# REDES NEURONALES

# 2021

## Clase 15  Parte 1

Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba

Martes 5 de octubre 2021

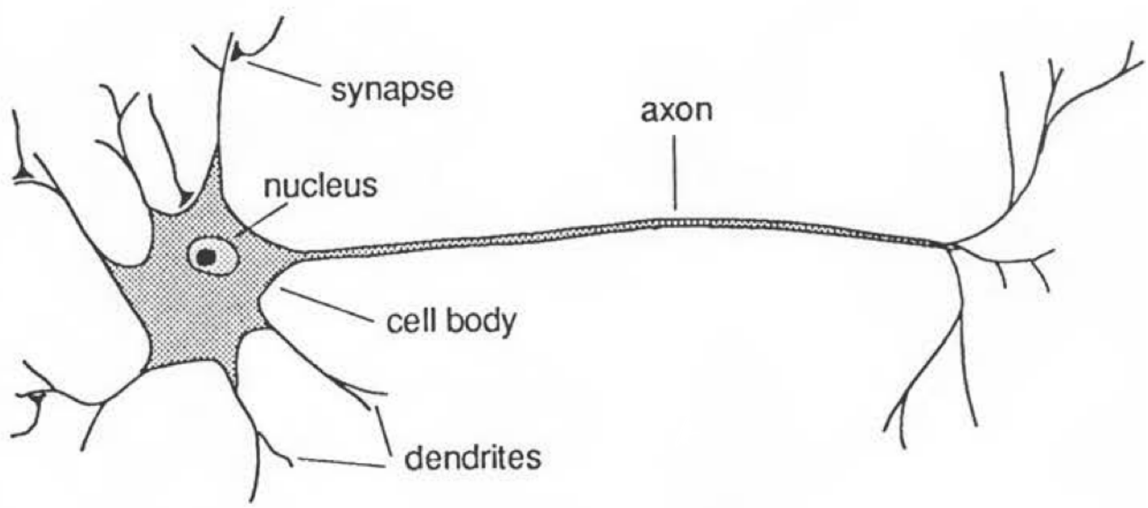
http://www.famaf.unc.edu.ar/~ftamarit/redes2021

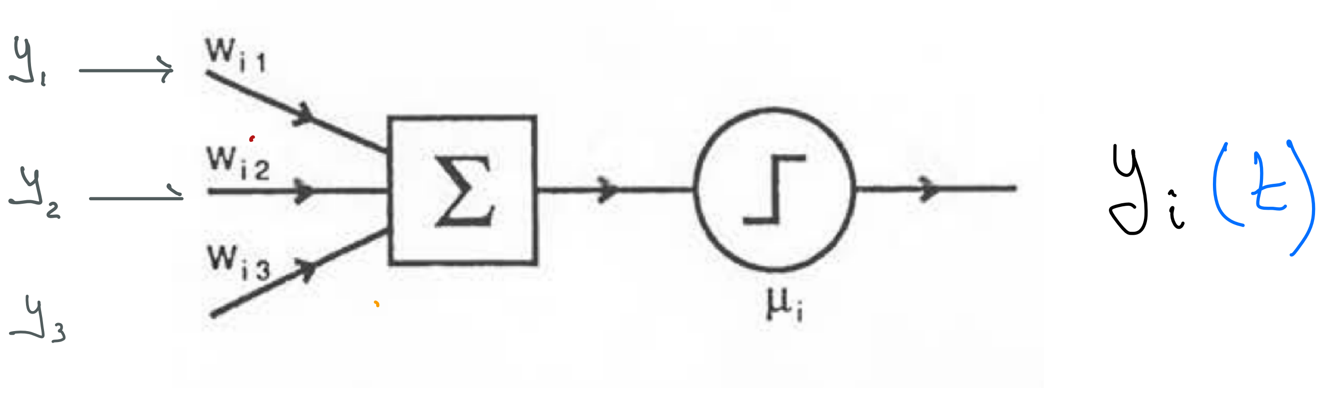
https://www.famaf.unc.edu.ar/course/view.php?id=798

## La neurona de McCulloch y Pitts

Comenzamos nuestro camino desde las neuronas artificiales hacia las redes neuronales profundas con la definición de la neurona matemática más simple que capta la esencia del funcionamiento de las neuronas biológicas, la cual se conoce como neurona de McCulloch y Pitts. Veremos que desde su aparición en la literatura hasta hoy, siendo la base de la inteligencia artificial neuronal.

