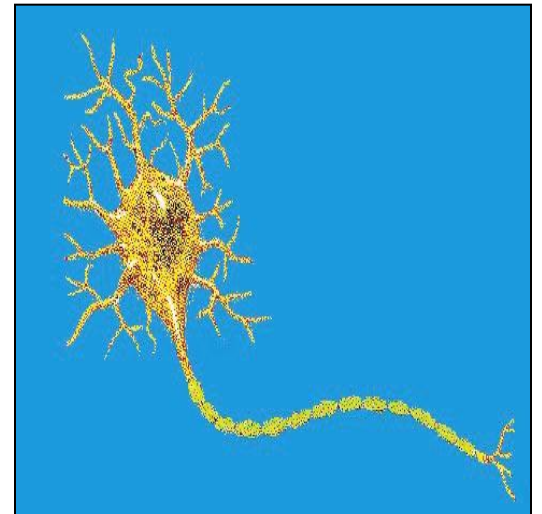




# ***MODELO DE UNA NEURONA***



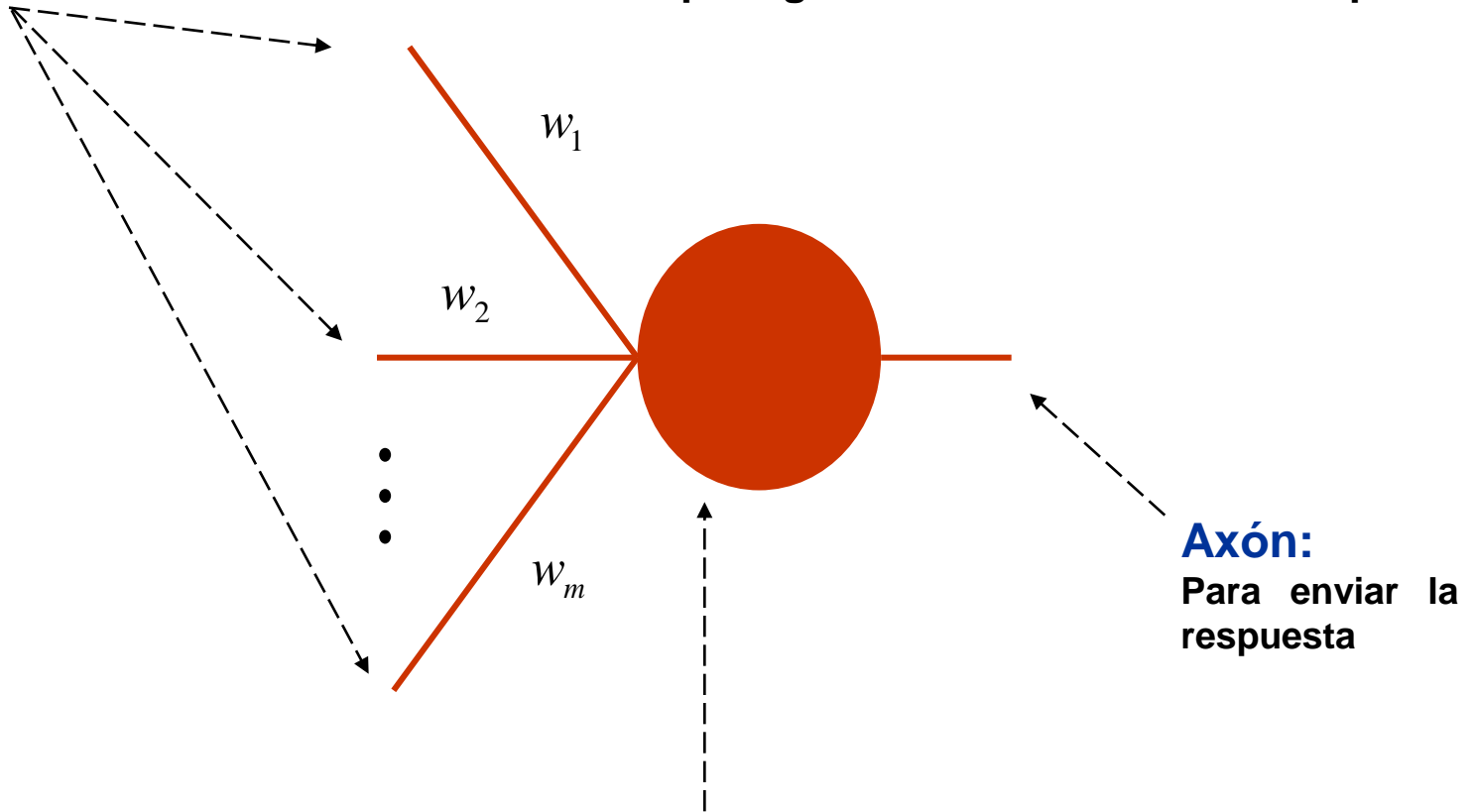


### Dendritas:

Para recibir el estímulo o establecer las conexiones sinápticas

### Pesos sinápticos:

Cada dendrita tendrá un peso  $w_j$  para guardar el conocimiento adquirido



### Núcleo:

Para procesar la información

### Axón:

Para enviar la respuesta

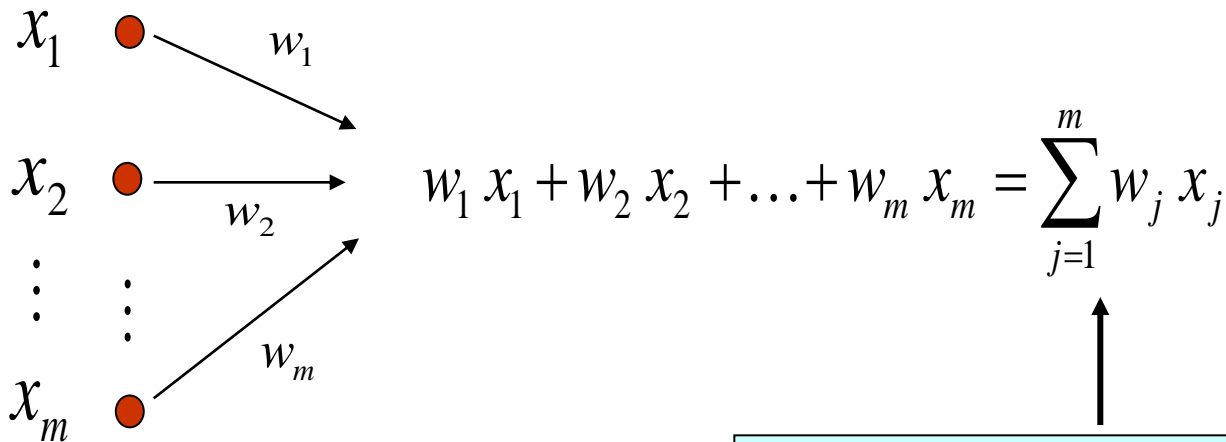


Los estímulos se consideran vectores

$$(x_1, x_2, \dots, x_m)$$

Cada entrada del vector corresponde a una variable en particular de la cual se tiene cierta cantidad de observaciones o valores.

Cuando se recibe el estímulo, cada entrada de este es multiplicada por el correspondiente peso sináptico de la dendrita que recibe dicho valor, y luego cada uno de estos resultados se suman.



Este valor se considera como el estímulo que recibe la neurona

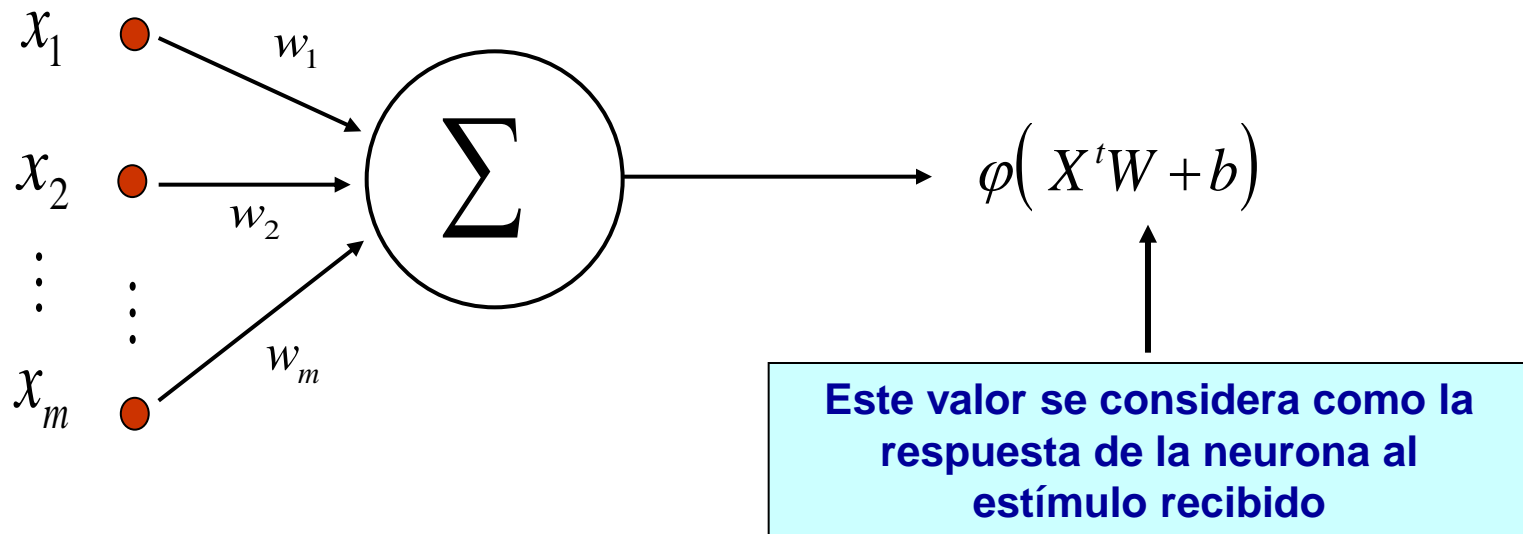


El estímulo es procesado en el núcleo mediante la operación

$$\varphi\left(\sum_{j=1}^m w_j x_j + b\right) = \varphi(X^t W + b)$$

$\varphi$  Se denomina función de transferencia o activación

$b$  es el parámetro de sesgo o *bias*





En conclusión, **los parámetros del modelo** de una neurona son:

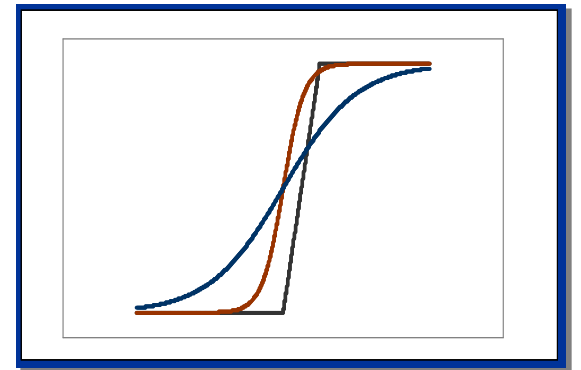
- La función de transferencia  $\varphi$
- El conjunto de pesos sinápticos  $W$
- El parámetro de sesgo  $b$

Este puede ser considerado como un peso sináptico  $w_0 = b$  asociado a una dendrita que recibe siempre el estímulo  $x_0 = 1$

$$\varphi\left(\sum_{j=1}^m w_j x_j + b\right) = \varphi\left(\sum_{j=0}^m w_j x_j\right) = \varphi(X^t W) \quad X = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{bmatrix} \quad W = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_m \end{bmatrix}$$



# ***ACERCA DE LA FUNCIÓN DE ACTIVACIÓN***





**La función de activación se utiliza para limitar el rango de valores de la respuesta de la neurona.**

**Generalmente los rangos de valores se limitan a  $[0,1]$  o  $[-1,1]$ , sin embargo otros rangos son posibles de acuerdo a la aplicación o problema a resolver.**

**Existen diversas funciones de activación y la decisión entre una u otra dependerá nuevamente de la aplicación o problema a resolver.**

**Existen funciones de activación comúnmente utilizadas y con las cuales se han obtenido resultados satisfactorios en diversas aplicaciones.**



## La función Umbral o *threshold function*

Esta función está definida como

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

La respuesta de la neurona es entonces 1 o 0.

