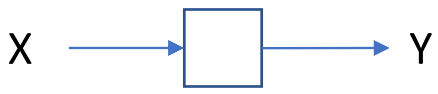
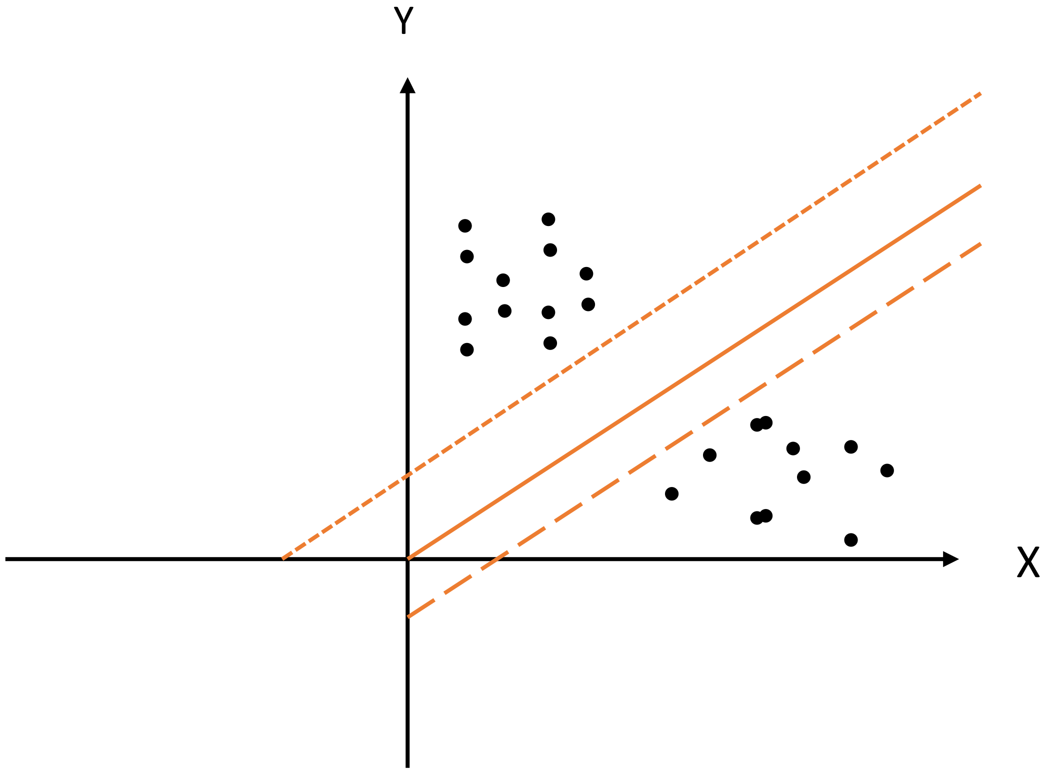
Perceptron y Retropropagación

Ismael López Juárez

Perceptrón

El perceptrón es la arquitectura neuronal mas simple (entrada-salida) y básicamente funciona como clasificador binario. Es muy útil en problemas en donde las variables en cuestión son linealmente separables.