En 1943 el neurólogo y cibernético norteamericano Warren McCulloch junto al matemático especialista en lógica Walter Pitts propusieron este modelo que funciona como unidad de umbral.






Neurona *i*

El gráfico esquematiza el modelo neuronal más simple que conocemos y que permite ensamblar grandes conglomerados de neuronas.

$y_1, y_2, y_3$ representan los señales que llegan de las cada neurona pre sinóptica

$W_{i1}, W_{i2}, W_{i3}$ representan las eficacias de cada sinopsis que le envía señales

$\mu_i$ representa el umbral de activación de la neurona *i*

El símbolo $\Sigma$ representa la suma ponderada que realiza la neurona i en la región de unión entre el cuerpo y el axón.

$$\boxed{\begin{aligned} h_i &= W_{i1}y_1 + W_{i2}y_2 + W_{i3}y_3 \\ &= \sum_{j=1}^{3} W_{ij}\, y_j \end{aligned}}$$

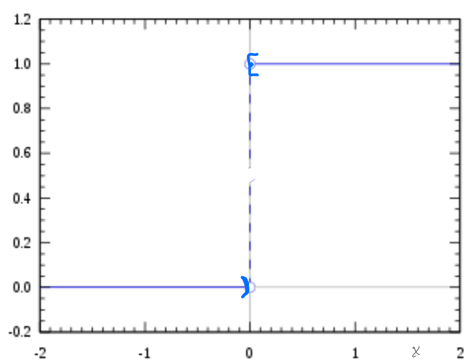
$h_i$ modela el potencial de acción, o sea, la diferencia se potencial entre el interior y el exterior de la neurona

$y_i$ representa la señal que la neurona *i* enviará a través de su axón.

representa la función de activación Heaviside

Nos interesa ver como la neurona *i* determina su valor $y_i$ en función de los valores de los estados de las tres neuronas pre sinópticas.

$$y_i = \Theta(h_i - \mu_i)$$




$$\Theta(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

*Función Heviside*

$$y_i(t) \begin{cases} +1 & \text{si la neurona } i \text{ dispara en tiempo } t \\ 0 & \text{si la neurona } i \text{ no dispara en tiempo } t \end{cases}$$

$$W_{ij} \begin{cases} > 0 & \text{si la sinápsis es exhitatoria} \\ = 0 & \text{si la sinápsis está prohibida o ausente} \\ < 0 & \text{si la sinápsis es inhibitoria} \end{cases}$$

*Observemos que hemos incluido el tiempo t tanto para describir la neurona i como para las neuronas presinápticas 1, 2 y 3 (que no vemos en el dibujo pero están)*

Si tuviéramos N neuronas pre-sinápticas en lugar de solo tres

$$y_i(t+1) = \Theta\left(h_i(t) - \mu_i\right)$$

$$= \Theta\left(\sum_{j=1}^{N} W_{ij}\, y_j(t) - \mu_i\right)$$

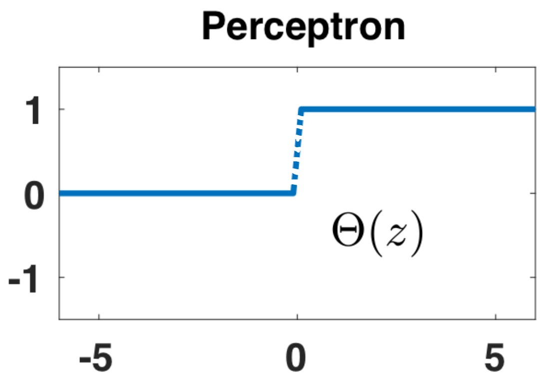
$$= \begin{cases} +1 & \text{si} \quad \sum_{j=1}^{N} W_{ij}\, y_j(t) \geq \mu_i \\ 0 & \text{si} \quad \sum_{j=1}^{N} W_{ij}\, y_j(t) < \mu_i \end{cases}$$

Entonces, la neurona *i* al tiempo *t* calcula la diferencia de potencial $h_i(t)$ en su cuerpo debido al input de las neuronas pre sinápticas como una suma ponderada y compara este valor con el umbral de disparo. Si es mayor o igual a cero, dispara (toma el valor +1). Si es menor que cero, se queda en reposo (toma el,valor 0)

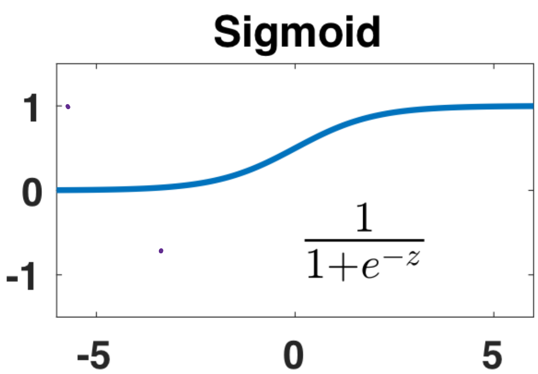
A lo largo de la historia de la Inteligencia Artificial, poco ha cambiado el modelado de neuronas desde 1943.