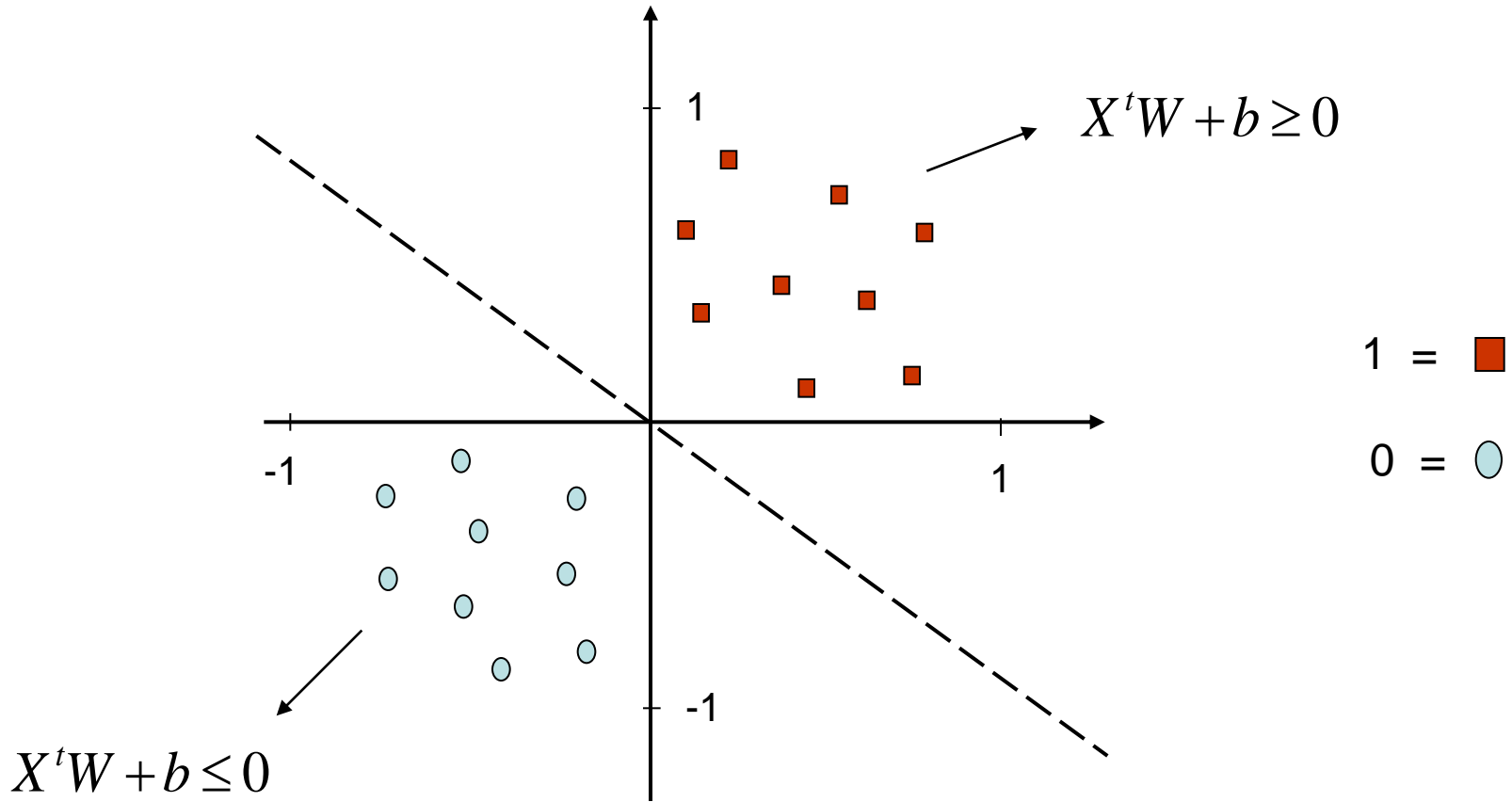
Estos valores se pueden asociar a *verdadero* o *falso* y esta condición dependerá de si

$$X^t W \geq -b \quad \text{o} \quad X^t W < -b$$

Generalmente se utiliza para establecer criterios de clasificación. Por ejemplo, 1=Elemento tipo A, 0=Elemento tipo B

En matlab la función se denomina *hard-limit function*





$$W = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

$$X^t W + b = x + y$$



**Ejemplo:**

**estímulo**  $X = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$       **Pesos sinápticos**  $W = \begin{bmatrix} 0.8 \\ 1 \\ 1 \\ 0.5 \end{bmatrix}$       **sesgo**  $b = -1$

