

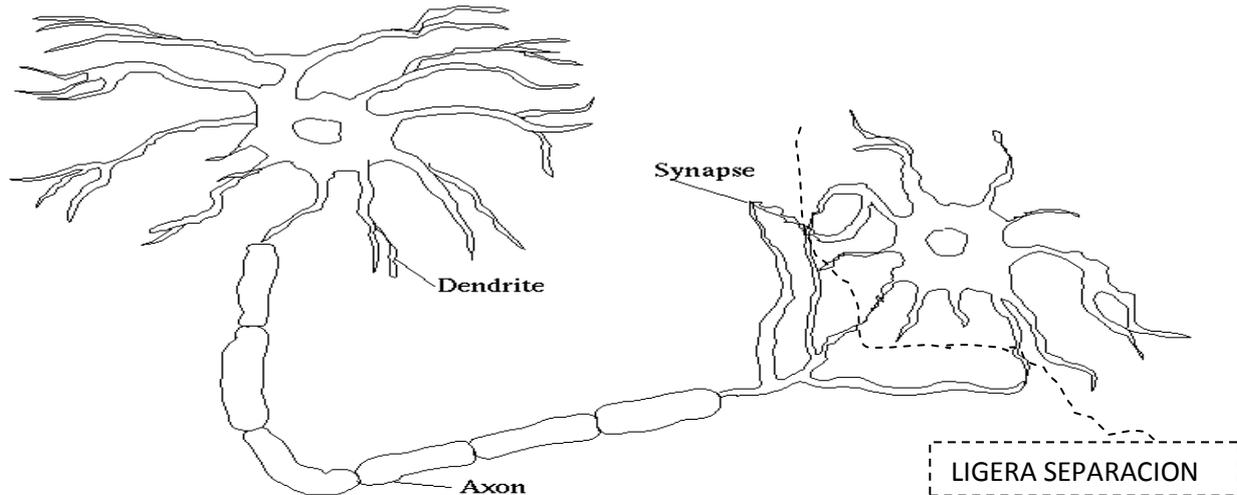
# RED NEURONAL

Las Redes Neuronales Artificiales (las cuales llamaremos RNA) son dispositivos o software programado de manera tal que funcionen como las neuronas biológicas de los seres vivos.

A continuación hablaremos un poco sobre ellas.

## Neuronas Biológicas

Nuestros cerebros cuentan con millones de neuronas que se interconectan para elaborar "Redes Neuronales". Estas redes ejecutan los millones de instrucciones necesarias para mantener una vida normal.



-  Las RNA están compuestas de un gran número elementos de procesamiento altamente interconectados (Neuronas) trabajando al mismo tiempo para la solución de problemas específicos.
-  Toma de decisiones, reconocimiento de formas, modelos, etc. Las RNA, aprenden de la experiencia.
-  Las RNA están compuestas de muchos elementos sencillos que operan en paralelo, el diseño de la red está determinado mayormente por las conexiones entre sus elementos.

## Pesos

Las RNA pueden tener factores de pesos fijos o adaptables.

1. Las que tienen **pesos adaptables** emplean leyes de aprendizaje para ajustar el valor de la **fuerza de una interconexión**.
2. Si las neuronas utilizan **pesos fijos**, entonces su tarea deberá estar previamente definida.

## Dos tipos de aprendizaje.

**1-Aprendizaje Supervisado:** ocurre cuando se le proporciona a la red tanto la entrada como la salida correcta, y la red ajusta sus pesos tratando de minimizar el error de su salida calculada.

Por ejemplo el reconocimiento de patrones.

**2-Aprendizaje no supervisado:** se presenta cuando a la red se le proporcionan únicamente los estímulos, y la red ajusta sus interconexiones basándose únicamente en sus estímulos y la salida de la propia red.

*Las leyes de aprendizaje determinan como la red ajustará sus pesos utilizando una función de error o algún otro criterio. La ley de aprendizaje adecuada se determina en base a la naturaleza del problema que se intenta resolver.*

## FASES

Las RNA adaptables tienen dos fases en su operación.

**1-Entrenamiento de la red.** El usuario proporciona a la red un número "adecuado" de estímulos de entrada, y de salida, la red entonces ajusta su pesos de interconexión o sinapsis hasta que la salida de la red está "lo suficientemente cerca" de la salida correcta.

## FASES

**2-Recuperación de lo aprendido.** A la red se le presenta un conjunto de estímulos de entrada y esta simplemente calcula su salida.

Cuando la red emplea entrenamiento no supervisado, algunas veces será necesario que reajuste su sinapsis durante la fase de recuperación.

