

# Redes Neuronales Artificiales

*Claudio Javier Tablada – Germán Ariel Torres*

**Resumen.** Una Red Neuronal Artificial es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro, y que es utilizada para resolver un amplio rango de problemas. Debido a su flexividad, una única red neuronal es capaz de realizar diversas tareas. En este artículo aplicaremos las mismas para resolver tareas de clasificación en el plano.

## 1. Introducción.

Una RNA (Red Neuronal Artificial) es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro. Esta también puede ser vista como un sistema inteligente que lleva a cabo tareas de manera distinta a como lo hacen las computadoras actuales. Si bien estas últimas son muy rápidas en el procesamiento de la información, existen tareas muy complejas, como el reconocimiento y clasificación de patrones, que demandan demasiado tiempo y esfuerzo aún en las computadoras más potentes de la actualidad, pero que el cerebro humano es más apto para resolverlas, muchas veces sin aparente esfuerzo (considere el lector como ejemplo el reconocimiento de un rostro familiar entre una multitud de otros rostros). El cerebro puede considerarse un sistema altamente complejo. Su unidad básica, la neurona, está masivamente distribuida con conexiones entre ellas (se calcula que hay aproximadamente 10 billones de neuronas en la corteza cerebral y 60 trillones de conexiones neuronales).

Si bien hay distintos tipos de neuronas biológicas, en la figura 1 se muestra un esquema simplificado de un tipo particular que es muy común. Vemos que la misma está compuesta por:

- El cuerpo central, llamado *soma*, que contiene el núcleo celular
- Una prolongación del soma, el *axón*
- Una ramificación terminal, las *dendritas*
- Una zona de conexión entre una neurona y otra, conocida como *sinapsis*

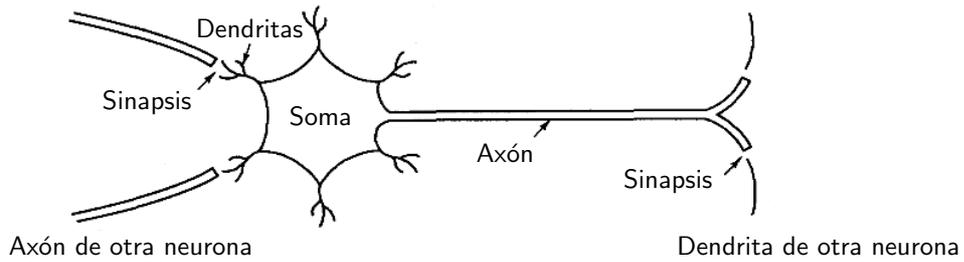


Figura 1: Modelo simplificado de una neurona biológica

La función principal de las neuronas es la transmisión de los impulsos nerviosos. Estos viajan por toda la neurona comenzando por las dendritas hasta llegar a las terminaciones del axón, donde pasan a otra neurona por medio de la conexión sináptica.

La manera en que respondemos ante los estímulos del mundo exterior y nuestro aprendizaje del mismo está directamente relacionado con las conexiones neuronales del cerebro, y las RNAs son un intento de emular este hecho.

## 2. Modelo neuronal de McCulloch-Pitts

El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples, fué presentado en el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts. Un ejemplo de modelo neuronal con dos entradas  $x$  e  $y$  es representado en la figura 2 (ver *Haykin* [1], pag. 33 y *Kröse and van der Smagt* [3], pag. 16).

El mismo consta de:

- Las entradas  $x$  e  $y$
- Los pesos sinápticos  $w_1$  y  $w_2$  correspondientes a cada entrada
- Un término aditivo  $b$
- Una función de activación  $f$
- Una salida  $z$

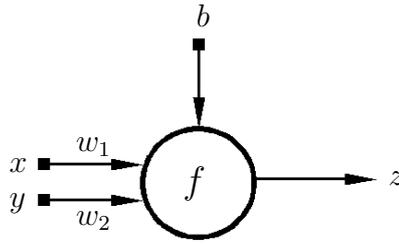


Figura 2: Modelo de McCulloch-Pitts para una neurona artificial

Las entradas  $x$  e  $y$  son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida  $z$  es la respuesta a tal estímulo. La neurona se adapta al medio circundante y aprende de él modificando el valor de sus pesos sinápticos  $w_1$  y  $w_2$  y su término aditivo  $b$ . Estos son conocidos como los parámetros libres del modelo, pues los mismos pueden ser modificados y adaptados para realizar una tarea determinada.

En este modelo, la salida neuronal  $z$  está dada por

$$z = f(w_1 x + w_2 y + b)$$

La función de activación  $f$  es seleccionada de acuerdo a la tarea realizada por la neurona.