**Cuál es la respuesta de la neurona si se utiliza la función umbral ?**

$$X^t W + b = 0 * 0.8 + 1 * 1 - 1 * 1 + 0 * 0.5 - 1 = -1$$

$$\varphi(-1) = 0$$

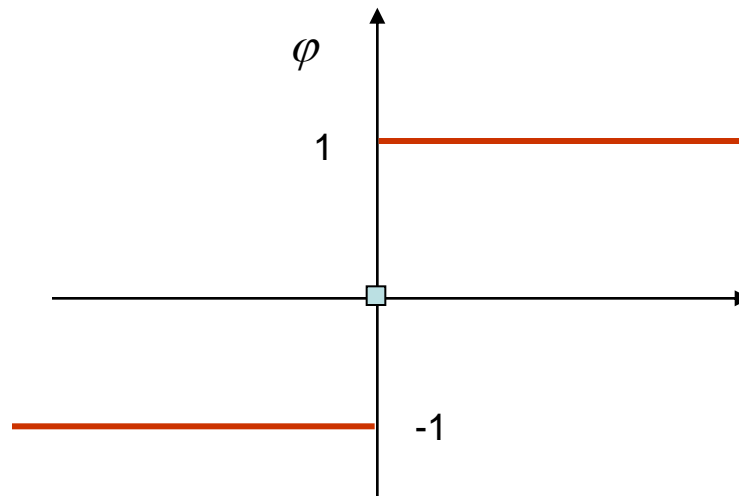


## La función signo

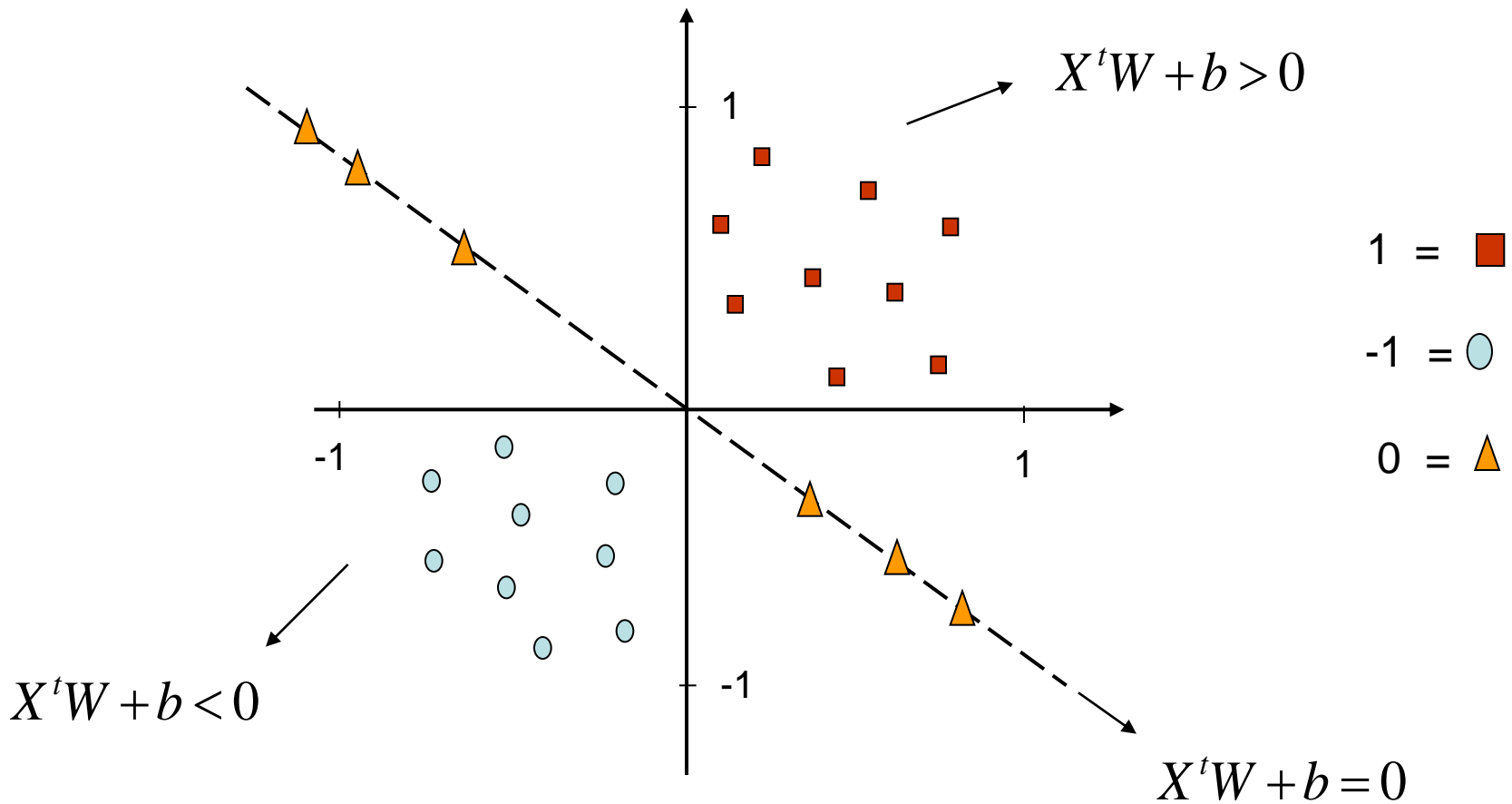
Esta función está definida como

$$\varphi(x) = \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases}$$

La respuesta de la neurona es entonces -1, 0 o 1.



En matlab la función se denomina *hardlimits*



$$X^t W + b = x + y$$



**Ejemplo:**

**estímulo**  $X = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$       **Pesos sinápticos**  $W = \begin{bmatrix} 0.8 \\ 1 \\ 1 \\ 0.5 \end{bmatrix}$       **sesgo**  $b = -1$

