

Algoritmo de entrenamiento del perceptrón

Perceptron learning algorithm

Sebastián Rave Correa

Facultad de ingenierías, Universidad tecnológica de Pereira, Pereira, Colombia

Correo-e: sebastian.rave@utp.edu.co

Resumen— 12-15 renglones

Al inicio del desarrollo de las redes neuronales se basaron en el funcionamiento de las neuronas biológicas, por eso se estudia a profundidad el perceptrón que es la unidad básica de una red neuronal, el cual simula matemáticamente una neurona biológica. Podemos encontrar que el perceptrón cumple una función de clasificación, por lo que es un método muy utilizado en diferentes ámbitos. A partir de un algoritmo de entrenamiento el perceptrón comienza a tener más fuerza en cuanto a reconocimiento de patrones de entrada complejos. Veremos el algoritmo que permite dicho aprendizaje a partir de ciertos ejemplos de entrada que serán evaluados al momento de calcular alguna salida de la red trabajada con perceptrones.

Palabras clave— Entrenamiento, perceptrón, clasificación, patrones

Abstract— At the beginning of the development of neural networks were based on the functioning of biological neurons, so the perceptron is studied in depth, which is the basic unit of a neural network, which simulates a biological neuron mathematically. We can find that the perceptron fulfills a classification function, so it is a method widely used in different areas. From a training algorithm the perceptron begins to have more strength in recognition of complex entry patterns. We will see the algorithm that allows this learning from certain input examples that will be evaluated when calculating some output of the network worked with perceptrons.

Key Words — Learning, perceptron, classification, patterns

INTRODUCCIÓN

La primera red neuronal conocida, fue desarrollada en 1943 por Warren McCulloch y Walter Pitts; esta consistía en una suma de las señales de entrada, multiplicadas por unos valores de pesos escogidos aleatoriamente. La entrada es comparada con un patrón preestablecido para determinar la salida de la red. Si en la comparación, la suma de las entradas multiplicadas por los pesos es mayor o igual que el patrón preestablecido la salida de la red es uno (1), en caso contrario la salida es cero (0).

El perceptrón usa un algoritmo basado en aprendizaje automático, el cual utiliza una información dada, en forma de ejemplo para así generalizar comportamientos.

I. CONTENIDO

Que es un Perceptrón

El perceptrón es una representación matemática de una neurona biológica, la cual recibe ciertos parámetros de entrada y genera una salida. Es la unidad básica que compone a una red neuronal. Fue creada por Frank Rosenblatt en 1958.

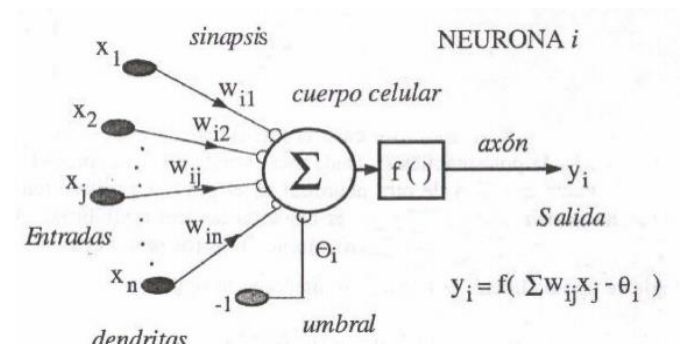


Figura 1

En la figura 1 podemos ver los componentes del perceptrón que es similar la neurona biológica.

A partir del perceptrón, mediante un algoritmo, podemos clasificar subgrupos de un grupo mayor de las posibles salidas que genera el perceptrón. El problema que podemos identificar en este modelo de un solo perceptrón, es que hay ciertos problemas los cuales no es posible hacer la clasificación, por ejemplo la función XOR.

Aprendizaje del perceptrón

A partir de unos ejemplos previamente evaluados podemos entrenar a un perceptrón. El reconocimiento de patrones es una de las principales aplicaciones de los perceptrones, dichos patrones son los linealmente separables (es decir, patrones que se encuentran a ambos lados de un hiperplano),

posteriormente clasificados en dos subgrupos los cuales pueden ser 1 o 0.

Inicialmente el perceptrón no fue diseñado para el reconocimiento de patrones de entrada muy complejos, pero se descubrió que mediante un proceso de aprendizaje era capaz de adquirir esta capacidad.

En esencia, el entrenamiento implicaba un proceso de refuerzo mediante el cual la salida de las unidades A se incrementaba o se decrementaba dependiendo de si las unidades A contribuían o no a las respuestas correctas del perceptrón para una entrada dada.

