

Reconocimiento mediante redes neuronales

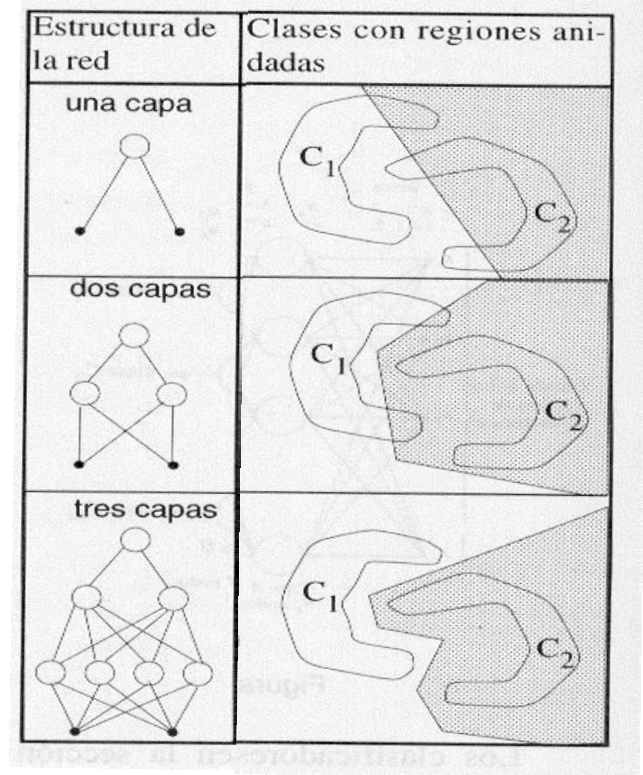
➤ Redes neuronales:

- ⇒ Basada en modelos matemáticos que tratan de imitar el modelo y funcionamiento de los sistemas biológicos: cómo las redes de neuronas almacenan y manipulan la información.
- ⇒ Están formadas por una serie de unidades de procesamiento muy simples (*neuronas*) que están conectadas entre sí y, cuyas salidas, dependen, además de la entrada, de las conexiones existentes entre las unidades de procesamiento.

⇒ Potente herramienta para el diseño de sistemas de reconocimiento de formas:

→ Capacidad de aprendizaje, facilidad de diseño, capacidad de manejo de problemas no-lineales:

❖ Capacidad de particionar el espacio de características mediante fronteras complejas lineales a trozos, a consta de añadir capas a la red.

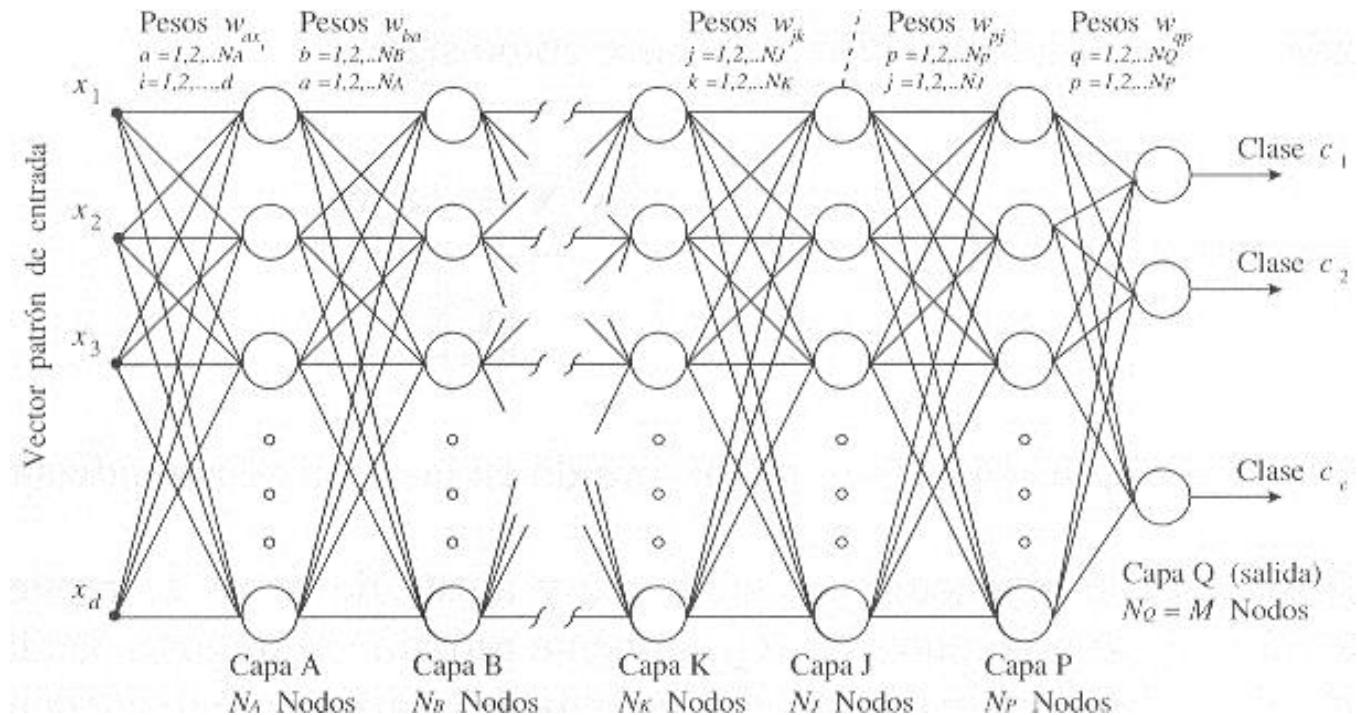


8.5.- Reconocimiento mediante redes neuronales

➤ Funcionamiento de una red neuronal: *arquitectura de la red*

- ⇒ Consta de capas de neuronas estructuradas de forma idéntica y colocadas de manera que la salida de cada neurona en una capa proporciona la entrada de cada neurona en la siguiente capa.
- ⇒ El número de neuronas en la primera capa (capa A) es igual a la dimensión del vector característica, mientras que en la capa de salida (capa Q) es igual al número de clases para las que la red se va a entrenar y sobre las que podrá clasificar un determinado vector de entrada X.
- ⇒ La red reconoce un vector X como perteneciente a la clase C_i si la salida i -ésima de la red es “alta”, mientras las otras salidas son” bajas”.

