

Entrenamiento de un perceptrón

Training a perceptron

Autor: Daniel Cifuentes Estupiñán

Ingenierías, Universidad tecnológica de Pereira, Pereira, Colombia

Correo-e: d.cifuentes@utp.edu.co

Resumen— El perceptrón surgió de la idea de lograr simular el comportamiento de una neurona humana usando modelos matemáticos, gracias a modelos computacionales este pudo ser simulado. A partir de este se crearon lo que se conoce en el mundo de la inteligencia artificial como redes neuronales, que nos permiten solucionar problemas mediante el aprendizaje de los perceptrones que componen estas redes. A partir de algoritmos se pueden entrenar perceptrones siguiendo los pasos que se explicaran en el contenido de este artículo

Palabras clave— Bias, entrenamiento, Perceptrón

Abstract— The perceptron arose from the idea of simulating the behavior of a human neuron using mathematical models, thanks to computational models it could be simulated. From this we created what is known in the world of artificial intelligence as neural networks, which allow us to solve problems by learning the perceptron's that make up these networks. Based on algorithms, perceptron's can be trained following the steps that will be explained in the content of this article

Key Word —Bias, training, Perceptron

I. INTRODUCCIÓN

El perceptron es un algoritmo que desarrollo en base al modelo matemático que simulaba una neurona biológica, donde al perceptron se le pasan unos datos de entrada y este después de pasar por una función generara un dato salida. Gracias a esto se pueden crear complejas redes neuronales computacionales que nos permitan resolver problemas de un alto grado de complejidad.

En este artículo se mostrará cual es el proceso para entrenar un perceptron y así poder escalar a procesos mas complejos

II. CONTENIDO

1. **Que es un Perceptron:** El perceptron es un modelo computacional que se desarrolló en base al modelo MCP, que gracias a este permitió simular una neurona y que tenga la habilidad de aprender. Este algoritmo consigue aprender los pesos adecuados para que, junto con los datos de entradas, pueda generar la salida correcta[1].

2. **Características del perceptron:** El perceptron está conformado por [2]:

- a. **Entradas:** Son los datos o información que recibe el perceptron
- b. **Pesos:** Son valores numéricos que influncian a las entradas para obtener una salida correcta.
- c. **Función de activación:** Es una función matemática que se encarga de procesar y validar las entradas que fueron afectadas por los pesos, para así determinar una salida
- d. **Bias:** Parámetro que permite encontrar fácilmente la separación de algunas salidas de las redes neuronales, necesario para el entrenamiento del perceptrón

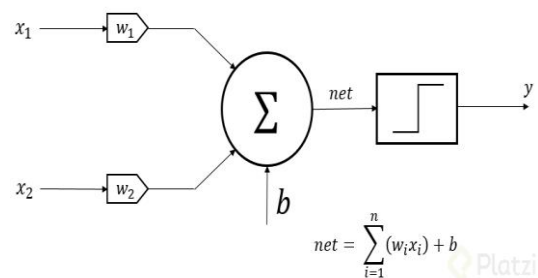


Figura 1: Modelo matemático de un perceptron

3. **Entrenamiento de un perceptrón[2]**

- a. **Paso 1:** En este paso se inicializan los valores de los pesos de las entradas y de los bias, estos valores pueden ser aleatorios, pero se recomienda que sean valores pequeños
- b. **Paso 2:** Se calcularán las salidas “net” a partir de los posibles valores que puedan tener las entradas, este proceso se realiza según las posibilidades de las entradas.

$$net_n = w_1 x_1 + w_2 x_2 + b$$

Figura 2: Función de cálculo de la salida

- c. **Paso 3:** En este paso se obtendrán las salidas “y” al pasar por la función de activación,

además se calcularán los errores de cada valor de la siguiente manera:

$$e = y' - y$$

Donde “e” es el valor del error, “y” es la salida deseada y “y’” es la salida obtenida.

Teniendo en cuenta que:

$y = 1$	$net > 0$
$y = 0$	$net \leq 0$

Figura 3: Posibles salidas de la función activación

Se pueden dar los siguientes casos:

Caso 1: Si “en=0”, se repite el paso 2 con las entradas siguientes.

Caso 2: Si “en \neq 0”, se continua con el paso 4.

Caso 3: Si todos los valores calculados de “e” son iguales a 0, el entrenamiento finaliza.

- d. **Paso 4:** En este paso se corrigen los pesos y bias a partir de los valores obtenidos en el paso 3, utilizando las siguientes formulas:

Corrección de pesos: $w_i(k+1) = w_i(k) + e \times x_i$.

Corrección de Bias: $b(k+1) = b(k) + e$.

Donde:

$w_i(k+1)$: Nuevo peso.

$w(k)$: Peso actual.

$b(k+1)$: Nuevo Bias.

$b(k)$: Bias actual.

e: Error actual.

x_i : Entrada.

Se deben corregir todos los pesos y bias hasta que el error sea igual a cero, cuando esto ocurra, el perceptrón ya estará entrenado.

III. CONCLUSIONES

Gracias a los perceptrones y redes neuronales artificiales, se pueden aplicar a un sinnúmero de campos para lograr resolver problemas iterativos de mucha complejidad. Los métodos de entrenamiento de los perceptrones se basan en técnicas ya trabajadas como el ensayo y error, algo muy parecido al proceso de aprendizaje humano, que si se repite constantemente hasta perfeccionar la técnica, se consigue un resultado exitoso.

REFERENCIAS

[1] <https://koldopina.com/como-entrenar-a-tu-perceptron/>

[2] <https://platzi.com/tutoriales/1157-ia-2017/2619-entrenamiento-del-perceptron/>

[3] <https://es.slideshare.net/rgfigueroa/perceptron-simple-y-regla-aprendizaje>

[4] <https://es.slideshare.net/aerdna07/regla-de-aprendizaje-del-perceptrn-simple>