

# Perceptrón

## Perceptron

Autor: **Carlos Daniel Morales Isaza y Juan Camilo Olarte Betancourt**

*IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: [juan.olarte@utp.edu.co](mailto:juan.olarte@utp.edu.co) y [carlos.morales@utp.edu.co](mailto:carlos.morales@utp.edu.co)

**Resumen**— En este documento se tratarán diferentes temas de la inteligencia artificial, los cuales son el perceptrón, su entrenamiento y el backpropagation. En el documento se va dar una breve explicación de estos, ya que se abordaron de la manera más simple posible para su fácil entendimiento. Las temáticas siguen un hilo conductor, permitiendo al lector entender desde la idea de como nace el concepto de perceptrón hasta como es utilizado en una red neuronal para aprender modificándose automáticamente a sí mismo, todo con el fin de realizar tareas complejas. Toda la información que se encontrará en este documento es tomada de diferentes fuentes las cuales están reseñadas al final del mismo.

**Palabras clave**— inteligencia artificial, perceptrón, backpropagation

**Abstract**— In this document different topics of artificial intelligence will be treated, which are the perceptron, its training and backpropagation. A brief explanation of these will be given in the document, they were addressed in the simplest way possible for an easy understanding. The themes follow a common thread, allowing the reader to understand from the idea of how the concept of perceptron was born to how it is used in a neural network to learn by automatically modifying itself, all in order to perform complex tasks. All the information that will be found in this document is taken from different sources which are listed at the end of it.

**Keywords**— artificial intelligence, perceptron, backpropagation

## I. INTRODUCCIÓN

El perceptrón o neurona artificial es la forma más simple de una red neuronal, estas se encargan de generar un criterio de selección y este se aplica a un grupo predeterminado. El perceptrón también se puede unir con otros tipos de perceptrones y así crear una red neuronal mucho más grande

El modelo biológico más simple de un perceptrón es una neurona y viceversa. Es decir, el modelo matemático más simple de una neurona es un perceptrón. La neurona es una célula especializada y caracterizada por poseer una cantidad indefinida de canales de entrada llamados dendritas y un canal de salida llamado axón. Las dendritas operan como sensores que recogen información de la región donde se hallan y la derivan hacia el cuerpo de la neurona que reacciona mediante

una sinapsis que envía una respuesta hacia el cerebro, esto en el caso de los seres vivos.

El algoritmo de backpropagation se introdujo originalmente en la década de 1970, pero su importancia no se apreció completamente hasta un famoso artículo de 1986 de David Rumelhart, Geoffrey Hinton y Ronald Williams. Este documento describe varias redes neuronales en las que la retropropagación funciona mucho más rápido que los enfoques de anteriores aprendizajes, lo que hace posible utilizar redes neuronales para resolver problemas que anteriormente habían sido insolubles.

### I.1 Perceptrón

Una definición de tantas que podemos encontrar sobre el perceptrón es:

“El perceptrón simple es una red neuronal artificial de una sola capa, inspirada en el comportamiento y funcionamiento del cerebro humano”.

Para empezar, el perceptrón simple es la red neuronal artificial más antigua. Esta máquina tuvo sus limitaciones pues realmente no se podía entrenar para que reconociera demasiados patrones debido a que era una red neuronal artificial de una sola capa y solamente permitía una clasificación lineal. En el campo de la inteligencia artificial, el perceptrón simple es una red neuronal artificial de una sola capa, inspirada en el comportamiento y funcionamiento del cerebro humano.

Una neurona cerebral posee el aspecto y las partes que se muestran en la figura 1.

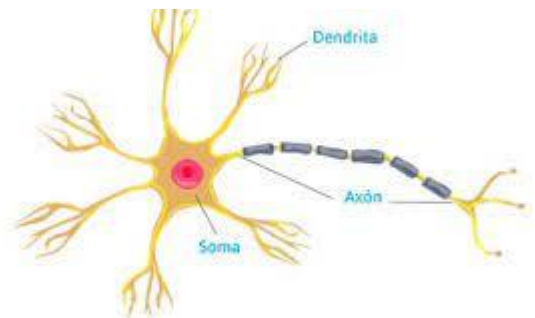


Figura 1. Neurona y sus partes.

El cerebro tiene un funcionamiento neuronal con un número inmenso de neuronas con interconexiones masivas entre ellas. La mayoría de las neuronas codifican sus salidas como una serie de breves pulsos periódicos, llamados *potenciales de acción*, que se originan cercanos al soma de la célula y se propagan a través del axón. Luego, este pulso llega a las sinapsis y de ahí a las dendritas de la neurona siguiente.