Las RNA generan ellas mismas sus propias "reglas", para asociar la respuesta a su entrada; es decir, ***aprende por ejemplos y de sus propios errores.***

El conocimiento de una RNA se encuentra en la función de activación utilizada y en los valores de sus pesos.

Las redes neuronales formadas por los perceptrones son capaces de:

Asociar y generalizar sin reglas.

Han sido utilizadas con gran éxito para reconocer retornos de sonar bajo el agua, escritura a mano, voz, topografía de terrenos, controlar brazos de robots, evaluar datos personales, modelar fenómenos cognoscitivos, y, predecir tendencias financieras.

## Procedimientos Comunes

Asociación, evaluación, y reconocimiento de patrones.

Las redes neuronales son perfectas para problemas que son muy difíciles de calcular pero que no requieren de respuestas perfectas, sólo respuestas rápidas y buenas.

Tal y como acontece con el escenario bursátil en el que se quiere saber ¿compro?, ¿vendo?, ¿mantengo?.

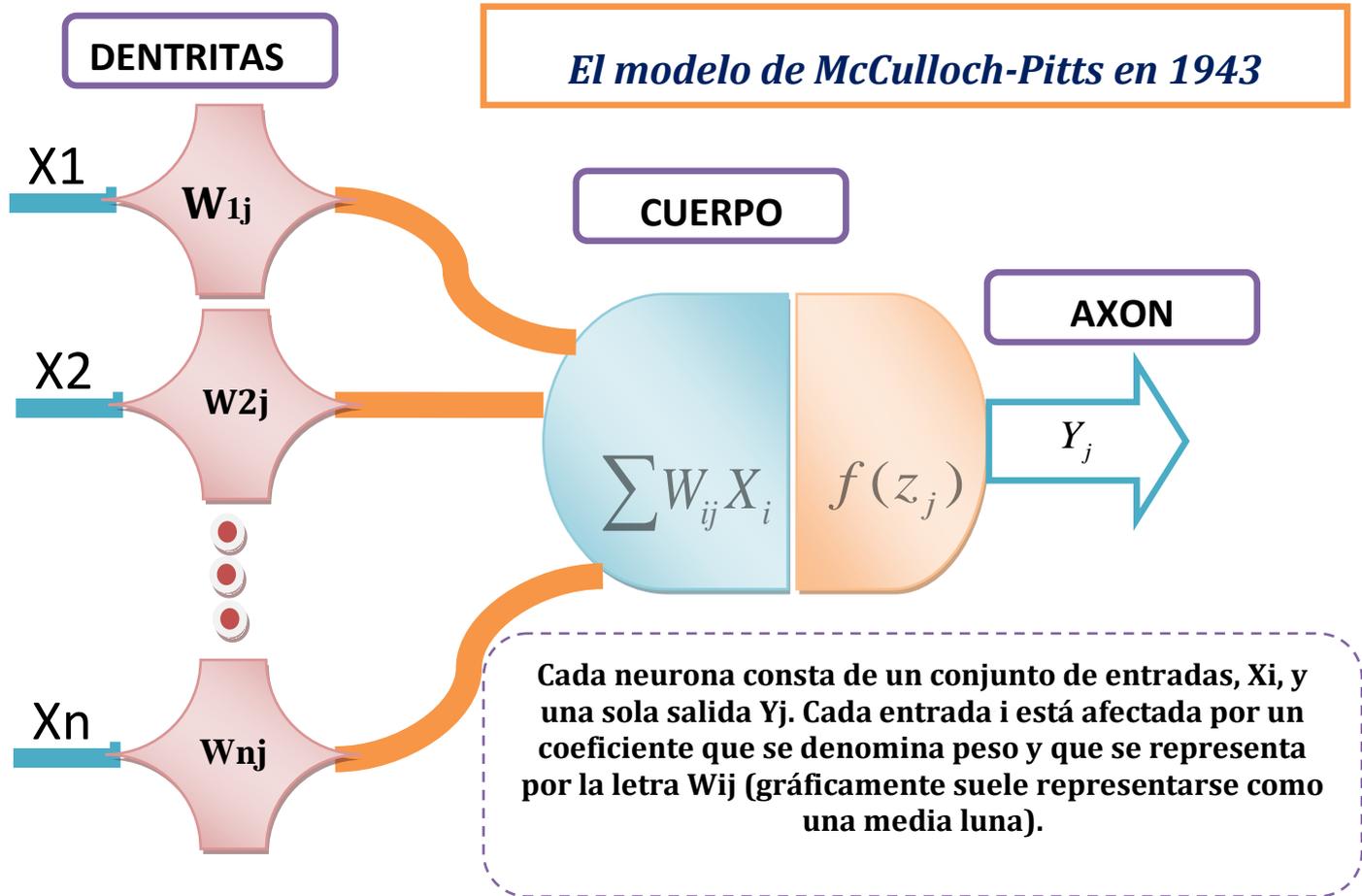
## En situaciones de reconocimiento:

Se desea saber ¿se parece? ¿es el mismo pero tienen una ligera modificación?

