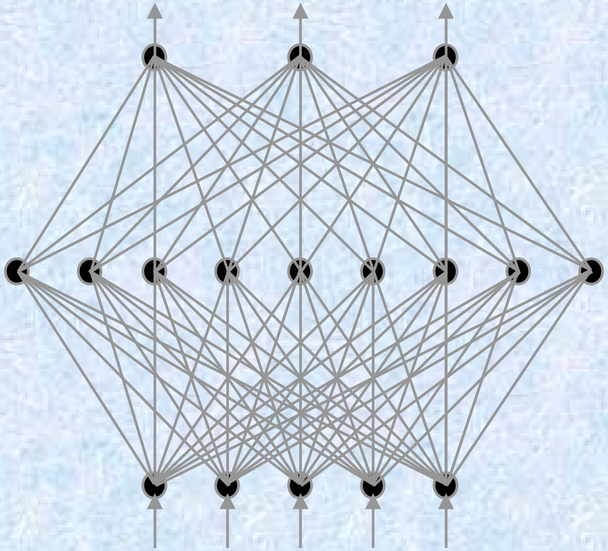
Redes Neuronales

Introducción



José Manuel Quero Reboul
Dpto. Ingeniería Electrónica
Universidad de Sevilla

Indice

- Motivación
- Arquitectura
- Leyes de Aprendizaje
- Aplicaciones

DILEMA

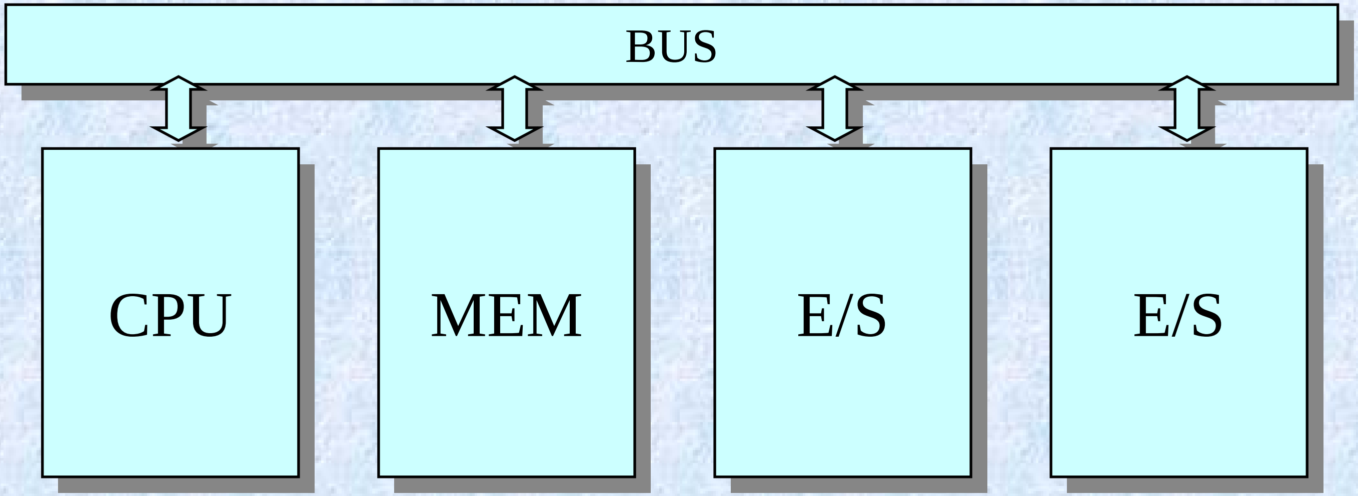
CALCULO

- Aritmética ➡ 1 cerebro=1/10 calculadora de bolsillo
- Visión ➡ 1 cerebro=1000 supercomputadores

MEMORIA

- Datos ➡ cerebro mucho peor
- Direccionada por contenido ➡ cerebro mucho mejor

Arquitectura Von Newman



Bus de datos: Cuello de Botella (Secuencialidad)