**Cuál es la respuesta de la neurona si se utiliza la función signo ?**

$$X^t W + b = 0 * 0.8 + 1 * 1 - 1 * 1 + 0 * 0.5 - 1 = -1$$

$$\varphi(-1) = -1$$



## La función logística

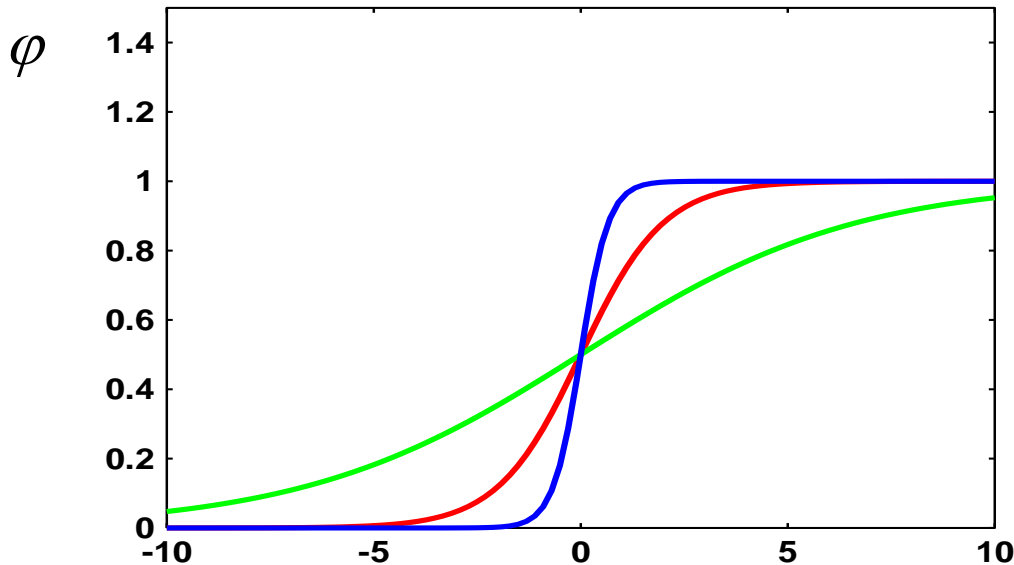
Esta función está definida como

$$\varphi(x) = \frac{1}{1 + \exp(-\alpha x)} \quad x \in \mathbb{R}, \alpha > 0$$

Es una de las funciones más utilizadas en la construcción de redes neuronales

Es continua a valores en  $[0, 1]$  e **infinitamente diferenciable**

Es la versión continua de la función umbral y se utiliza en problemas de aproximación



- $\alpha = 3$
- $\alpha = 1$
- $\alpha = 0.3$

En matlab la función se denomina *logsig*



**Ejemplo:**

**estímulo**  $X = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$       **Pesos sinápticos**  $W = \begin{bmatrix} 0.8 \\ 1 \\ 1 \\ 0.5 \end{bmatrix}$       **sesgo**  $b = -1$

**Cuál es la respuesta de la neurona si se utiliza la función logística con parámetro 1 ?**

$$X^t W + b = 0 * 0.8 + 1 * 1 - 1 * 1 + 0 * 0.5 - 1 = -1$$

$$\varphi(-1) = \frac{1}{1 + \exp(1)} = 0.2689$$



## La función tangente hiperbólica

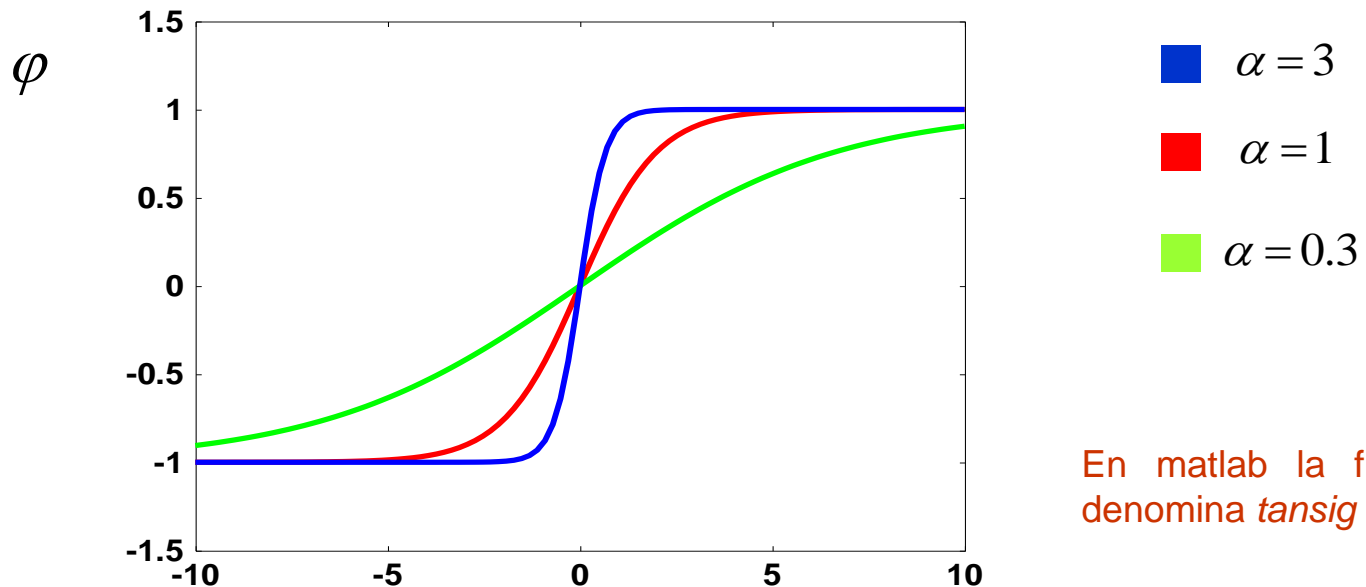
Esta función está definida como

$$\varphi(x) = \tanh(\alpha x / 2) = \frac{1 - \exp(-\alpha x)}{1 + \exp(-\alpha x)} \quad x \in \mathbb{R}, \alpha > 0$$

Es la versión continua de la función signo y se usa en problemas de aproximación

Es importante por sus propiedades analíticas

Es continua a valores en  $[-1, 1]$  e **infinitamente diferenciable**



En matlab la función se denomina *tansig*





**Ejemplo:**

**estímulo**  $X = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$       **Pesos sinápticos**  $W = \begin{bmatrix} 0.8 \\ 1 \\ 1 \\ 0.5 \end{bmatrix}$       **sesgo**  $b = -1$