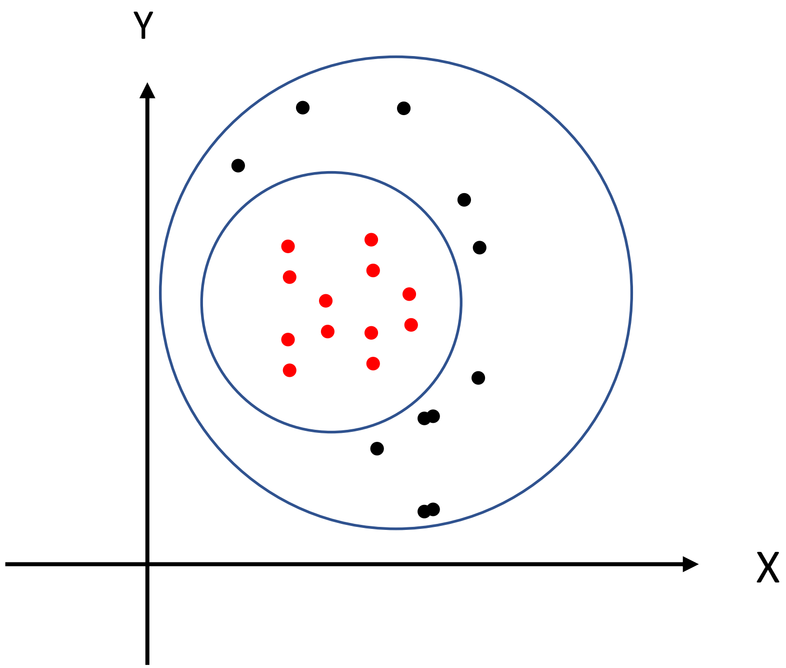


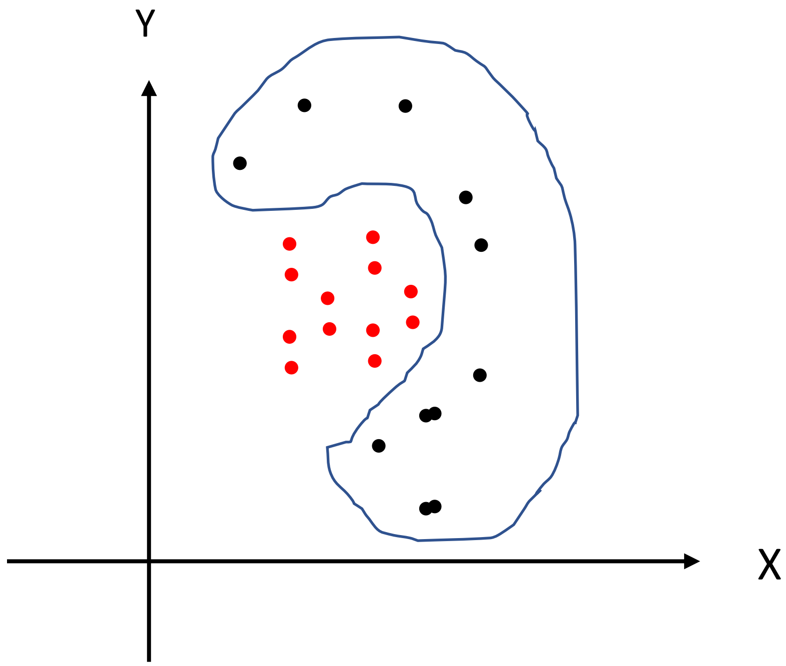
$$Y = WX \pm b$$

Perceptrón

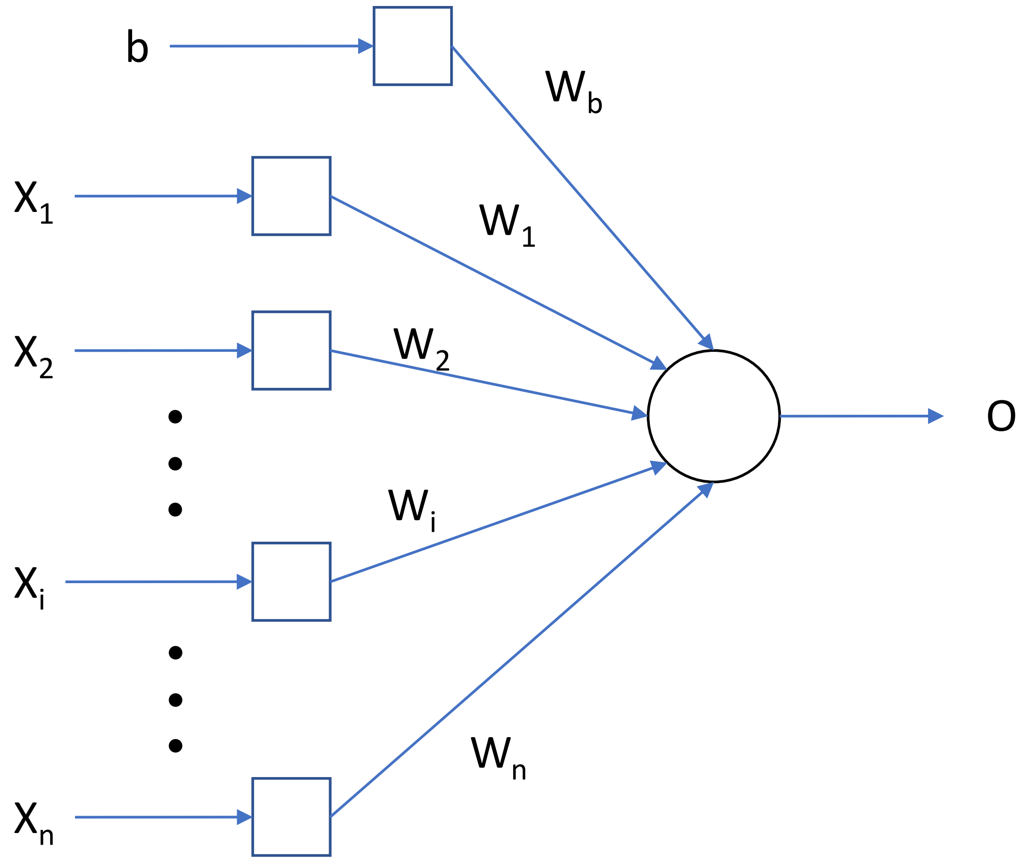
Es posible también utilizar el Perceptrón en problemas que involucran funciones que no son separables linealmente. Sin embargo, ello requiere una transformación de coordenadas esféricas por ejemplo, lo que supone mayor complejidad.