Cerebro

- 10^{10} neuronas (10ms)
- 10^4 dendritas
- 10^{14} pesos de conexión
- Los pesos almacenan y procesan

Definición

Computación neuronal:

Computación en redes masivas paralelas de procesadores simples y no lineales, que almacenan todo su contenido en los pesos de conexión

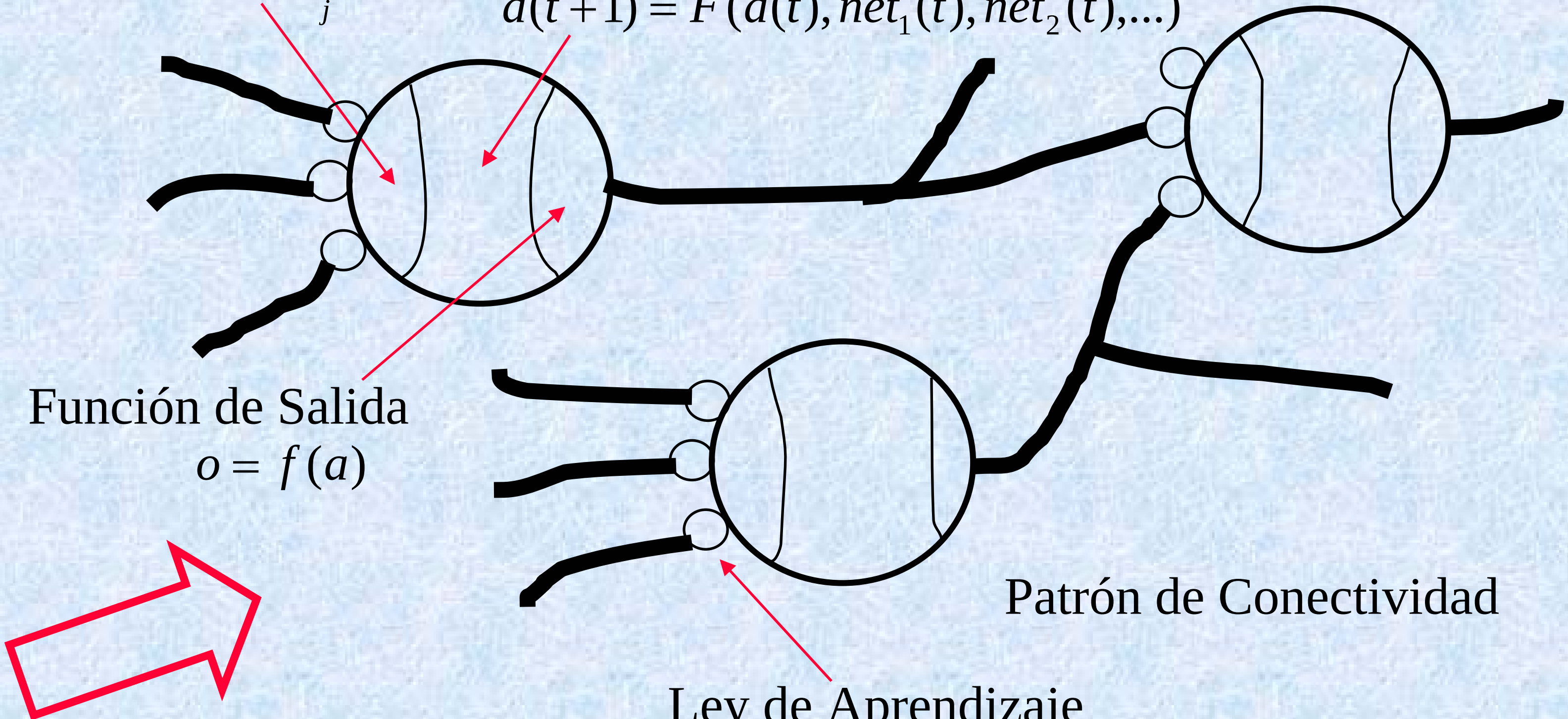
Propiedades:

- Eliminación del cuello de botella
- Inteligencia artificial llevada al límite

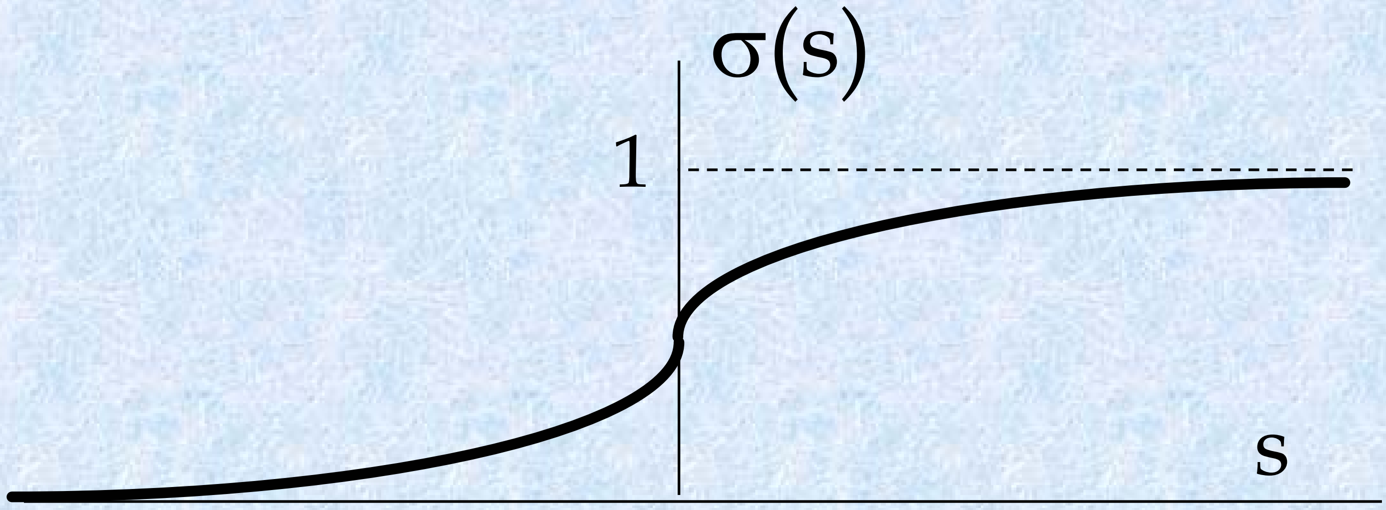
Indice

- Motivación
- Arquitectura
- Leyes de Aprendizaje
- Aplicaciones

Neurona Artificial

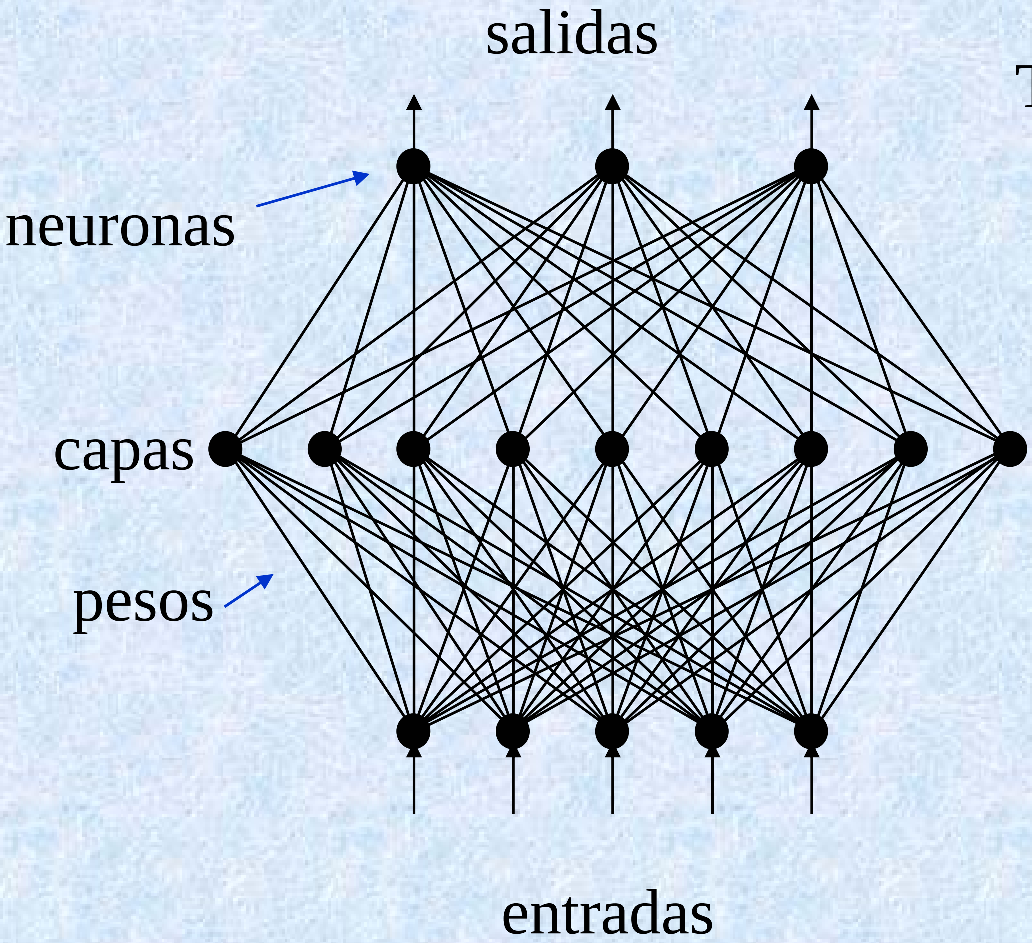


Función de activación



Función sigmoidal: $\sigma(x) = \dfrac{1}{1 + e^{-x}}$

Red neuronal



Topologías:

- Monocapa
- Multicapa
- Redes forward
- Redes recurrentes
- Redes realimentadas
- Redes de funciones radiales

Indice

- Motivación
- Arquitectura
- Clasificación
- Leyes de Aprendizaje
- Aplicaciones

Aprendizaje

- El conocimiento es obtenido a partir de la experiencia y almacenado en los pesos
- Tipos de aprendizajes:
  - Preprogramadas
  - Ley de Hebbs
  - Aprendizaje competitivo
  - Aprendizaje supervisado
- La red extrae y copia la estructura interna del conocimiento del entorno

Leyes de Aprendizaje

$$o = f(W^T I) \qquad \frac{dW}{dt} = \dot{W} = \phi(\cdot) I - \gamma(\cdot) W$$

W :Vector de pesos

I : Señales de entrada

$\phi(\cdot), \gamma(\cdot)$:Funciones escalares (W,I,o)

o :salida

Función de Tranferencia no lineal

Leyes de Aprendizaje

- Justificación

  1) Ley de Hebb: Cuanto mayor sea la excitación, mayor será el refuerzo de la conexión $\phi(\cdot)I$

  2) Factor de Olvido: Proporcional a la propia magnitud $-\gamma(\cdot)W$

- Condiciones (estabilidad dinámica)

  1) $\textbf{si } o(t) \textbf{ acotada} \Rightarrow W(t) \textbf{ finita } \forall t$

  2) $\textbf{si } I \neq 0 \Rightarrow W(t) \not\to 0 \quad t \to \infty$

Indice

- Motivación
- Arquitectura
- Clasificación
- Leyes de Aprendizaje
- <u>Aplicaciones</u>

Propiedades de las RN

- Procesamiento de un gran conjunto de datos
- Baja densidad de información
- Robustez ante fallo en estructura
- Robustez ante inconsistencia en los datos de entrada
- Datos y reglas de procesamiento confundidos en las conexiones
- Procesamiento altamente paralelo
- Capacidad de Autoorganización. Adaptabilidad