**Cuál es la respuesta de la neurona si se utiliza la función tangente hiperbólica con parámetro 2 ?**

$$X^t W + b = 0 * 0.8 + 1 * 1 - 1 * 1 + 0 * 0.5 - 1 = -1$$

$$\varphi(-1) = \frac{1 - \exp(2)}{1 + \exp(2)} = -0.76159$$



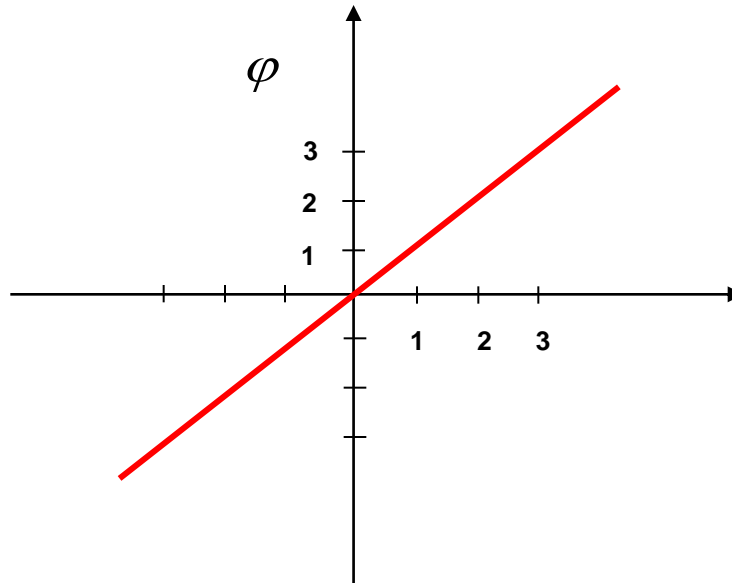
## La función Lineal

Esta función está definida como

$$\varphi(x) = x \quad x \in \mathbb{R}$$

No limita la respuesta de la neurona

Se utiliza en problemas de aproximación o estimación lineal

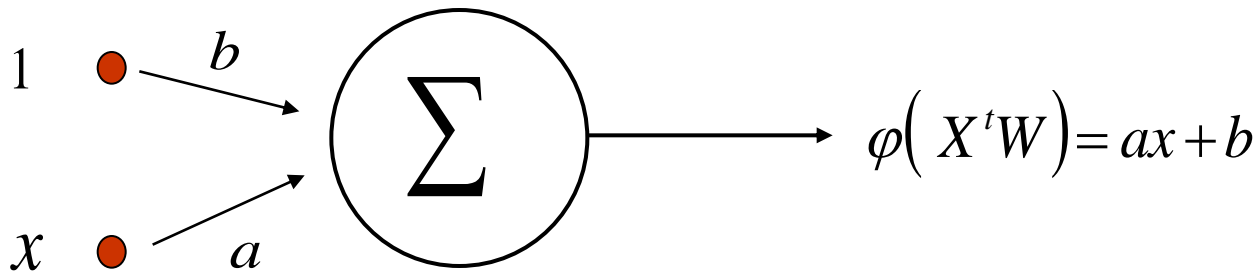


En matlab la función se denomina *purelin*



Una regresión lineal simple puede ser vista como la respuesta de una neurona con función de activación lineal

$$X = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad W = \begin{bmatrix} b \\ a \end{bmatrix} \quad X^t W = ax + b$$

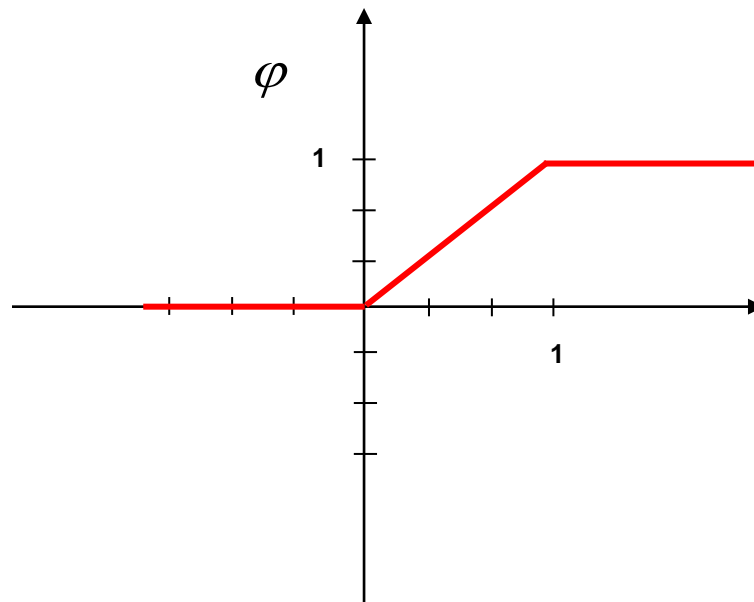




## La función Lineal acotada

Esta función está definida como

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



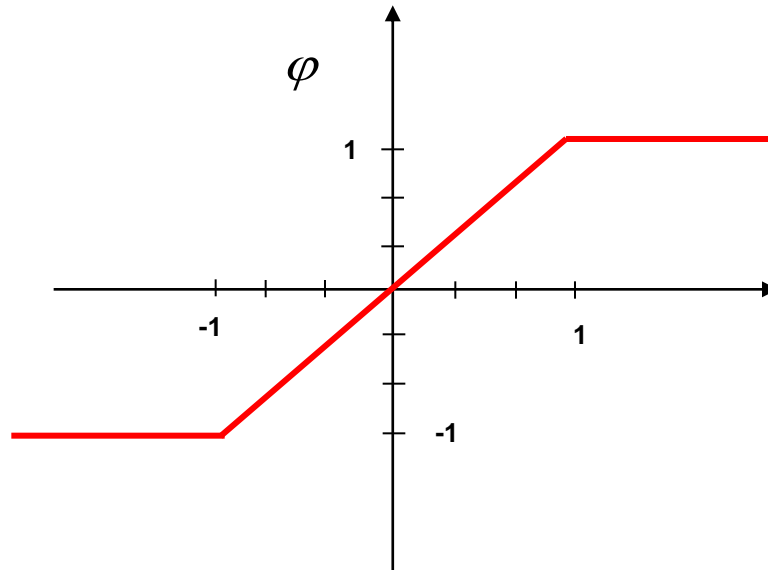
En matlab la función se denomina *satlin* (*saturating linear*)



## La función Lineal acotada simétrica

Esta función está definida como

$$\varphi(x) = \begin{cases} -1 & \text{si } x \leq -1 \\ x & \text{si } -1 \leq x \leq 1 \\ 1 & \text{si } x \geq 1 \end{cases}$$