Los dos comportamientos importantes son:

- El impulso que llega a una sinapsis y el que sale de ella no son iguales, depende de la cantidad de neurotransmisor(cambia durante el proceso de aprendizaje y es aquí donde se almacena la información)
- En el soma se suman las entradas de todas las dendritas y si estas sobrepasan un cierto umbral, entonces se transmitirá un pulso a lo largo del axón, contrario no se transmitirá.

Lo que se hace con el perceptrón es modelar un esquema basado en los dos comportamientos anteriores. La figura 2. muestra el modelo del perceptrón, en este se observa cómo de forma similar a lo que pasa con una neurona el perceptrón recibe señales enviadas por otros perceptrones ( $x_1$  a  $x_5$ ), esas señales de entrada son valores numéricos que representan lo que será procesado. Los valores  $w_1$  a  $w_5$  representan los pesos sinápticos en las dendritas de una neurona, lo que hace cada peso sináptico es simplemente multiplicar a su entrada correspondiente y define la importancia relativa de cada entrada.

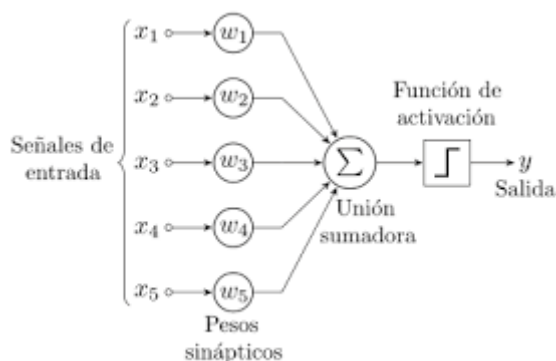


Figura 2. El perceptrón

Los perceptrones convierten sus múltiples entradas en una entrada global. Para combinar este montón de entradas se utiliza una función de entrada.

Se emplean variedades de funciones de entrada según sean apropiadas para el caso de estudio, algunas de las funciones de entradas más comúnmente utilizadas y conocidas son:

1) Sumatorio de las entradas pesadas: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos.

2) Multiplicatorio de las entradas pesadas: es el producto de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos.

3) Máximo de las entradas pesadas: solamente toma en consideración el valor de entrada más fuerte, previamente multiplicado por su peso correspondiente

El funcionamiento de un perceptrón inspirado en una neurona consiste en que si la suma de las entradas excitatorias supera el umbral de activación sin la presencia de una entrada inhibitoria(negativa), la neurona se activa y emite una respuesta (equivalente al valor de 1). Pero si hay presencia de una entrada inhibitoria, no se activará y no emitirá respuesta(representado por el valor de 0). Lo que se hace es aplicar una función de activación, que puede ser, por ejemplo, una función tipo escalón o sigmoideas, como la tangente hiperbólica.

La función de activación calcula el estado de activación de un perceptrón; transformando la entrada global (menos el umbral) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (-1 a 1). Esto es así, porque un perceptrón puede estar totalmente inactivo(0 o -1) o activo (1).

Algunas funciones de activación típicas, no lineales, siendo  $y^{(in)}$  la entrada global y  $Y$  la función de activación, se presentan en las figuras 3 y 4.

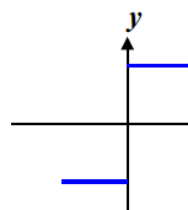


Figura 3. Escalón

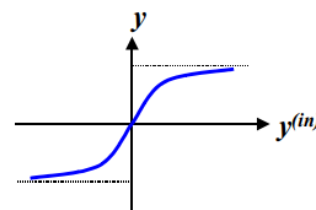


Figura 4. Sigmoidea

El último componente del perceptrón es la función de salida, la función de salida determina que valor se transfiere a los perceptrones vinculados. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa al perceptrón subsiguiente. Los valores por salida normalmente están comprendidos en el rango  $[0,1]$  o  $[-1,1]$ . También pueden ser binarios  $\{0,1\}$  o  $\{-1,1\}$ .

## I.2 Redes neuronales

Las Redes Neuronales Artificiales son una parte de la Inteligencia Artificial (IA) que sigue la corriente conexionista. Ésta hace referencia a una forma de ver la mente computacional que defiende que la información se encuentra entre los elementos que forman el sistema.

El inicio de las redes neuronales se sitúa en 1943 con la aportación de McCulloch y Pitts, creadores de la primera neurona artificial.