### 3. Usando el Perceptron para clasificar clases en el plano

Aplicaremos el modelo neuronal de la sección anterior para realizar tareas de clasificación en el plano. Para tal fin, consideraremos como función de activación a la *función signo* definida por:

$$f(s) = \begin{cases} 1, & \text{si } s \geq 0 \\ -1, & \text{si } s < 0 \end{cases}$$

Por lo tanto, la salida neuronal  $z$  estará dada en este caso por:

$$z = \begin{cases} 1, & \text{si } w_1 x + w_2 y + b \geq 0 \\ -1, & \text{si } w_1 x + w_2 y + b < 0 \end{cases} \quad (1)$$

Supongamos que tenemos dos clases en el plano: la clase  $\mathcal{C}_1$ , formada por los círculos blancos, y la clase  $\mathcal{C}_2$ , formada por los círculos negros (ver figura 3), donde cada elemento de estas clases está representado por un punto  $(x, y)$  en el plano. Supondremos además que tales clases son separables linealmente, es decir, es posible trazar una recta que separe estrictamente ambas clases (figura 3 (a)).

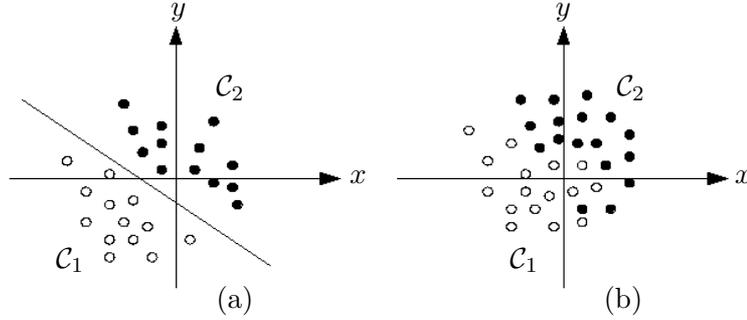


Figura 3: (a) Clases en el plano separables linealmente. (b) Clases en el plano no separables linealmente.

Consideremos entonces el modelo neuronal representado en la figura 2, cuya salida está dada por (1). Diremos que la neurona artificial clasifica correctamente las clases  $\mathcal{C}_1$  y  $\mathcal{C}_2$  si dados los pesos sinápticos  $w_1$  y  $w_2$  y el término aditivo  $b$ , la recta con ecuación

$$y = -\frac{w_1}{w_2}x - \frac{b}{w_2}$$

es una recta separante de las dos clases. La ecuación implícita de la recta es

$$w_1 x + w_2 y + b = 0$$

Obsérvese que si el punto  $(x_0, y_0) \in \mathcal{C}_1$ , entonces  $w_1 x_0 + w_2 y_0 + b < 0$  y si  $(x_0, y_0) \in \mathcal{C}_2$ , entonces  $w_1 x_0 + w_2 y_0 + b > 0$ . Por lo tanto, dado el par  $(x_0, y_0) \in \mathcal{C}_1 \cup \mathcal{C}_2$ , la neurona clasifica al mismo de la siguiente manera:

$$(x_0, y_0) \in \mathcal{C}_1 \iff z = -1$$

$$(x_0, y_0) \in \mathcal{C}_2 \iff z = 1$$

Si ahora tomamos dos clases  $\mathcal{C}_1^*$  y  $\mathcal{C}_2^*$  (separables linealmente) distintas a las anteriores, entonces la neurona puede no clasificar correctamente a estas clases, pues la recta  $y = -\frac{w_1}{w_2}x - \frac{b}{w_2}$  puede no ser una recta separante de las mismas. Sin embargo, es posible modificar los parámetros libres y obtener nuevos parámetros  $w_1^*$ ,  $w_2^*$  y  $b^*$  tal que la recta  $y = -\frac{w_1^*}{w_2^*}x - \frac{b^*}{w_2^*}$  sea la recta separante. El proceso por el cual la neurona pasa de los parámetros  $w_1$ ,  $w_2$  y  $b$  a los parámetros  $w_1^*$ ,  $w_2^*$  y  $b^*$  se conoce como *método de aprendizaje*. Este proceso es el que permite modificar los parámetros libres con el fin de que la neurona se adapte y sea capaz de realizar diversas tareas.

El método de aprendizaje que detallaremos a continuación y que utilizaremos para adaptar los parámetros libres con el fin de clasificar correctamente las clases  $\mathcal{C}_1$  y  $\mathcal{C}_2$  se conoce como *método de error-corrección* (ver *Kecman* [2], pag. 204). Para aplicarlo es necesario:

- Un conjunto de entrenamiento  $\mathcal{D}$ .
- Un instructor.
- Valores iniciales  $w_1^{inicial}$ ,  $w_2^{inicial}$  y  $b^{inicial}$  arbitrarios de los parámetros libres.