En este sentido se prefieren funciones no lineales que son mapeadas utilizando múltiples capas del perceptrón como veremos mas adelante.





Perceptrón



La activación de la neurona es una suma ponderada de sus pesos.
Es el producto punto $W \cdot X$

$$S^p = \sum_{i=1}^{n} W_i^p X_i^p + W_{n+1}^p$$

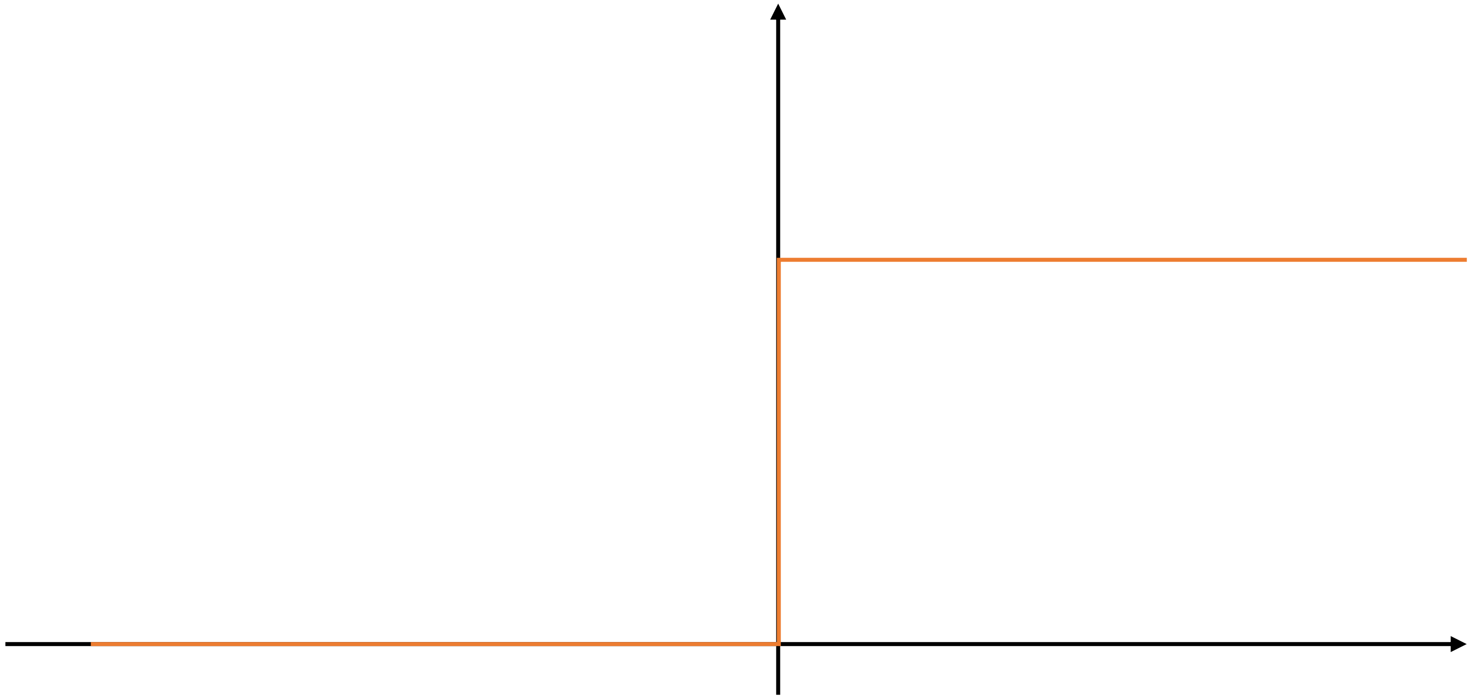
Para la salida utilizamos una funcion de umbral, dado que se trata de una red de clasificacion binaria:

$$O^p = f(S^p)$$

Umbral

$$O^p = f(S^p)$$

$$O^p = \begin{cases} 0, \ \forall \ f(S^p) \leq 0 \\ 1, \ \forall \ f(S^p) > 0 \end{cases}$$



Aprendizaje

Los pesos se actualizan de la siguiente forma:

$$W_i = W^{-1} + \Delta W_i$$

Donde: W^{-1} representa el peso en la iteración anterior

$$\Delta W_i = \eta(O - \hat{O})X_i$$

O representa la salida real

$\hat{O}$ representa la salida obtenida

Ejercicio en clase

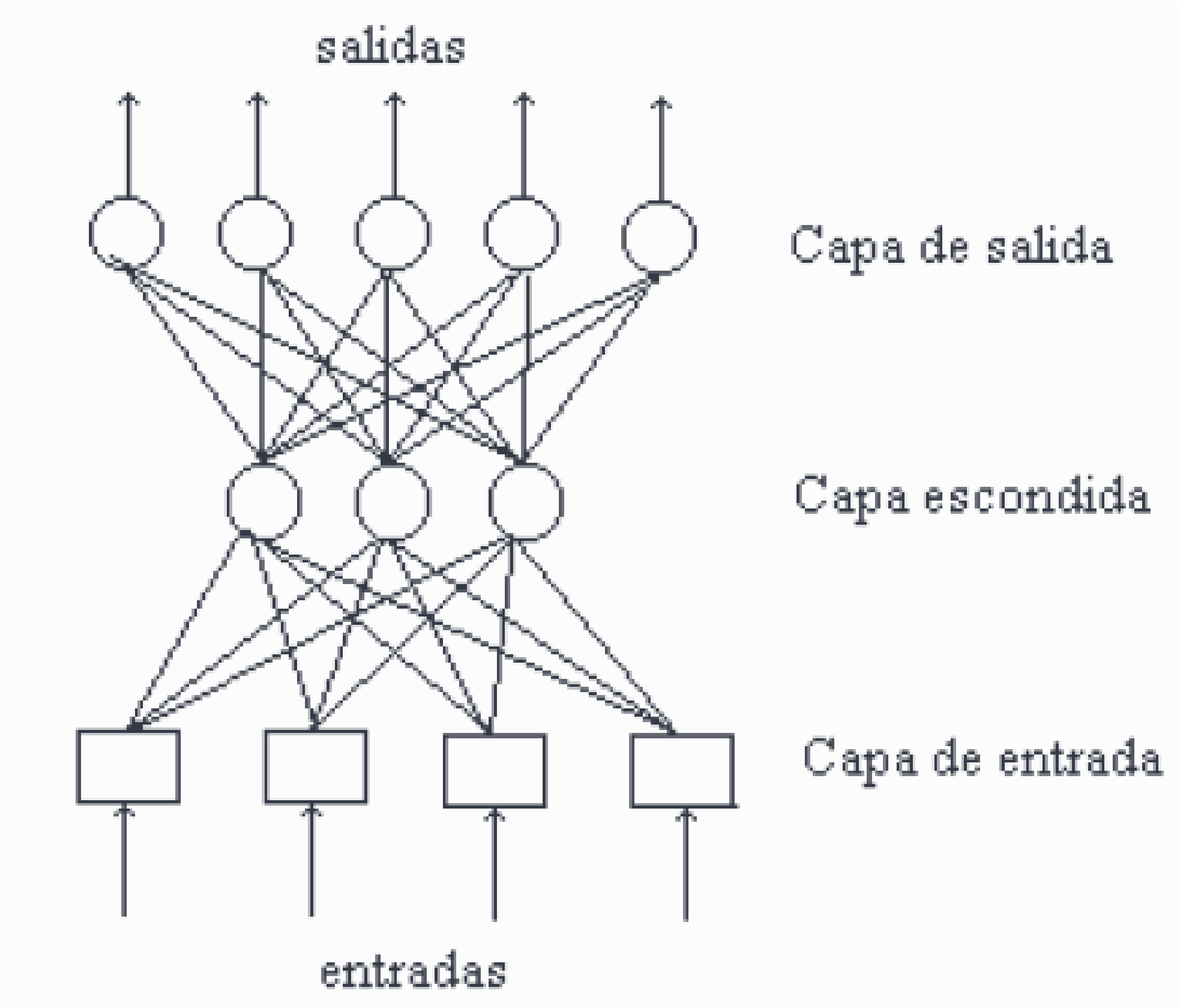
Epoca	bias	I0	I1	Yd	W0	W1	W2	Suma	Activacion	Error	Converge?	Razón Aprendizaje
1	1	0	0	0	1	1	1	1	1	-1		1
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	No Converge	
2	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	
3	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	
4	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	
5	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	
6	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	
7	1	0	0	0	0	1	1	0	0	0		
	1	0	1	1	0	1	1	1	1	0		
	1	1	0	1	0	1	1	1	1	0		
	1	1	1	1	0	1	1	2	1	0	Converge	