La segunda aportación más importante fue la de A. Hebb y su regla de aprendizaje hebbiano en 1949, la cual explica el mecanismo de regulación entre las conexiones. Este

aprendizaje propone que los pesos se incrementan si las neuronas de origen y de destino están activadas simultáneamente (Díaz, Gómez, Graciani y López, 1994)

El Perceptrón Simple (Rosenblatt, 1959), consta de una red con una capa de salida de  $n$  neuronas y otra de salida con  $m$  neuronas. Utiliza señales binarias, tanto de entrada como de salida de las neuronas y su función de activación es de tipo signo

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta (Funahashi, 1989; Hornik, Stinchcombe y White, 1989). En la figura 5 podemos observar un perceptrón multicapa formado por una capa de entrada, una capa oculta y una de salida

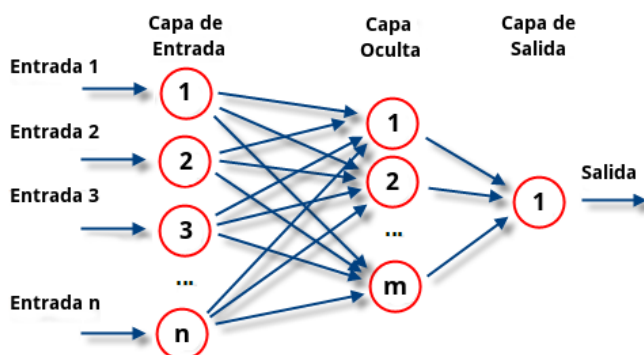


Figura 5 - Perceptrón multicapa

En este tipo de arquitectura, las conexiones entre neuronas son siempre hacia delante, es decir, las conexiones van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa, no hay conexiones laterales (conexiones entre neuronas pertenecientes a una misma capa), ni conexiones hacia atrás (conexiones que van desde una capa hacia la capa anterior). Por tanto la información siempre se transmite desde la capa de entrada hacia la capa de salida.

Una red neuronal debe aprender a calcular la salida correcta para cada constelación (arreglo o vector) de entrada en el conjunto de ejemplos. Este proceso de aprendizaje se denomina: proceso de entrenamiento o acondicionamiento. El conjunto de datos (o conjunto de ejemplos) sobre el cual este proceso se basa es, por ende, llamado: conjunto de datos de entrenamiento. El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el mismo se reducen a la destrucción, modificación y creación de conexiones entre las neuronas.

### I.3 Backpropagation

La propagación hacia atrás de errores o retropropagación (del inglés backpropagation) es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales. El método emplea un ciclo propagación – adaptación de dos fases.

Es un algoritmo ampliamente utilizado para entrenar redes neuronales de Backpropagation. Existen generalizaciones de Backpropagation para otras redes neuronales artificiales y para funciones en general. Todas estas clases de algoritmos se denominan genéricamente "Backpropagation".

El algoritmo tiene dos partes, en la primera, cada grupo o vector de entradas es presentado a la red, y está, usualmente inicializada con pesos aleatorios, genera algunas salidas. Estas salidas de la red son comparadas con las salidas que se espera que la red asocie a esas entradas, y se calcula la diferencia o error. Enseguida se presenta el segundo vector de entrada, también conocido como ejemplo; nuevamente se propaga la red, y se evalúa el error entre la salida de la red y las salidas que se esperarían. El error debido a todos los ejemplos es acumulado, y se repite esto con todos los ejemplos que incluya el conjunto de datos que se tenga. En la segunda parte, con base en el error acumulado, se empiezan a cambiar los pesos de cada conexión en la capa de salida, de manera que el error disminuya.

Se utiliza el método del descenso del gradiente para lograr el mínimo valor posible de error cuadrático medio. Con esto se modifican los pesos sinápticos hasta que el sistema evidencie una conducta deseable. Este método de aprendizaje es de tipo no supervisado, para un perceptrón multicapa con  $n$  entradas  $[x_j]$  y  $m$  salidas  $[y_i]$  el programador debe definir un conjunto de  $p$  patrones, donde cada patrón  $[p_\mu]$  especifica las  $m$  salidas deseadas  $[t_i]$  para alguna posible combinación numérica de entrada. Con la diferencia entre las salidas deseadas y las obtenidas puede calcularse el error cuadrático medio  $[E_\mu]$  del sistema y, para minimizarlo, en cada nueva iteración del sistema se ajusta cada una de las conexiones sinápticas  $[w_{ij}]$  según la variación sináptica  $[\Delta w_{ij}]$  calculada mediante la siguiente expresión:

$$\Delta w_{ij} = -\varepsilon \sum_{\mu=1}^p \frac{\partial E_\mu}{\partial w_{ij}} \quad ; \text{ donde } E_\mu = \frac{1}{2} \sum_{i=1}^m (t_i^\mu - y_i^\mu)^2$$

Figura 6. Ecuación para el algoritmo de aprendizaje backpropagation

El valor  $[\varepsilon]$  se utiliza como tasa de convergencia o velocidad de aprendizaje. Las funciones de activación utilizadas para calcular las  $m$   $[y_i]$  deben ser totalmente derivables.

