# Entrenamiento de un perceptrón

## Simple perceptron training.

Autor: Luis Fernando Martínez Muñoz
*Ingeniería de sistemas y computación, Universidad Tecnológica de Pereira, Pereira, Colombia*
Correo-e: f.martinez@utp.edu.co

***Resumen*— Existe un algoritmo para el entrenamiento de un perceptrón para el cual se necesita que las salidas del perceptrón sean separables (reconocibles). Al llevar la ecuación del perceptrón para la cual el resultado es igual al umbral a forma matricial se puede observar que la ecuación es una recta que separa las salidas del perceptrón que superan el umbral de las que no lo superan. Si se desconoce la distribución de las salidas, se puede empezar asignando un valor arbitrario al vector peso y realizar ajustes a los valores dependiendo de los resultados obtenidos.**

***Palabras clave*—Perceptrón, neurona, entrenamiento, inteligencia artificial.**

***Abstract*— it exists an algorithm for simple perceptron training where is necessary that the perceptron's outputs are separable. It can be evident that the threshold equation in its matrix form is a straight line that separates the output values that surpass the threshold value from the ones that do not reach it. If the output distribution is unknown, the algorithm starts assigning an arbitrary value to the weights vector and adjusts future values depending on past results.**

***Key Word* —Perceptron, neuron, training, artificial intelligence.**

## I. INTRODUCCIÓN

El perceptrón es la parte esencial de la inteligencia artificial. Permite el procesamiento de información que requiere cálculos que son complejos y tediosos para el ser humano por sí solo simulando el comportamiento de una neurona humana, pero para lograr este cometido es necesario entrenar el perceptrón de tal forma que sus resultados en el procesamiento sean fiables y fieles a la realidad del problema.

El entrenamiento del perceptrón no es incierto, puesto que se apoya sobre el teorema de la convergencia el cual dice que, si las clases son linealmente separables, el algoritmo del perceptrón converge a una solución correcta en un número finito de pasos para cualquier elección inicial de pesos.[1]

## II. CONTENIDO

El modelo matemático de un perceptrón, el cual se puede expresar como una ecuación, la cual puede tomar 3 diferentes valores.

$$X_1W_1 + X_2W_2 = 0$$

Representando las entradas que pertenecen al umbral del perceptrón y que no se sabe si sus salidas valen 1 o 0.

$$X_1W_1 + X_2W_2 > 0$$

Representando las entradas que a su salida de la neurona dan un valor de 1, ya que superan el umbral de la neurona.

$$X_1W_1 + X_2W_2 < 0$$

Representando las entradas que a su salida de la neurona no superan el umbral, y por lo tanto toman un valor de 0.

La recta anterior se puede representar en su forma matricial como $[X1 \quad X2]\ \begin{bmatrix} W1 \\ W2 \end{bmatrix}$. Siendo así un producto punto entre dos vectores, el vector $\underset{X}{\rightarrow}$ y el vector $\underset{W}{\rightarrow}$ .

Aritméticamente la relación entre los dos vectores se puede representar como un producto punto

$$\left|\underset{X}{\rightarrow}\right| \cdot \left|\underset{W}{\rightarrow}\right| * \cos\theta$$

Si tenemos en cuenta que cos90°=0, podríamos observar gráficamente que el vector $\underset{X}{\rightarrow}$ y el vector $\underset{W}{\rightarrow}$ son perpendiculares, por lo tanto, los valores de X ingresados al perceptrón se encuentran sobre el umbral, y no se puede saber si son 1 o 0.

Teniendo en cuenta está modelación matemática del perceptrón se llegó ala formulación de un algoritmo de aprendizaje.

El algoritmo de aprendizaje del perceptrón, desarrollado por Rosenblatt en 1962, pertenece al grupo de algoritmos fundamentados en la corrección de errores.

- Se convierten los datos de entrada del perceptrón en un vector de características X.
- Si se tiene un umbral positivo se tiene un sesgo negativo, por lo que podemos evitar tratar el umbral de forma separa añadiendo una entrada $X_0 = 1$, así el umbral se trata como un peso más.
- Se asignan valores arbitrarios de peso a cada entrada.
- Se seleccionan ejemplos del conjunto de entrenamiento utilizando cualquier política que

garantice que todos los ejemplos de entrenamiento se acabarán escogiendo.

- Si la salida es correcta, se dejan los pesos tal cual.
- Si la unidad de salida incorrectamente da un cero, se añade el vector de entrada X al vector de pesos W.

$$\underset{w}{\rightarrow} = \underset{w}{\rightarrow} + \underset{x}{\rightarrow}$$

- Si la unidad de salida incorrectamente da un uno, se resta el vector de entrada del vector de pesos. [2]

$$\underset{w}{\rightarrow} = \underset{w}{\rightarrow} + (\underset{-x}{\rightarrow})$$

Para obtener resultados positivos se requiere que las clases sean linealmente separables, ya que la ecuación de un solo perceptrón es una sola línea recta, así que si la distribución de las clases no es linealmente separable no es posible que sean procesadas todas las entradas satisfactoriamente.

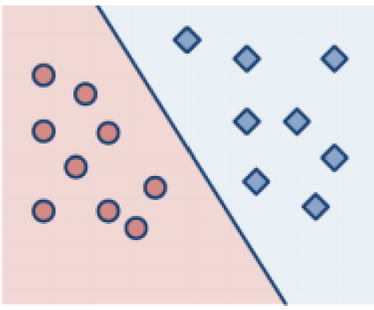

Figura 1. Clases linealmente separables.

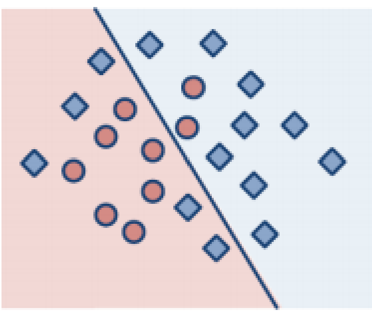

Figura 2. Clases linealmente no separables.

## III. CONCLUSIONES

El algoritmo de aprendizaje funciona siempre y cuando las clases sean linealmente separables.

Si existen más de dos clases se requiere más de una neurona para procesar las entradas.

Cuando hay múltiples clases el algoritmo tiene que ser modificado puesto que la complejidad del problema aumenta.

## REFERENCIAS


[1]. A. B. Novikoff. “ON CONVERGENCE PROOFS FOR PERCEPTRONS,”. 1962.

[2]. F. Berzal. “Redes Neuronales y Deep Learning I,”, 1era ed., vol. 1. 2018.