El conjunto de entrenamiento es definido por  $\mathcal{D} = \mathcal{C}_1 \cup \mathcal{C}_2$ . El entrenamiento consiste en lo siguiente: El instructor toma un elemento  $(x_0, y_0) \in \mathcal{D}$  al azar y presenta éste a la neurona. Si la neurona clasifica mal este punto, es decir, si la salida de la neurona es  $z = -1$  cuando  $(x_0, y_0) \in \mathcal{C}_2$  ó  $z = 1$  cuando  $(x_0, y_0) \in \mathcal{C}_1$ , entonces la siguiente corrección es aplicada a los parámetros libres iniciales

$$w_1 = w_1^{inicial} + d \cdot x_0$$

$$w_2 = w_2^{inicial} + d \cdot y_0$$

$$b = b^{inicial} + d$$

donde el valor de  $d$  se obtiene de la siguiente manera:

$$d = \begin{cases} 1, & \text{si } z = -1 \text{ y } (x_0, y_0) \in \mathcal{C}_2 \\ -1, & \text{si } z = 1 \text{ y } (x_0, y_0) \in \mathcal{C}_1 \end{cases}$$

Si la neurona clasifica bien el punto  $(x_0, y_0)$ , entonces ninguna corrección es realizada. El procedimiento es repetido ingresando a la neurona otro punto del conjunto  $\mathcal{D}$  y usando los últimos parámetros  $w_1$ ,  $w_2$  y  $b$  corregidos (no los

parámetros iniciales). Nuevamente, si la neurona clasifica mal el punto ingresado, entonces una corrección similar a la anterior es aplicada. Esto es repetido hasta que todos los puntos del conjunto  $\mathcal{D}$  son presentados a la neurona. Si en este transcurso hubo correcciones, entonces el procedimiento es repetido nuevamente con todos los puntos de  $\mathcal{D}$ . El entrenamiento termina cuando la neurona clasifica correctamente todos los elementos del conjunto de entrenamiento. Este procedimiento converge, es decir, en un número finito de pasos es posible obtener los parámetros finales  $w_1^{final}$ ,  $w_2^{final}$  y  $b^{final}$  tales que  $y = -\frac{w_1^{final}}{w_2^{final}}x - \frac{b^{final}}{w_2^{final}}$  es una recta separante de las clases  $\mathcal{C}_1$  y  $\mathcal{C}_2$ .

En la figura 4 se muestran dos clases  $\mathcal{C}_1$  y  $\mathcal{C}_2$  formada por 50 elementos cada una, las cuales fueron separadas linealmente usando el método de error-corrección (estas clases constan de elementos aleatorios, donde  $\mathcal{C}_1$  tiene media  $\mu_1 = (1, -1)$  y  $\mathcal{C}_2$  tiene media  $\mu_2 = (3, 2)$ ). Ambas clases tienen varianza  $\sigma^2 = 0.4$ ).

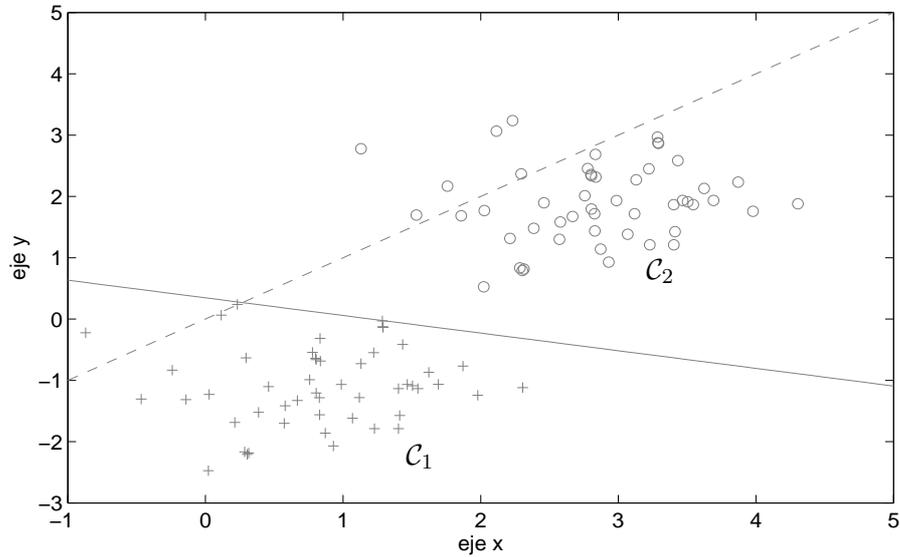


Figura 4: Clasificación de dos clases en el plano usando el método de error-corrección.

