

Entrenamiento de un Perceptrón

Training a Perceptron

Juan Felipe Bermúdez Andrade

Facultad de ingenierías, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: felipe98@utp.edu.co

Resumen— Una de las características más significativas de las redes neuronales es su capacidad para aprender a partir de alguna fuente de información interactuando con su entorno. En 1958 el psicólogo Frank Rosenblatt 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 este documento se podrá apreciar cómo se entrena este Perceptrón a través de el algoritmo de Rosenblatt, que en resumidas cuentas consiste en pasar un conjunto de datos de entrenamiento para que la regla de aprendizaje del Perceptrón vaya añadiendo un incremento a los pesos de las entradas hasta que las salidas obtenidas coincidan con las salidas esperadas.

Palabras clave — *Perceptrón, sinapsis, aprendizaje, neurona, Rosenblatt, clasificación.*

Abstract— One of the most significant characteristics of neural networks is their ability to learn from some source of information interacting with their environment. In 1958 the psychologist Frank Rosenblatt developed a simple neuron model based on the McCulloch and Pitts model and a learning rule based on error correction. This model was called Perceptron. In this document it will be possible to appreciate how this Perceptron is trained through the Rosenblatt algorithm, which in summary consists of passing a set of training data so that the Perceptron learning rule adds an increase to the weights of the inputs, until the outputs obtained coincide with the expected outputs.

Key Word — *Perceptron, synapse, learning, neuron, Rosenblatt, classification.*

I. INTRODUCCIÓN

En 1943 McCulloch y Pitts desarrollaron un modelo matemático que emula el comportamiento de una neurona. En el modelo biológico, las dentritas tienen la función de recibir impulsos. En función de esos impulsos, se transmite una señal electroquímica a otras neuronas a través del Axón y sus terminales. La neurona McCulloch-Pitts (MCP) emula este comportamiento. El modelo recibe unas entradas, que llevan unos pesos asociados que simulan la intensidad de los impulsos. El valor de dichas señales se suma y, si dicha suma supera un cierto valor umbral, se produce una salida. El umbral actúa sobre una función de activación, que decidirá entre dos valores el valor de la salida, dependiendo si la suma de las señales supera o no dicho valor umbral.

En 1957, Frank Rosenblatt puso nombre a lo que, en Machine Learning, podríamos llamar una neurona artificial. La llamó Perceptrón. Basándose en el modelo MCP, desarrolló un algoritmo matemático que, simulando el comportamiento de las neuronas, tiene la habilidad de aprender. Lo que consigue aprender dicho algoritmo son los pesos adecuados

para que la salida sea la correcta. Y como al final está decidiendo entre dos valores, podemos decir que el perceptrón está clasificando en dos clases los datos.

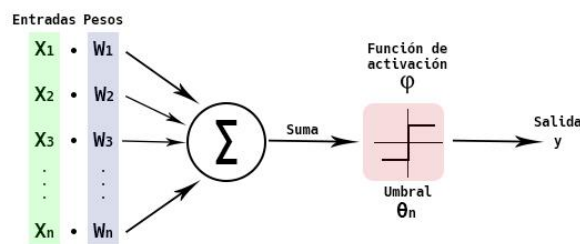


Figura 1. Neurona de McCulloch-Pitts

II. CONTENIDO

1. Algoritmo de Rosenblatt

El algoritmo de Rosenblatt consiste en que al perceptrón se le pasa un conjunto de datos de entrenamiento. La regla de aprendizaje del perceptrón consiste en ir añadiendo un incremento a los pesos de las entradas hasta que las salidas obtenidas por el perceptrón coincidan con las salidas esperadas del conjunto de datos de entrenamiento. Prueba y error.

Al comienzo del algoritmo se escoge un valor igual a cero para todos los pesos. A continuación, para todas las entradas del conjunto de datos, se calcula la salida de la neurona y se incrementan los pesos. El incremento de los pesos se calcularía así:

$$\Delta w_n = \eta(\delta_n - y_n)x_n$$

Donde:

- δ sería la salida esperada del conjunto de datos de entrenamiento.
- y sería la salida calculada por la neurona.
- η sería la tasa de aprendizaje, con un valor entre 0 y 1.

Si se quiere que el perceptrón logre calcular los pesos correctos y las salidas calculadas por la neurona coincidan con las salidas esperadas se debe tener en cuenta lo siguiente:

- El conjunto de datos tiene que ser linealmente separable.
- La tasa de aprendizaje tiene que ser suficientemente pequeña.

Si no se cumplen las dos condiciones anteriores, nuestro perceptrón seguirá intentando calcular los pesos correctos hasta el infinito, y más allá.

De forma más detallada, el aprendizaje de un perceptrón está definido por los siguientes pasos:

2. Algoritmo de aprendizaje del Perceptrón con entrenamiento individualizado

A continuación, de forma más detallada se mostrará como es el proceso de aprendizaje del Perceptrón.

Paso 0: Inicialización

Inicializar los pesos con números aleatorios en el intervalo $[-1,1]$.

Paso 1: k-ésima iteración

Calcular

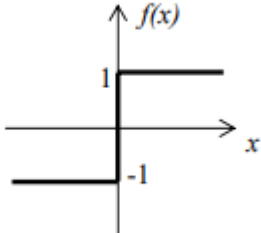
$$y(k) = \text{sgn}\left(\sum_{j=1}^{n+1} w_j x_j(k)\right)$$


Figura 2. Función Signo, para hacer la transferencia.

Paso 2: Corrección de los pesos

Si la salida dada por el perceptrón es diferente a la que se desea obtener, se debe modificar los pesos con la siguiente expresión.

$$w_j(k+1) = w_j(k) + \eta[z(k) - y(k)]x_j(k),$$

Paso 3: Parada

Si los pesos no se han modificado en las últimas iteraciones, es decir:

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

Si se da, se debe parar porque la red se ha estabilizado.

III. CONCLUSIONES

El Perceptrón es un modelo de red neuronal de aprendizaje por refuerzo, es decir, se fija en si la salida está bien o está mal, pero no en el valor de la discrepancia entre ambas, además, es síncrono y sin realimentación.

Las aplicaciones del Perceptrón se limitan a los problemas separables linealmente como puede ser la separación de elementos en varias clases, siempre y cuando sean separables linealmente una de otras. Para este caso se tendrá que tener tantos perceptrones en la capa de salida como clases de existan.

REFERENCIAS

1. <https://koldopina.com/comoentrenar-a-tu-perceptron/>
2. http://www.lcc.uma.es/~munozp/documentos/modelos_computacionales/temas/Tema4MC-05.pdf
3. <http://disi.unal.edu.co/profesores/lctorress/RedNeu/LiRna004.pdf>