección del error. A este modelo le llamó Perceptrón en 1958.

Una de las características que más interés despertó de este modelo fue la capacidad de aprender a reconocer patrones. El Perceptrón está constituido por conjunto de sensores de entrada que reciben los patrones de entrada a reconocer o clasificar y una neurona de salida que se ocupa de clasificar a los patrones de entrada en dos clases, según que la salida de la misma es binaria.

Sin embargo, este modelo tiene muchas limitaciones, como, por ejemplo, no es capaz de aprender la función lógica XOR; además tuvieron que pasar varios años hasta que se propusiera la regla de aprendizaje de retro propagación del error para demostrarse que el Perceptrón multicapa es un aproximador universal.

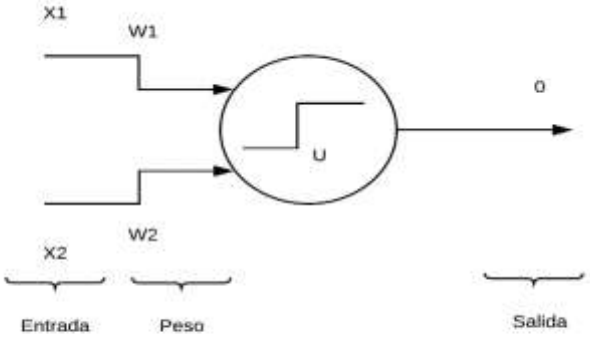
II. DESARROLLO DEL TEMA

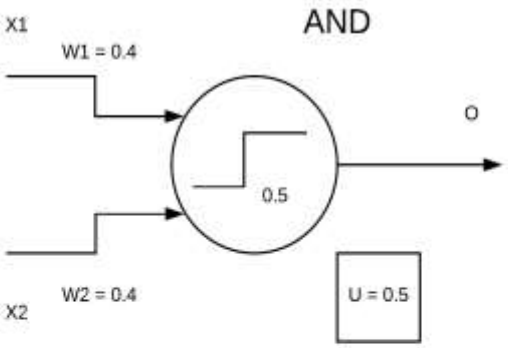
REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales buscan simular las tres características básicas de una red neuronal biológica [6]; procesamiento paralelo, memoria distribuida y adaptabilidad; lo anterior les permite a las redes neuronales artificiales aprender y generalizar a partir de un conjunto de datos de relación matemática desconocida. Existen muchas definiciones para las RNA, usando la planteada por se establece que una red neuronal artificial es un grafo dirigido que cumple con las siguientes propiedades: – A cada nodo i se le asocia una variable de estado yi – A cada conexión (i,j) de los nodos i y j se le asocia un peso sináptico wjidonde wjiR – A cada nodo j se le asocia un bias bjR – Para cada nodo j se define una función j(yi, wji, bj) que depende de los pesos, conexiones, bias y estados de los nodos i, conectados a él.

El proceso SA se formula sobre el concepto que cuando una estructura se calienta, sus moléculas empiezan a moverse con altas velocidades debido a la gran cantidad de energía, con el tiempo y por la pérdida de energía, dichas moléculas poseerán un movimiento más lento y se acomodaran permitiendo establecer estructuras cristalinas simétricas [5, 10]. El proceso SA se basa en el algoritmo de Metrópolis, el cual genera un conjunto de estructuras vecinas a partir de una estructura actual con sus respectivos niveles de energía; a su vez el algoritmo de metrópolis se basa en el método de Montecarlo que asigna una función de probabilidad de aceptar los cambios hechos en la estructura. Si se supone una estructura que se encuentra en el estado i a un nivel de energía Eiel cual es un indicativo de su comportamiento.

PROCESO





$$O = \begin{cases} 1 \text{ SI, } X1W1+X2W2 > U \\ ? \text{ SI, } X1W1+X2W2 = U \\ 0 \text{ SI, } X1W1+X2W2 < U \end{cases} \quad \sum$$

X1	X2	X1.W1	X2.W2	O
0	0	0	0	0
0	1	0	0.4	0
1	0	0.4	0	0
1	1	0.4	0.4	1

XOR

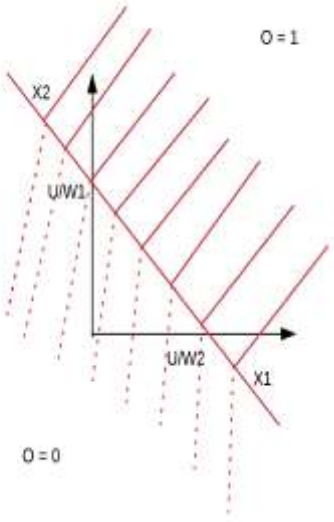
X1	X2	O
0	0	1
0	1	1
1	0	1
1	1	0

Ecuacion Limite:

X1W1+X2W2 = U

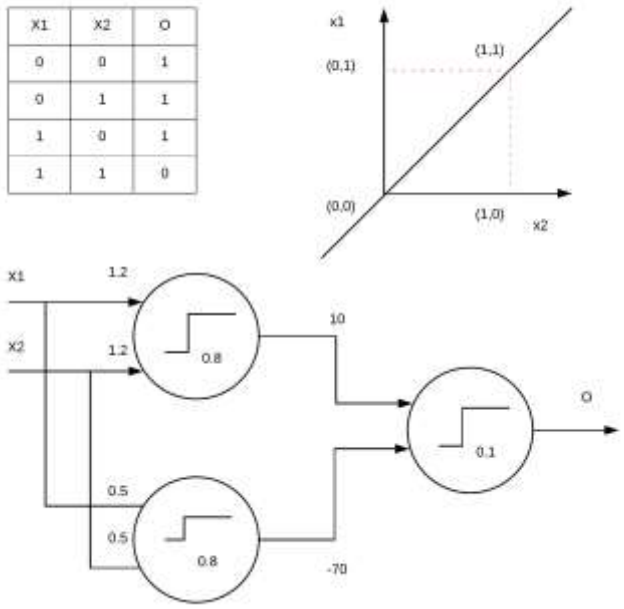
SI X2 = 0, X1 = U/W1

SI X1 = 0, X2 = U/W2



XOR

X1	X2	O
0	0	1
0	1	1
1	0	1
1	1	0



$$X1W1+X2W2 = 0$$

$$[\,X1 \;\; X2\,] \begin{bmatrix} W1 \\ W2 \end{bmatrix} = 0$$

$$\mathbf{X}.\mathbf{W} = 0$$

$$|\mathbf{X}|.|\mathbf{W}|\cos\alpha = 0$$

Según el Angulo que se forme entre X y W podemos organizar todos las coordenas en su correspondiente grupo

III. Algoritmo de aprendizaje del Perceptrón con entrenamiento individualizado

Paso 0: Inicialización
Inicializar los pesos sinápticos con números aleatorios del Intervalo [-1,1]. Ir al paso 1 con k=1

Paso 1: (k-ésima iteración)
Calcular

$$y(k) = \operatorname{sgn}\left(\sum_{j=1}^{n+1} w_j x_j(k)\right)$$

Paso 2: Corrección de los pesos sinápticos

Si z(k)≠y(k) modificar los pesos sinápticos según la Expresión:

$$w_j(k+1) = w_j(k) + \eta[z(k) - y(k)]x_j(k), \quad j = 1,2,...,n+1$$

Paso 3: Parada
Si no se han modificado los pesos en las últimas p Iteraciones, es decir,

$$w_j(r) = w_j(k), \quad j = 1,2,...,n+1, \; r = k+1,...,k+p,$$

Parar. La red se ha estabilizado.

En otro caso, ir al Paso 1 con k=k+1.

IV. CONCLUSIONES

- El perceptrón, a pesar de ser una de las redes más utilizadas, no es una de las más potentes ya que posee ciertas limitaciones, por ejemplo, el caso del aprendizaje en problemas complejos.
- Este tipo de redes se pueden implementar en la vida moderna en ámbitos como análisis de series temporales, procesamiento de imágenes, reconocimiento automático del habla, diagnósticos médicos, entre otros. [2]
- Aunque el perceptrón simple tiene algunas limitaciones como se pudieron ver en el desarrollo del documento como lo son: datos deben ser linealmente separables, y tasa pequeña de aprendizaje, esto no le quita la utilidad que este posee, y el avance que significo para la época.

REFERENCIAS

[1] A JavaScript Perceptron. (2015) Disponible en: https://planspace.org/20150610-a_javascript_perceptron/

[2] Perceptrón simple y multicapa. Disponible en: https://es.slideshare.net/Jeffo92/perceptrn-simple-y-multicapa

1. Las notas de pie de página deberán estar en la página donde se citan. Letra Times New Roman de 8 puntos