Los parámetros libres iniciales tomados fueron

$$w_1^{inicial} = -1 \quad , \quad w_2^{inicial} = 1 \quad , \quad b^{inicial} = 0$$

Obsérvese que la recta inicial  $y = x$  , la cual es mostrada en línea de trazos, no es una recta separante de las clases. Después de terminado el entrenamiento, se obtuvieron los parámetros finales

$$w_1^{final} = 0.8319943 \quad , \quad w_2^{final} = 2.892537 \quad , \quad b^{final} = -1$$

La recta separante obtenida

$$y = -\frac{0.8319943}{2.892537}x + \frac{1}{2.892537}$$

se muestra en línea llena.

Un modelo neuronal utilizado para clasificación, cuya salida  $z$  está dada por (1) y que utiliza el método de error-corrección para modificar sus parámetros libres se conoce como *Perceptron* (el nombre deriva de la palabra en inglés “perception”). Estas neuronas pueden agruparse formando una RNA conocida como *Perceptron múltiple*.

#### 4. El Perceptron multicapa

Un Perceptron múltiple puede tener sus neuronas organizadas por capas (figura 5). Así, tenemos los *nodos de entrada* formados por las entradas a la red, la *capa de salida* formada por las neuronas que constituyen la salida final de la red, y las *capas ocultas* formadas por las neuronas que se encuentran entre los nodos de entrada y la capa de salida (ver *Haykin* [6], pag. 44). Una RNA puede tener varias capas ocultas o no tener ninguna de ellas. Los links sinápticos (las flechas llegando o saliendo de una neurona) indican el flujo de la señal a través de la red y tienen asociado un peso sináptico correspondiente. Si la salida de una neurona va dirigida hacia dos o más neuronas de la siguiente capa, cada una de estas últimas recibe la salida neta de la neurona anterior. La cantidad de capas de una RNA es la suma de las capas ocultas más la capa de salida. En el caso de existir capas ocultas nos referimos a la RNA como un *Perceptron multicapa*.

Entre las diversas tareas en las que una RNA puede aplicarse podemos mencionar:

- Clasificación lineal y no lineal de una cantidad arbitraria  $\mathcal{C}_1, \dots, \mathcal{C}_m$  de clases [Schürmann 1996]
- Regresión lineal y no lineal [Hartman et al. 1990]
- Series temporales [Shah 1998]
- Control de procesos [Hunt et al. 1991]
- Robótica [Kröse and van der Smagt 1996]
- Optimización [Polyak 1987]
- Procesamiento de señales [Haykin 1991]

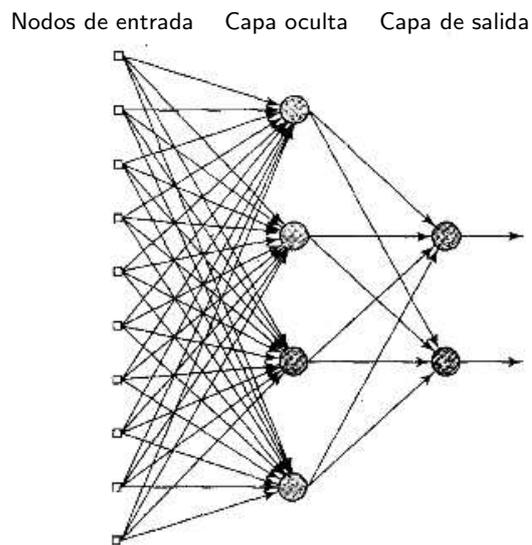


Figura 5: Esquema de una RNA de dos capas.

## 5. Análisis y conclusiones

A lo largo de las secciones hemos introducido los principios básicos de RNAs y hemos ejemplificado su uso utilizando un caso particular de red, el Perceptron, para tareas de clasificación lineal, con resultados positivos. No obstante,

la importancia de RNAs radica en problemas de tipo no lineal. Por su naturaleza experimental, su base teórica es en su gran mayoría heurística, por lo que su implementación se sustenta principalmente en métodos basados en la experiencia. Sin embargo, esta rama de investigación está en vías de desarrollo, y los resultados obtenidos hasta el momento son prometedores para el futuro de RNAs.

## Referencias

- [1] Haykin S. . *Neural Networks. A Comprehensive Foundation*, second edition. Pearson Prentice Hall, 1999.
- [2] Kecman V. . *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. The MIT Press, 2001.
- [3] Kröse B. and Van der Smagt P. 1996. *An introduction to Neural Networks*, eighth edition. University of Amsterdam.
- [4] Veelenturf L. . *Analysis and Applications of Artificial Neural Networks*. Prentice Hall, 1995.

Claudio J. Tablada  
(Email: cjtablada@yahoo.com.ar)

Germán A. Torres  
Facultad de Matemática, Astronomía y Física (FaMAF) – CIEM (CONICET)  
Universidad Nacional de Córdoba  
Ciudad Universitaria (5000), Córdoba, Argentina.  
(Email: torres@famaf.unc.edu.ar)