Las técnicas más modernas preservan la idea de una suma ponderada y la importancia de considerar los umbrales. Lo que usualmente cambia es la llamada función de activación, que para McCulloch y Pitts era la función Heaviside.

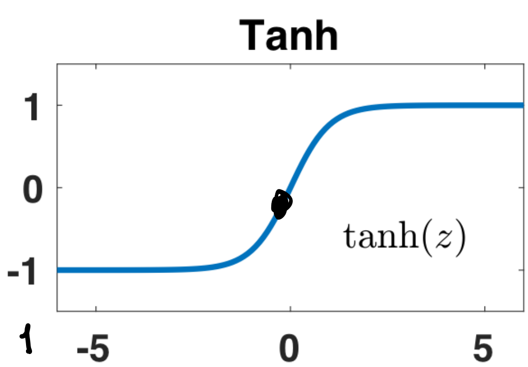
Hoy admitimos otras funciones que preservan la principal propiedad de la Heaviside, la no linealidad.



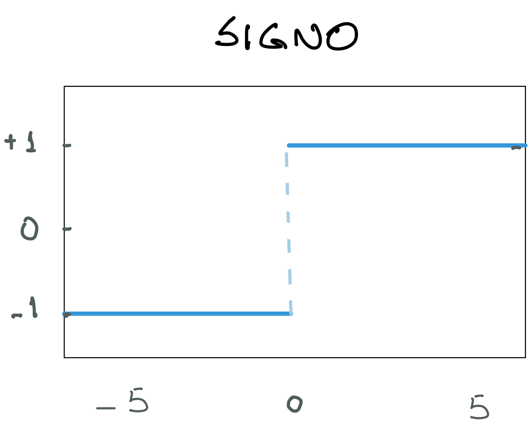

$y = 0$ ó $1$



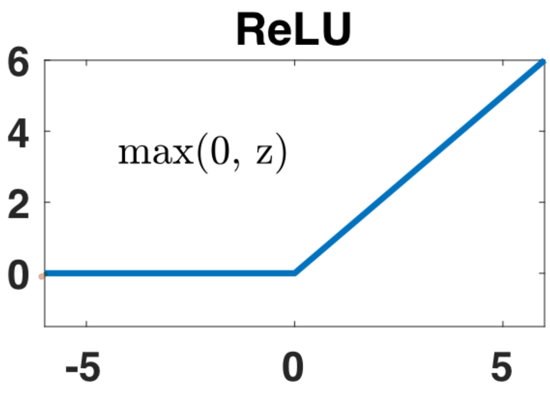

$0 < y < 1$



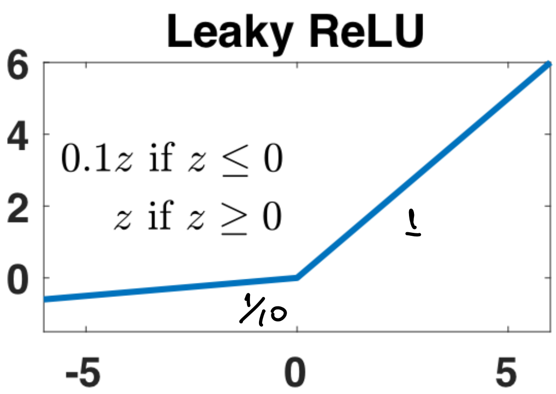

$-1 < y < 1$



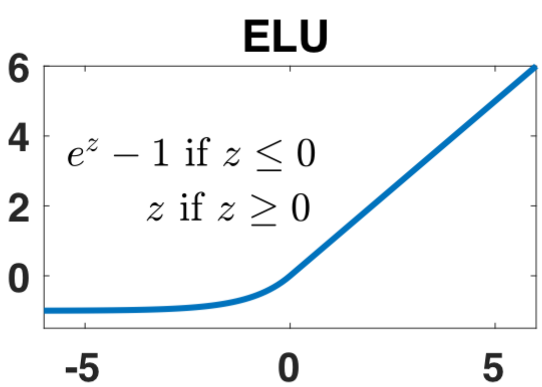

$y = -1$ ó $y = +1$




$0 \leq y \leq +\infty$




$-\infty \leq y \leq \infty$




$-1 < z < \infty$

**Una RED NEURONAL es un ensamblado de**

**muchas neuronas artificiales.**

$$\bar{y}(t) = (y_1(t), y_2(t), \ldots, y_N(t))$$

La red está definida no solo por la ***N*** neuronas sino también por

- Las ***NxN*** eficacias sinápticas, determinadas por medio de una matriz cuadrada.
- Los ***N*** umbrales pensados como un vector.