En matlab la función se denomina *satlins* (*symmetric saturating linear*)



## Estímulo

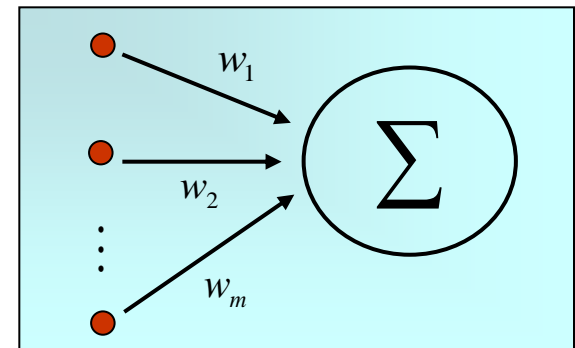
$$X^tW + b = 0*0.8 + 1*1 - 1*1 + 0*0.5 - 1 = -1$$

<b>Función de activación</b>	<b>Respuesta</b>
Umbral o <i>threshold</i>	0
Signo	-1
logística	0.2689
Tangente Hiperbólica	-0.76159
Lineal	-1
Lineal Acotada	0
Lineal Acotada Simétrica	-1

**La escogencia de la función de activación depende de la aplicación o problema a resolver !**



# ***ACERCA DE LOS PESOS SINÁPTICOS***





Recordemos que los pesos sinápticos sirven para guardar el **conocimiento adquirido**.

En analogía con el funcionamiento del cerebro, los valores de los pesos sinápticos se obtienen mediante un **proceso de aprendizaje**.

Este **proceso de aprendizaje** puede ser dividido en: *Paradigmas de aprendizaje y algoritmos de entrenamiento*.

- **Paradigmas de aprendizaje**

Corresponden a la manera en que la red neuronal interactúa o se relaciona con el ambiente en el cual opera. Básicamente existen 2 tipos de paradigmas:

**Aprendizaje supervisado** (*Supervised learning*)

**Aprendizaje no supervisado** (*Non supervised learning*)





- **Algoritmos de entrenamiento**

Son técnicas matemáticas o heurísticas bien establecidas a través de los cuales se obtienen los valores de los pesos sinápticos y el parámetro de sesgo. Estos algoritmos **NO modifican la función de activación** elegida para el modelo de red neuronal que se entrena.

Lo que hacen es modificar los pesos sinápticos utilizando el valor anterior de este mas un factor de corrección

$$W^{\text{nuevo}} = W^{\text{viejo}} + \Delta W$$

y se diferencian uno de otro en la forma en que dicha corrección es obtenida

Existen una amplia variedad de algoritmos de entrenamiento cada cual con sus fortalezas y debilidades.

Distintos paradigmas de aprendizaje junto con distintos algoritmos de entrenamiento producen diversas redes neuronales



**Existen muchos algoritmos de entrenamiento, algunos de los cuales se mencionan aquí:**

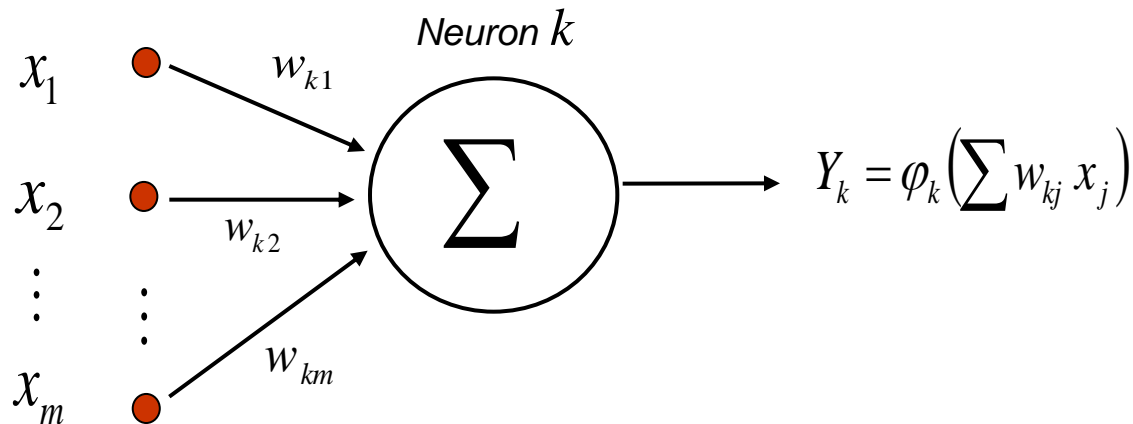
- **Error correction learning or delta rule:** es un aprendizaje supervisado, en donde se usa el error obtenido en la presentación del estímulo para determinar el factor de corrección.
- **Competitive learning:** las neuronas compiten entre si para hacerse activas Estableciendo representaciones de la data. Solo una se activa a la vez.
- **Hebbian learning:** correlaciona la actividad pre y post sináptica (si ambas neuronas se activan se fortalece la conexión y se debilita en caso contrario). Aquí el factor de corrección es multiplicativo entre la actividad pre y post sináptica.
- **Memory Based learning:** la red almacena data ya procesada en memoria. Un nuevo patrón es procesado según su cercanía (en sentido euclídeo) a un patrón en memoria.



## Error correction learning or delta rule

Se utiliza bajo un paradigma de entrenamiento supervisado.