Modelo de la
red neuronal
multicapa



8.5.- Reconocimiento mediante redes neuronales

➤ Funcionamiento de una red neuronal: *estructura básica de una neurona en la red*

Suponiendo la capa K como la capa que precede a la J :

→ Entradas O_k de cada nodo en la capa J :

- $O_k \equiv$ salidas de los nodos en la capa K , $k = 1, 2, \dots, N_k$
- $N_k \equiv$ número de nodos en la capa K .

→ Entrada I_j del elemento de activación de cada nodo en la capa J :

$$I_j = \sum_{k=1}^{N_k} w_{jk} O_k \quad j=1, 2, \dots, N_j \quad (\text{suma promediada de las salidas que provienen de la capa previa}).$$

- $N_j \equiv$ número de nodos en la capa J
- $w_{jk} \equiv$ pesos que modifican las salidas O_k antes de que sean suministradas a los nodos en la capa J .

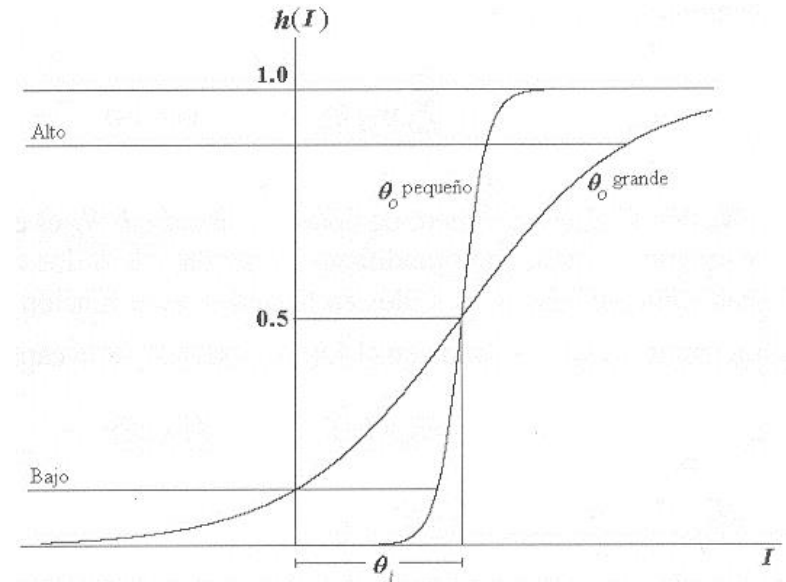
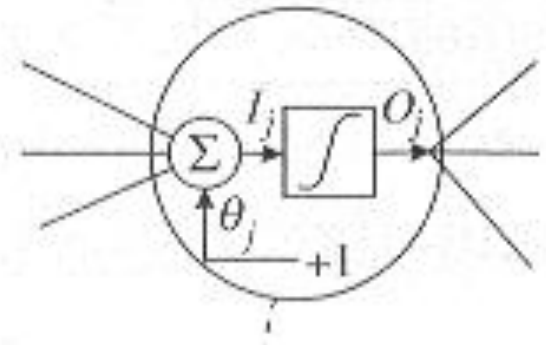
→ Salida del elemento de activación de la capa J : $O_j = h_j(I_j)$

$h_j \equiv$ función de activación

Función de activación sigmoideal:

$$O_j = h_j(I_j) = \frac{1}{1 + \exp\left[-(I_j + \theta_j)/\theta_0\right]}$$

- $\theta_j \equiv$ parámetro de ajuste
- $\theta_0 \equiv$ controla la forma de la función sigmoideal



➤ Funcionamiento de una red neuronal: *consideraciones*

⇒ Número total de coeficientes que se necesitan para especificar los nodos de la capa J :

→ Coeficientes de ajuste θ_j : se necesitan N_j , uno por cada nodo de la capa J .

→ Pesos de las salidas de la capa precedente K : se necesitan $N_j \times N_k$.

(Las N_k entradas al nodo i -ésimo en la capa J (tiene N_j nodos) son promediadas por los coeficientes w_{ik} , $k = 1, 2, \dots, N_k$)

⇒ Durante de entrenamiento:

→ La adaptación de las neuronas en la capa de salida es simple ya que se conoce la salida de cada nodo.

→ El principal problema estriba en el ajuste de los pesos en las capas intermedias (*capas ocultas*)

⇒ Algoritmo de entrenamiento más implementado: *entrenamiento por retropropagación* (“*backpropagation*”)

→ Durante el entrenamiento, la salida de la red se compara con la respuesta ideal y el error que exista se utiliza para corregir la red. El aprendizaje se produce como resultado de cambiar los pesos para reducir el error.

→ Es decir, comenzando por la capa de salida, se desarrolla una regla de entrenamiento que permita el ajuste de los pesos en cada una de las capas buscando minimizar una función de error de la forma:

$$E_Q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2 \quad (\text{error al cuadrado entre las respuestas deseadas, } r_q, \text{ y las respuestas actuales, } O_q, \\ \text{de los nodos en la capa } Q \text{ de salida}) \quad (N_Q \equiv \text{número de nodos en la capa de salida } Q)$$