Por otra parte, las redes neuronales son muy malas para:

- 👤 Cálculos precisos.
- 👤 Procesamiento serie.
- 👤 No reconocen nada que no tenga inherentemente algún tipo de patrón.

***¿Qué sucedería si usarías neuronas a juegos de lotería?***



$$\sum W_{ij} X_i \quad : \text{FUNCION SUMA}$$

$$f(z_j) : \text{FUNCION DE TRANSFERENCIA}$$

La cantidad calculada como la suma del producto de cada entrada multiplicada por su respectivo peso se denomina *activación* de la neurona,  $Z_j$ . La salida,  $Y_j$ , de la neurona es una función de la activación de ésta. Es decir:

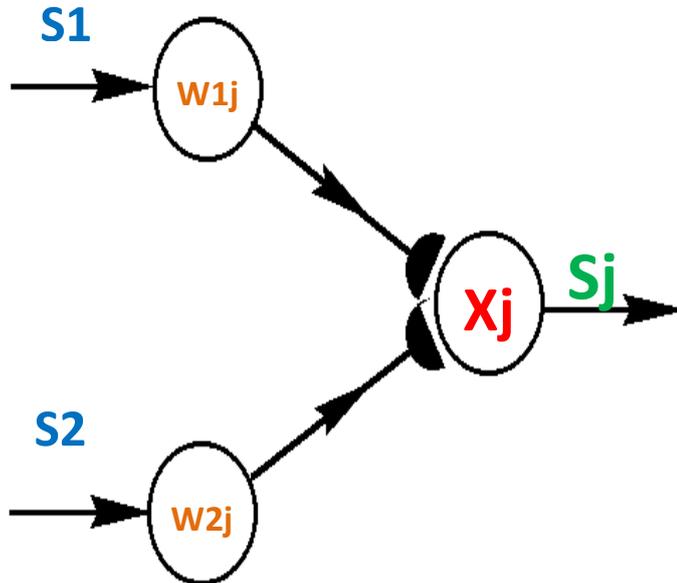
$$Z_j = \text{Sum} ( X_i * w_{ij} ) + Q_j ,$$

$$Y_j = f(z_j)$$

-   $Q_j$  es un valor «umbral»,
-   $f(z_j)$  es una función de la activación de la neurona.

En términos generales:

“Una red neuronal no se programa aunque se utilice un programa para simularlo, se «educa». La red es capaz de retener y asociar el conocimiento a través de la adaptación de los pesos de las neuronas siguiendo *una regla de aprendizaje*. Estas reglas son ecuaciones expresadas en función de las entradas y salidas de las neuronas y describen la forma de variación de los pesos. En definitiva, son el instrumento empleado por las neuronas para adaptarse a la información que se le presenta.



## *El Perceptron de Rosenblatt.*

*$S_1, S_2$ : Unidades*

El Perceptron se diseñó para trabajar con patrones de entrada y salida de tipo binario.

Así, la salida será **1** cuando la activación de la neurona alcance un valor. Y **0** en caso contrario.

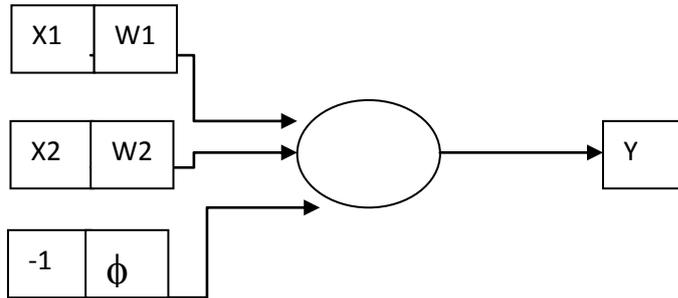
$$X_j = S_1 * W_{1j} + S_2 * W_{2j}$$

$$S_j = \begin{cases} 1 & \text{si } X_j > h ; \\ 0 & \text{si } X_j \leq h \end{cases}$$