La respuesta de la red neuronal se compara con la respuesta deseada y el error cometido se utiliza para modificar los pesos sinápticos



$$w_{kj}(\eta+1) = w_{kj}(\eta) + \alpha e_k(\eta) x_j(\eta)$$

$\Delta w_{kj}$

$\alpha$  = Constante de aprendizaje

$e_k = d_k - Y_k$  = Error cometido



## Hebbian learning or activity product rule

Es el postulado de entrenamiento más antiguo y se basa en consideraciones desarrolladas por Hebb, D.O (1949) en su libro *The organization of behavior: A neuropsychological theory*

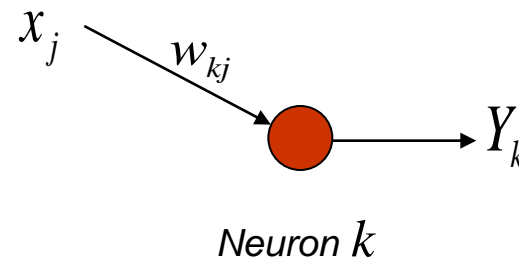
El postulado de Hebb establece que:

*Cuando dos neuronas conectadas (sinápsis) se activan simultáneamente, la relación entre estas se fortalece. En caso contrario, la relación se debilita o es eliminada.*

La formulación matemática apropiada es establecer entonces la correlación entre el estímulo de entrada y la respuesta a dicho estímulo y utilizar esta información para actualizar el valor de los pesos sinápticos

$$w_{kj}(\eta+1) = w_{kj}(\eta) + \alpha y_k(\eta) x_j(\eta)$$

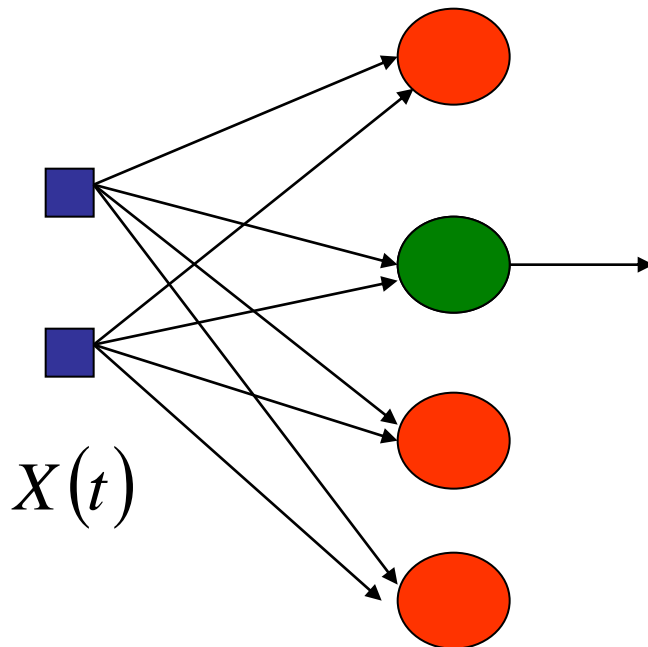
$\Delta w_{kj}$





## Competitive Learning

En este tipo de algoritmos se establece un **mecanismo de competencia** entre las neuronas y la respuesta de la red ante un determinado estímulo proviene exclusivamente de la **neurona ganadora**



$$W_j(t+1) = W_j(t) + \eta(t) [X(t) - W_j(t)]$$

$\Delta w_{kj}$

Es esta característica la que hace que este tipo de algoritmos sean apropiados para descubrir patrones en un conjunto de datos y que por lo tanto sean usados para clasificación



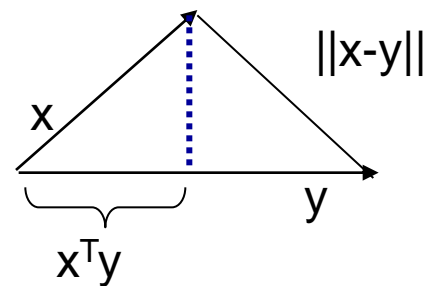
## Memory based learning

- En este tipo de entrenamiento las experiencias pasadas están almacenadas en la “**memoria**” de ejemplos que han sido clasificados correctamente.
- Un nuevo patrón de entrada es comparado con  $k$  patrones cercanos (concepto de vecindad) y luego es asignado la clasificación más frecuente en la vecindad.

Las **RBF** utilizan este tipo de postulado en su entrenamiento, al igual que las redes **SOM**.

### Distancias

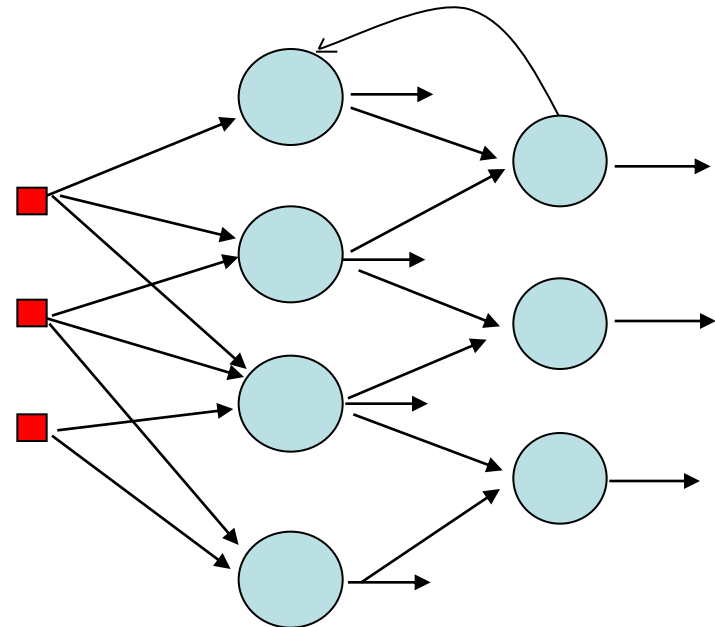
- Euclidea :  $\|x-y\|$
- Producto interno :  $x^T y$





# Arquitecturas

- Una capa, feedforward
- Multicapa feedforward
- Redes recurrentes





## Qué tipos de problemas se pueden resolver?

- Asociación de patrones: escritura. La red es presentada con muchos ejemplos de un patron hasta “guardar” el patron, luego al ser presentado una versión con ruido esta es capaz de reconocer el patrón.
- Clasificación: una la red es entrenada con ejemplos de clasificación conocida.
- Aproximación de funciones. Identificación, problemas inversos.
- Control: se conoce una señal de referencia y el objetivo es ajustar los parámetros libres para seguir la salida deseada.
- Filtro: son útiles en predicciones.