Como sabemos, un algoritmo de aprendizaje es un método adaptativo por el que una red se auto modifica para implementar el comportamiento deseado. Esto se hace presentando algunos ejemplos de la función entrada-salida a la red. Se presenta un ejemplo y se ejecuta una acción correctiva iterativamente hasta que la red aprende a producir la respuesta deseada. El conjunto de entrenamiento es el conjunto de los ejemplos de los que la red va a aprender.

Usamos la siguiente notación: El vector de entrada al perceptrón es $x = (x_1, \dots, x_n)$. Si los pesos son los valores reales w_1, \dots, w_n y el umbral es U , diremos que $w = (w_1, \dots, w_n, w_{n+1})$ con $w_{n+1} = -U$ es el vector extendido de pesos del perceptrón y que $(x_1, \dots, x_n, 1)$ es el vector extendido de entradas (se añade la entrada de tendencia o bias con valor 1 fijo). La computación de un perceptrón puede expresarse mediante un producto escalar, ya que $\sum x_i w_i \cdot U$ es equivalente a $w \cdot x \cdot 0$ donde w y x son los vectores extendidos de pesos y entradas, respectivamente.

La convergencia del algoritmo de aprendizaje del perceptrón se basa en que cada perceptrón realiza la comprobación $w \cdot x > 0$ ó $w \cdot x \cdot 0$, pero son equivalentes cuando el conjunto de entrenamiento es finito, lo cual siempre es cierto en problemas prácticos. Una forma habitual de comenzar el algoritmo de entrenamiento es inicializando aleatoriamente los pesos de la red y mejorar los parámetros iniciales, comprobando a cada paso, si puede lograrse una separación mejor del conjunto de entrenamiento. Cada vector de pesos w define un hiperplano que separa los puntos con salida 1 ($w \cdot x \cdot 0$) de los puntos con salida 0 ($w \cdot x \cdot 0$).

Algoritmo de entrenamiento del perceptrón

Iniciar: w_0 aleatoriamente; $t \leftarrow 0$

Repetir Mientras quede vectores mal clasificados

 Seleccionar aleatoriamente un vector x

 Si $x \in P$ (salida esperada 1) y $w_t \cdot x < 0$ (salida calculada 0) entonces

 [Hay un error]

$$w_{t+1} \leftarrow w_t + x; t \leftarrow t+1$$

Fin si

Si $x \in N$ (salida esperada 0) y $w_t \cdot x \geq 0$ (salida calculada 1) entonces

 [Hay un error]

$$w_{t+1} \leftarrow w_t - x; t \leftarrow t+1$$

fin si

fin repite

También se encuentra otra forma de escribir el algoritmo, en forma esquemática, para varias salidas.

Notamos los ejemplos de entrenamiento (X, d) , con X la entrada y d la salida esperada.

0.- Fijar los pesos para las entradas $W_{ij} \ i = 1..p, j = 1..q$

1.- Repetir pasos 2-4 Mientras existan ejemplos mal clasificados

2.- Seleccionar otro elemento de entrenamiento (X, d) mientras haya.

3.- Introducir X en red, calcular la salida Y : $Y_j = 1$ si $\sum_i X_i W_{ij} > U$ y 0 en otro caso

4.- [Comparar Y con d (salida esperada)]

 Para $j = 1..q$

 Si salida j es incorrecta, Entonces; $W'_{ij} = W_{ij} - (d_j - Y_j) X_i$; FinSi

FinPara [para cada salida (j) no correcta, Si $Y_j = 0$ [$x \in P$; $\sum_i X_i W_{ij} < U$], $W'_{ij} = W_{ij} + X_i$]

[Si $Y_j = 1$ [$x \notin N$; $\sum_i X_i W_{ij} > U$], $W'_{ij} = W_{ij} - X_i$]

5.- FIN

II. CONCLUSIONES

En el desarrollo del área de la inteligencia artificial la simulación del cerebro humano ha sido uno de los grandes objetos de estudios. La unidad básica (la neurona) es simulada matemáticamente por el perceptrón, este nos ayuda a desarrollar modelos que nos permiten realizar un proceso de selección o clasificación, entrenado por un conjunto de datos ya analizados y corroborados.

El algoritmo de entrenamiento permite que podamos manejar entradas de patrones más complejas y que siga funcionando sin ningún problema.

REFERENCIAS

- [1] http://www.lcc.uma.es/~munozp/documentos/modelos_computacionales/temas/Tema4MC-05.pdf
- [2] <http://bibing.us.es/proyectos/abreproy/11084/fichero/Memoria+por+cap%C3%ADtulos+%252FCap%C3%ADtulo+4.pdf>