Tarea

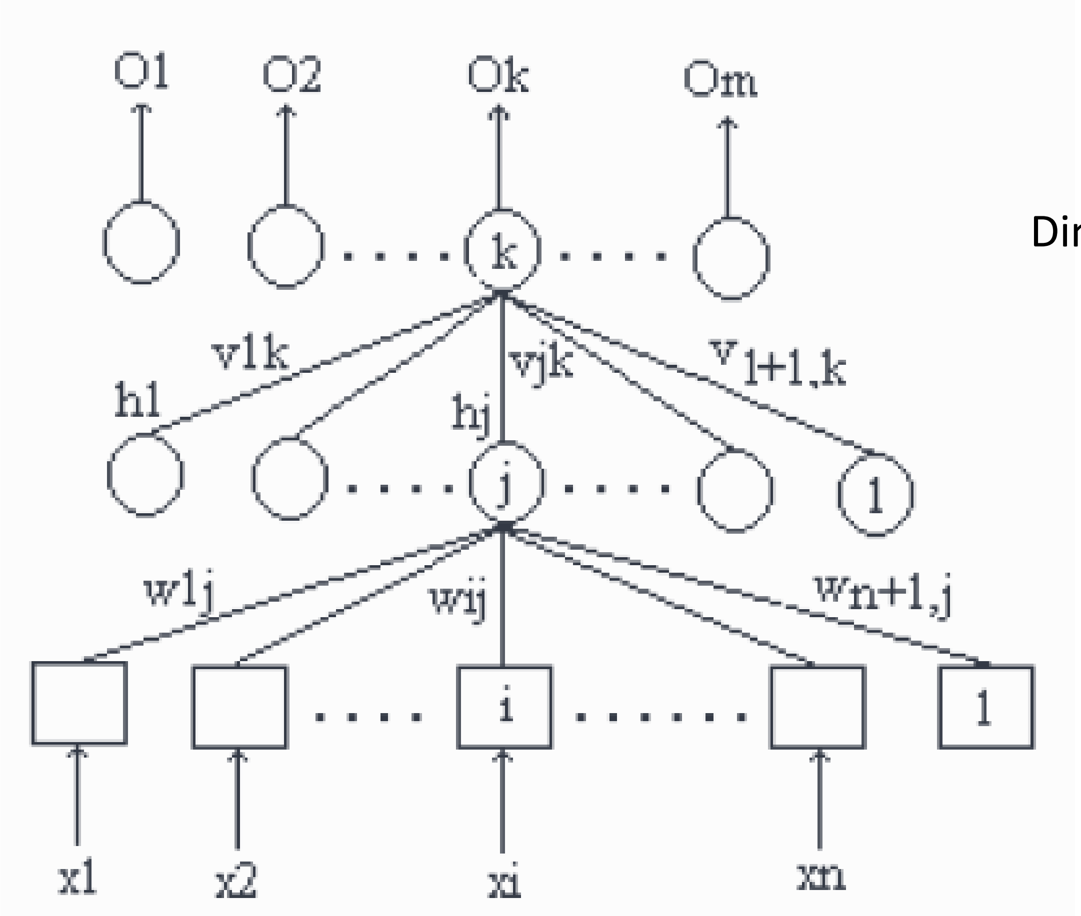
Realizar un programa del Perceptrón en Python, Matlab o cualquier lenguaje, considerando lo siguiente:

1. Utilizar como entrada la función OR y AND
2. Utilizar Wo = 0, 1
3. Utilizar $\eta = \{0.1, 0.2, \ldots 3\}$
4. Utilizar b = 1, -1
5. Entregar programa fuente y ejecutable en la Plataforma de Teams y adjuntar 1 pagina anotando conclusiones.

Retropropagación


salidas
Capa de salida
Capa escondida
Capa de entrada
entradas

Algoritmos de cálculo



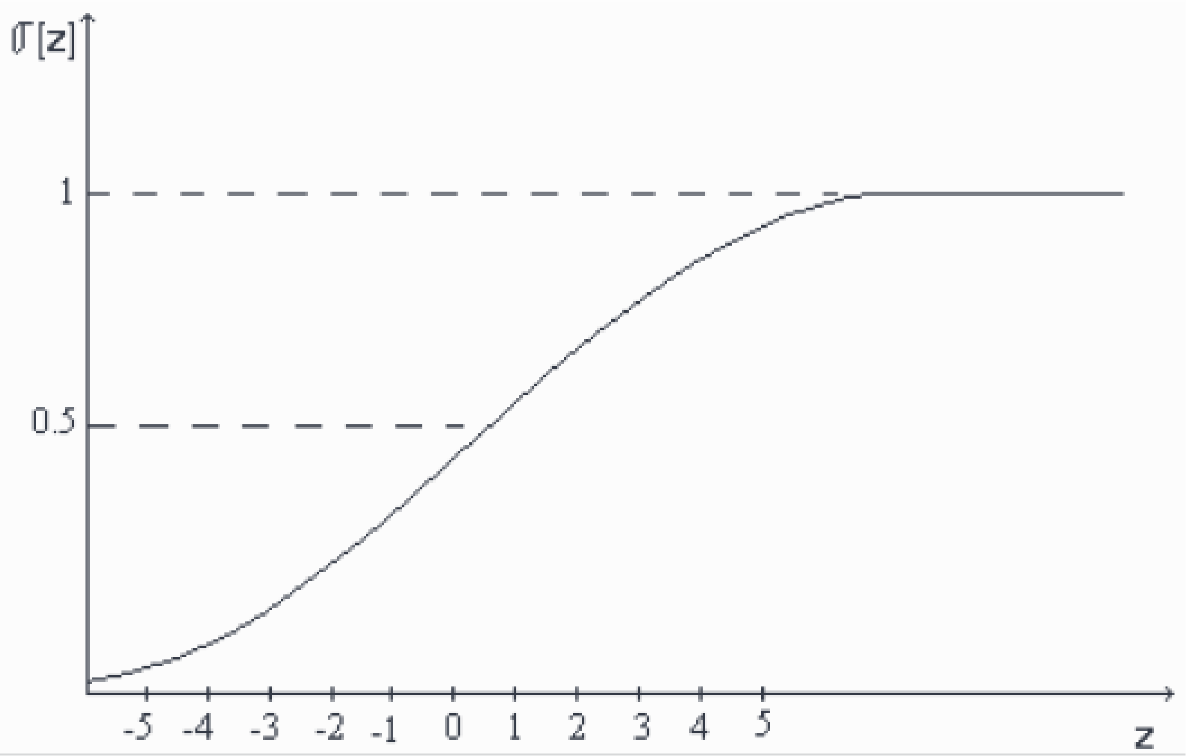
Dimension m

Dimension L

Dimension n

$$S_j^{\,p} = \sum_{i=1}^{n} w_{ij}^{\,p} x_i^{\,p} + w_{n+1,j}^{\,p} \qquad (1)$$

Función Sigma



$$h_j^{\,p} = f(S_j^{\,p}) = \frac{1}{1+\exp(-S_j^p)} \tag{2}$$

$$r_k^{\,p} = \sum_{j=1}^{1} v_{jk}^{\,p}\, h_j^{\,p} + v_{1+1,k}^{\,p} \tag{3}$$

$$O_k^{\,p} = f(r_k^{\,p}) = \frac{1}{1+\exp(-r_k^p)} \tag{4}$$

$$e_k^{\,p} = y_k^{\,p} - O_k^{\,p} \tag{5}$$

El criterio a minimizar:

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(e_k^{\,p})^2 = \frac{1}{2}\sum_{k=1}^{m}(y_k^{\,p} - O_k^{\,p})^2 \tag{6}$$

$$E_p = \frac{1}{2}\sum_{k=1}^{m}(y_k^{\,p} - f(r_k^{\,p}))^2 \tag{7}$$

Minimizar la función Ep con respecto a los pesos

$$\nabla E_p = \begin{bmatrix} \dfrac{\partial E_p}{\partial v_{jk}{}^p} \\ \dfrac{\partial E_p}{\partial w_{ij}{}^p} \end{bmatrix} \tag{8}$$