El error viene dado por la diferencia entre la salida deseada  $b_j$ , y la salida real de la neurona,  $S_j$ . Es decir:

$$d_j = b_j - s_j$$

**¿Cuándo se cumple  $b_j$ ?**



<b>X1</b>	<b>X2</b>	<b>X1VX2</b>
<b>V</b>	<b>V</b>	<b>V</b>
<b>V</b>	<b>F</b>	<b>V</b>
<b>F</b>	<b>V</b>	<b>V</b>
<b>F</b>	<b>F</b>	<b>F</b>

# OR

X1	X2	X1VX2
1	1	1
1	-1	1
-1	1	1
-1	-1	-1

Ahora usaremos otras fórmulas de activación, con umbral  $\phi$  - 0.4

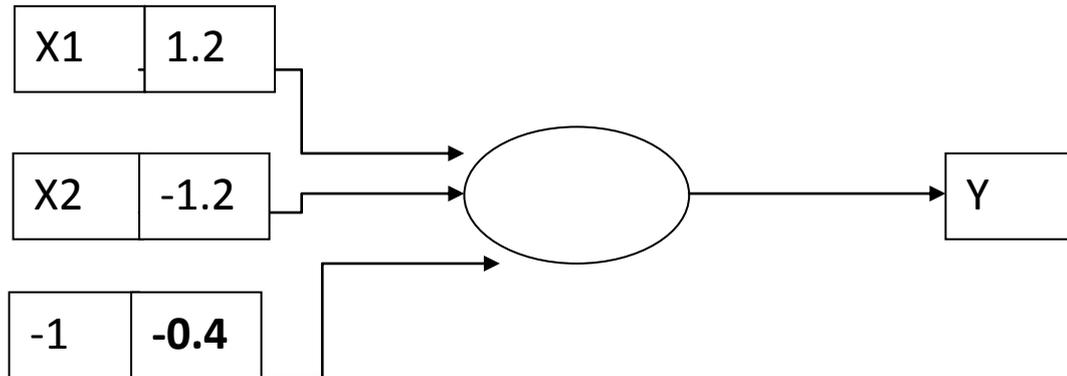
$$y = \begin{cases} 1 & \text{si } S_1 * W_1 + S_2 * W_2 \geq h; \\ 0 & \text{si } S_1 * W_1 + S_2 * W_2 < h \end{cases}$$

Elegimos pesos aleatorios:

Pesos  $W_1=1.2$ ,  $W_2=-1.2$ , Umbral  $\phi=-0.4$

Factor de Aprendizaje  $E=0.5$

Nuestro Modelo entonces es:



APLIQUE UD. LA FORMULA DE ACTIVACION... Y CONTRASTE

## SUS RESULTADOS.

Ahora aplicamos las formula de activación:

Para la entrada **(1,1)** T=1

$$Y = 1*(1.2) + 1*(-1.2) - 1*(-0.4) = 0.4 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada **(1,-1)** T=1

$$Y = 1*(1.2) + -1*(-1.2) - 1*(-0.4) = 2.8 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada **(-1,1)** T=1

$$Y = -1*(1.2) + 1*(-1.2) - 1*(-0.4) = -2 \quad ES < 0 \Rightarrow Y = -1 \quad \text{¿?}$$

Para la entrada **(-1,-1)** T= -1

$$Y = -1*(1.2) + -1*(-1.2) - 1*(-0.4) = 0.4 \quad ES \geq 0 \Rightarrow Y=1 \text{ ¿?}$$

Como podemos observar, la entrada  $(-1, 1)$ , necesitan un reajuste de sus pesos. Para ello utilizaremos la siguiente ecuación.

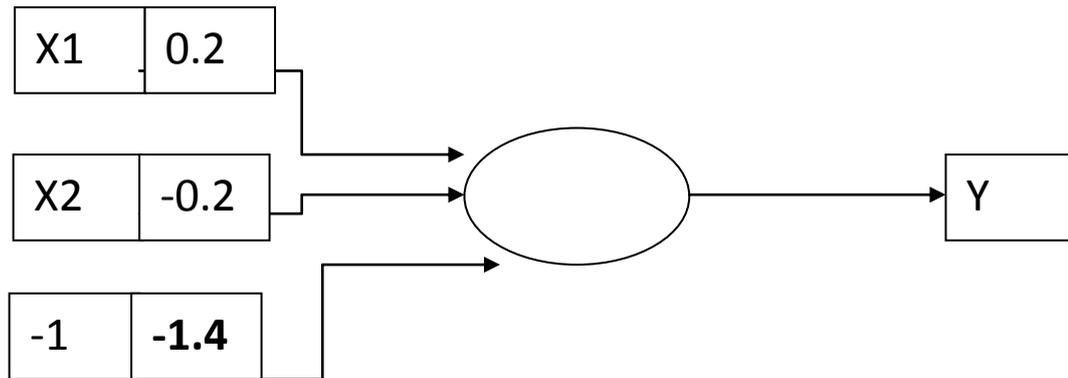
$$\Delta W_{ij}(t) = w_{ij}(0) + 2\epsilon \cdot (t_i) X_j$$