## REFERENCIAS

### Referencias en la Web:

[1]

[http://bibing.us.es/proyectos/abreproy/11084/fichero/Memoria+por+cap%C3%ADtulos+%252FCap%C3%ADtulo+4.pdf+#:~:text=El%20perceptr%C3%B3n%20es%20la%20forma,ambos%20lados%20de%20un%20hiperplano\).&text=La%20prueba%20de%20convergencia%20del,teorema%20de%20convergencia%20del%20perceptr%C3%B3n.](http://bibing.us.es/proyectos/abreproy/11084/fichero/Memoria+por+cap%C3%ADtulos+%252FCap%C3%ADtulo+4.pdf+#:~:text=El%20perceptr%C3%B3n%20es%20la%20forma,ambos%20lados%20de%20un%20hiperplano).&text=La%20prueba%20de%20convergencia%20del,teorema%20de%20convergencia%20del%20perceptr%C3%B3n.)

[2]

<https://dspace.uib.es/xmlui/bitstream/handle/11201/1126/TFG%20Marta%20Vidal%20Gonz%C3%A1lez.pdf?sequence=1>

[3]

<https://empresas.blogthinkbig.com/como-funciona-el-algoritmo-backpropagation-en-una-red-neuronal/>

[4]

<http://avellano.fis.usal.es/~lalonso/RNA/introMLP.htm>

[5]

<https://www.crehana.com/co/blog/desarrollo-web/que-es-perceptron-algoritmo/>

[6]

[https://www.researchgate.net/publication/331640802\\_Tutorial\\_sobre\\_Redes\\_Neuronales\\_Artificiales\\_El\\_Perceptron\\_Multicapa](https://www.researchgate.net/publication/331640802_Tutorial_sobre_Redes_Neuronales_Artificiales_El_Perceptron_Multicapa)

[7]

[https://es.wikipedia.org/wiki/Propagaci%C3%B3n\\_hacia\\_atrás](https://es.wikipedia.org/wiki/Propagaci%C3%B3n_hacia_atrás)

[8]

<https://revistas.javeriana.edu.co/index.php/iyu/article/view/1342/17535>

[9]

[https://www.academia.edu/27805755/REDES\\_NEURONALES\\_ARTIFICIALES\\_pdf?from=cover\\_page](https://www.academia.edu/27805755/REDES_NEURONALES_ARTIFICIALES_pdf?from=cover_page)

[10]

[https://d1wqtxts1xzle7.cloudfront.net/36957218/redesneuronales-with-cover-page-v2.pdf?Expires=1633888705&Signature=gjY~eF7AMEOm~RROOaQ7jGDntfCXfZqvFq-AdOYsagK~87CHdolG0xVQmWDBjSo40AyAEKeK3lbdtpTBeFbaOc0XdcVjKZ77x5bcDEHce6AKOcJRNjJA~gEprfWZFU~IT7UNV7mneUhBf713DISMnDOzKynWkdy4iHfOuAOtEVKHNRNUrkCvoOC0aXONi9ESididCWOkOrBiWf-wTotcDryx-hjNV7V1i3tBVzS1e16tQSa-LO-kWXUjn20YNEDVonaNfNLuBbAZSvEgX1x6M8bisMJDctSvOYJDctb0u9L6mXXHtZzg7o5fxySXDOQWcTEBnkoobNPU9TxyqQGA\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/36957218/redesneuronales-with-cover-page-v2.pdf?Expires=1633888705&Signature=gjY~eF7AMEOm~RROOaQ7jGDntfCXfZqvFq-AdOYsagK~87CHdolG0xVQmWDBjSo40AyAEKeK3lbdtpTBeFbaOc0XdcVjKZ77x5bcDEHce6AKOcJRNjJA~gEprfWZFU~IT7UNV7mneUhBf713DISMnDOzKynWkdy4iHfOuAOtEVKHNRNUrkCvoOC0aXONi9ESididCWOkOrBiWf-wTotcDryx-hjNV7V1i3tBVzS1e16tQSa-LO-kWXUjn20YNEDVonaNfNLuBbAZSvEgX1x6M8bisMJDctSvOYJDctb0u9L6mXXHtZzg7o5fxySXDOQWcTEBnkoobNPU9TxyqQGA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)

[11]

<https://disi.unal.edu.co/~lctorress/RedNeu/LiRna003.pdf>