Steepest descent

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = \frac{\partial E_p}{\partial O_k{}^p}\frac{\partial O_k{}^p}{\partial r_k{}^p}\frac{\partial r_k{}^p}{\partial v_{jk}{}^p}$$

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = -\underbrace{(y_k{}^p - O_k{}^p)\,O_k{}^p\,(1 - O_k{}^p)}_{\delta_k{}^p}\,h_j{}^p$$

$$\frac{\partial E_p}{\partial v_{jk}{}^p} = -\delta_k{}^p\,h_j{}^p$$

$$\frac{\partial E_p}{\partial w_{ij}{}^p} = -\underbrace{\left(\sum_{k=1}^{m} \delta_k{}^p \, v_{jk}{}^p\right) h_j{}^p (1 - h_j{}^p)}_{\Delta_j{}^p} x_i{}^p$$

$$\frac{\partial E_p}{\partial w_{ij}{}^p} = -\Delta_j{}^p \, x_i{}^p$$

$$\nabla E_p = -\begin{bmatrix} \delta_k{}^p h_j{}^p \\ \Delta_j{}^p x_i{}^p \end{bmatrix} \Rightarrow \begin{bmatrix} v_{jk}{}^p \\ w_{ij}{}^p \end{bmatrix} = \begin{bmatrix} v_{jk}{}^{p-1} \\ w_{ij}{}^{p-1} \end{bmatrix} - \eta \nabla E_p$$

$$\begin{bmatrix} v_{jk}{}^p \\ w_{ij}{}^p \end{bmatrix} = \begin{bmatrix} v_{jk}{}^{p-1} \\ w_{ij}{}^{p-1} \end{bmatrix} + \eta \begin{bmatrix} \delta_k{}^p h_j{}^p \\ \Delta_j{}^p x_i{}^p \end{bmatrix}$$

$$v_{jk}{}^p = v_{jk}{}^{p-1} + \eta \delta_k{}^p h_j{}^p$$

$$w_{ij}{}^p = w_{ij}{}^{p-1} + \eta \Delta_j{}^p x_i{}^p$$