$$W_1 = 1.2 + 2*(0.5)*(1)*(-1) = 0.2$$

$$W_2 = -1.2 + 2*(0.5)*(1)*(1) = -0.2$$

$$\phi = -0.4 + 2*(0.5)*(1)*(-1) = -1.4$$

## Nuestro nuevo modelo de Perceptrón



Y nuevamente procedemos a repetir todos los procesos anteriormente realizados:

Para la entrada  $(1,1)$   $T=1$

$$Y = 1*(0.2) + 1*(-0.2) - 1*(-1.4) = 1.4 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada  $(1,-1)$   $T=1$

$$Y = 1*(0.2) + -1*(-0.2) - 1*(-1.4) = 1.8 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada  $(-1,1)$   $T=1$

$$Y = -1*(0.2) + 1*(-0.2) - 1*(-1.4) = 1 \quad ES \geq 0 \Rightarrow Y=1 \quad \text{OK}$$

Para la entrada  $(-1,-1)$   $T=-1$

$$Y = -1*(0.2) + -1*(-0.2) - 1*(-1.4) = 1.4 \quad ES \geq 0 \Rightarrow Y=1 \quad \text{¿?}$$

Como podemos observar, la entrada **(-1,-1)** , necesitan un reajuste de sus pesos. Para ello utilizaremos la siguiente ecuación.

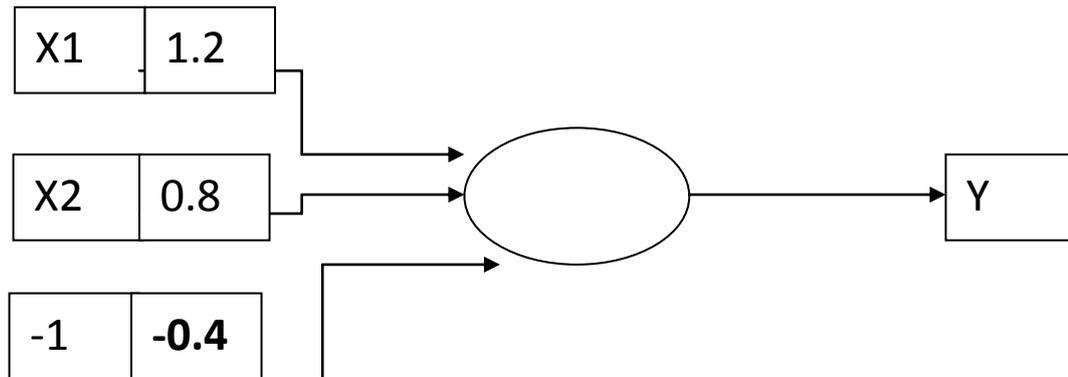
$$\Delta W_{ij}(t) = w_{ij}(0) + 2\epsilon \cdot (t_i) X_j$$

$$W1 = 0.2 + 2*(0.5)*(-1)*(-1) = 1.2$$

$$W2 = -0.2 + 2*(0.5)*(-1)*(-1) = 0.8$$

$$\phi = -1.4 + 2*(0.5)*(-1)*(-1) = -0.4$$

Pues nuestro modelo será ahora:



Para la entrada  $(1,1)$   $T=1$

$$Y = 1*(1.2) + 1*(0.8) - 1*(-0.4) = 2.4 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada  $(1,-1)$   $T=1$

$$Y = 1*(1.2) + -1*(0.8) - 1*(-0.4) = 0.8 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada  $(-1,1)$   $T=1$

$$Y = -1*(1.2) + 1*(0.8) - 1*(-0.4) = 0 \quad ES \geq 0 \Rightarrow Y=1$$

Para la entrada  $(-1,-1)$   $T=-1$

$$Y = -1*(1.2) + -1*(0.8) - 1*(-0.4) = -1.6 \quad ES \geq 0 \Rightarrow Y=1 \text{ ok}$$

**Por lo cual concluimos que los pesos definitivos son:**

**Peso1  $W1 = 1.2,$**

**Peso2  $W2 = 0.8,$**

**Umbral  $\phi = -0.4$**

Ahora Ud. desarrolle el problema de una compuerta lógica AND usando el algoritmo de Perceptrón Simple.