***APRENDER*****, en términos de Inteligencia Artificial, es asignar los valores de las *NxN* eficacias sinápticas y los *N* umbrales de forma tal que la red asigne, dinámicamente, una salida correcta ante una entrada dada.**

Ñ

Cuando nos referimos a APRENDER queremos decir que, partiendo de valores aleatorios de las sinápsis y los umbrales vamos variándolo lentamente hasta alcanzar valores que le permiten a la red hacer las conexiones entrada salida correctas.

$$w_{ij}^{\text{nuevo}} \longrightarrow w_{ij}^{\text{anterior}} + \Delta w_{ij}$$

$$\mu_i^{\text{nuevo}} \longrightarrow \mu_i^{\text{anterior}} + \Delta \mu_i$$

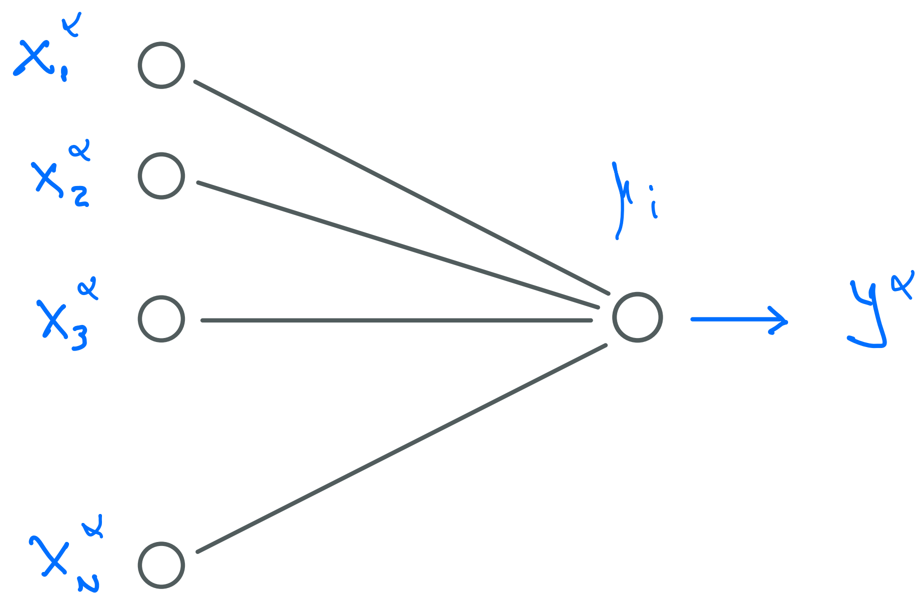
La actualización puede ser

- En línea: se aplica back propagation después de cada ejemplo.
- En batch: se aplica back propagation después de mostrar todo el conjunto de entrenamiento.

Para crear una red neuronal que aprenda de la experiencia a resolver algún problema particular debemos contar con un conjunto de ejemplos solucionados, o sea, muchos conjuntos de entrada y sus respectivas soluciones correctas.

$$\vec{x}^{\alpha} = (x_1^{\alpha}, x_2^{\alpha}, \ldots, x_N^{\alpha}) \longrightarrow \hat{y}^{\alpha}$$

Supongamos que nuestra red tiene una única capa de una única capa de entrada con *N* valores y una única neurona de salida




Digamos que tenemos P elementos en el conjunto de entrenemiento

$$\{\bar{x}^{\alpha}, \hat{y}^{\alpha}\}_{\alpha=1}^{P}$$

Ante la falta de mejor información elegimos las $N$ sinápsis y el único umbral $\mu$ al azar, con promedio cero y varianza 1.