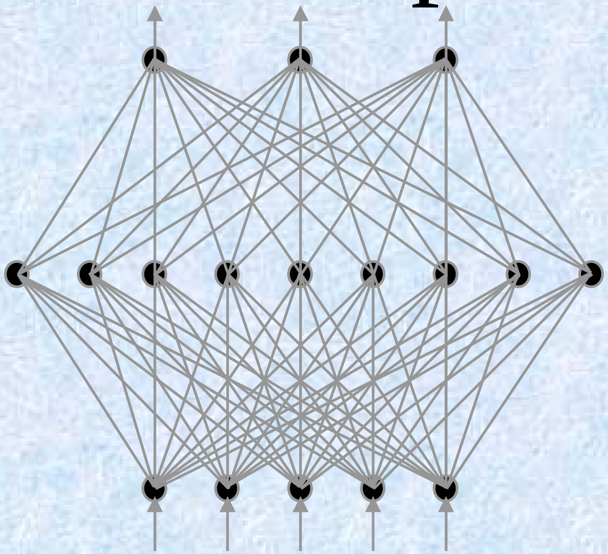
¿Cuando usar Redes Neuronales?

- Cuando se quiere desarrollar un modelo (funcional, clasificador, predicción de serie temporal,...)
- Ejemplos
  - Finanzas: Modelos de mercado
  - Ingeniería: Modelado de procesos y control adaptativo
  - Medicina: Diagnosis

¿Cuando usar Redes Neuronales?

- En análisis de datos con baja densidad de información
- Ejemplos:
  - Reconocimiento de imágenes
  - Reconocimiento de firmas
  - Análisis de encuestas
  - Predicción meteorológica

Redes Neuronales
Aprendizaje Supervisado

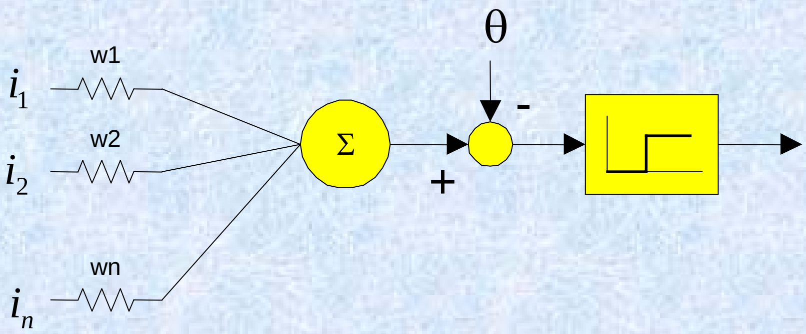


Indice

- Perceptrón
  - Regla delta
- Perceptrón Multicapa
  - Retropropagación
- Ejemplos