Ahora podemos definir una función error que mida, matemáticamente, cuán bien resuelve nuestra red el problema de asociar las entradas $\bar{x}^{\alpha}$ del conjunto de entrenamiento con las salidas correctas $\hat{y}^{\alpha}$.
En realidad hay diferentes formas pero aquí veremos la más intuitiva, el Error Cuadratico Medio.

$$Ecm(\alpha) = (\hat{y}^{\alpha} - y^{\alpha})^2$$

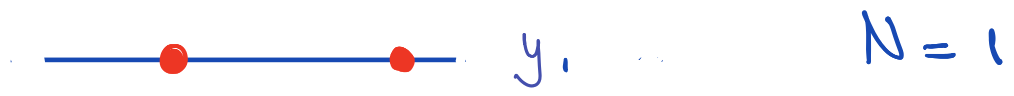
$$= (\hat{y}^{\alpha} - g(h^{\alpha}))^2$$

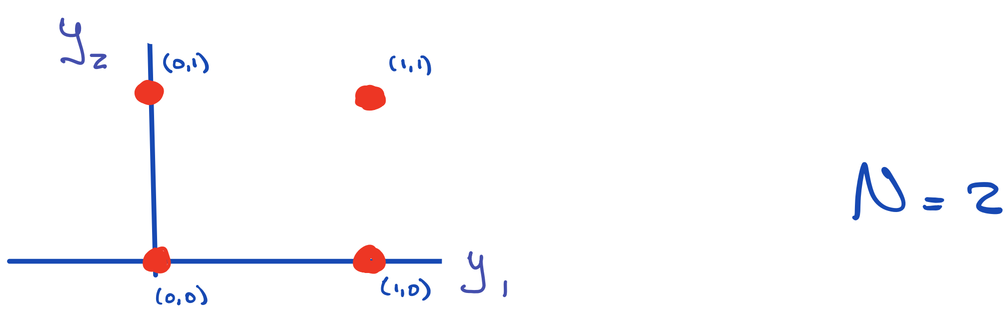
$$= (\hat{y}^{\alpha} - g(\sum w_{ik}\xi_k^{\alpha} - \mu_i))$$

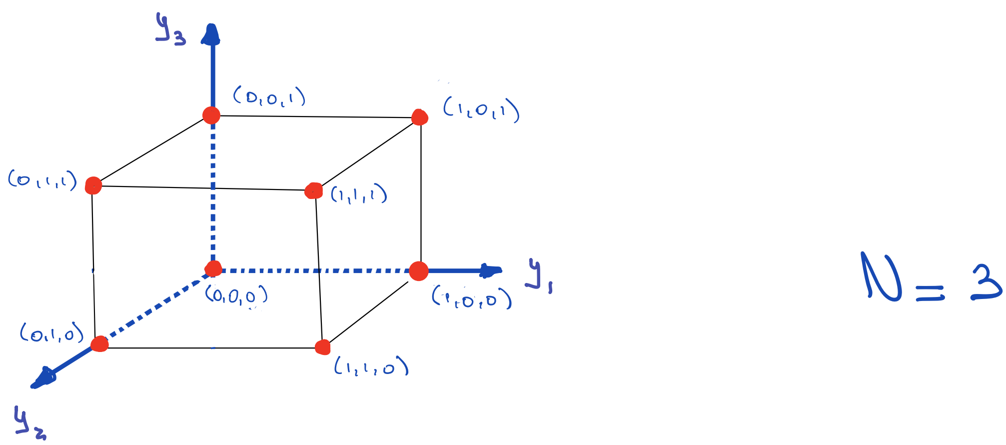
La salida que da la red

Vemos que el error $Ecm(\alpha)$ depende de todos los acoplamientos $w_{ij}$ y del umbral $\mu_i$.

Los estados de una red neuronal binaria con función de activación Heaviside viven en los vértices de un hipercubo de dimensión $N$, donde $N$ es el número de neuronas binarias.



Dado que la red neuronal tiene *N* neuronas de *2* estados posibles cada una,el número total de estados posibles es

$$2^N$$

Por ejemplo, con *N=4* neuronas, la red tiene *16* estados posibles

$$\vec{n}(0) = (1, 0, 0, 1)$$

$$\vec{n}(1) = (0, 0, 0, 1)$$

$$\vec{n}(2) = (1, 0, 1, 1)$$

$$\vec{n}(3) = (1, 0, 1, 1)$$

estado estacionario

Observemos que la red es un autómata determinista que ante un estímulo (un estado inicial) evolucionará hasta el estado final. El,estado final depende del estado inicial, la matriz de eficacias sinápticas y los umbrales.

$$\underset{\text{input}}{(1, 0, 0, 1)} \longrightarrow \underset{\text{output}}{(1, 0, 1, 1)}$$