Perceptrón

- Memoria Asociativa

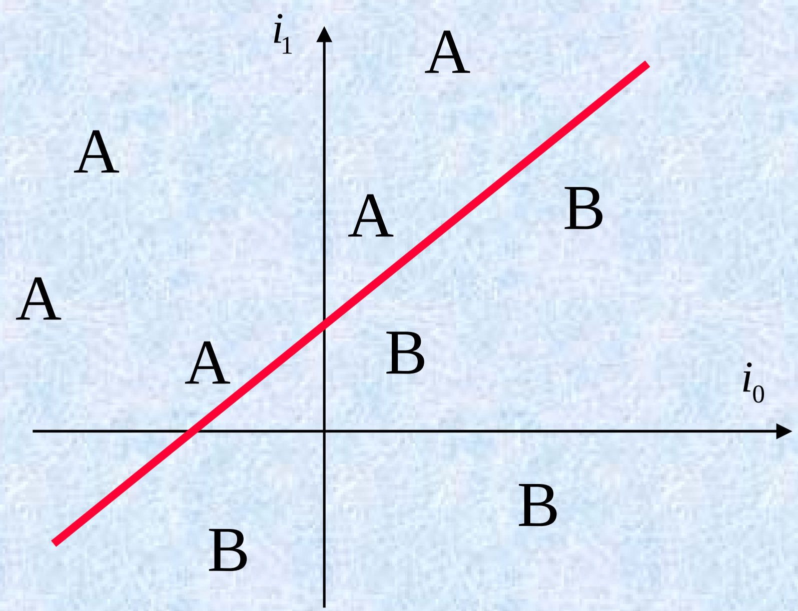


$$o = f_h\left(\sum_{j=1}^{n} w_j i_j - \theta\right)$$

Interpretación Geométrica

$$i_1 = -\frac{w_0}{w_1} i_0 + \frac{\theta}{w_1}$$

Recta de Decisión



Perceptrón

- Aprendizaje Supervisado: Regla δ

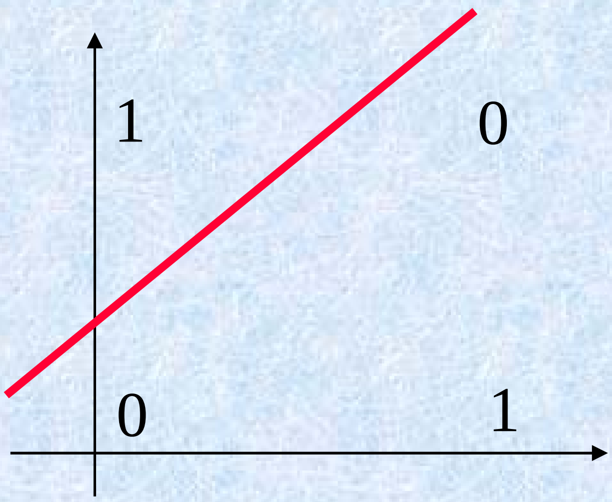
$$\frac{dw_i}{dt} = \alpha[d(t) - o(t)]i_i(t) \qquad 0 \le i \le n-1 \qquad d(t), o(t) \in \{-1,1\}$$

Aprendizaje a partir de w_i aleatorios

Problema: Oscilación ante entradas no separables

Ejemplo: función XOR

Patron de Entrada		Patron de Salida
00	$\longrightarrow$	0
01	$\longrightarrow$	1
10	$\longrightarrow$	1
11	$\longrightarrow$	0

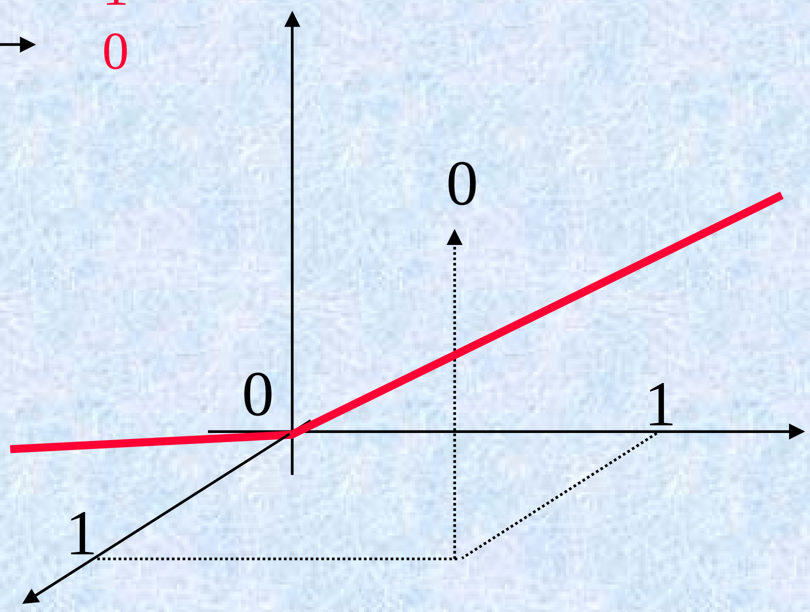


Perceptrón

Solución: Añadir una dimensión adicional

Patron de Entrada		Patron Intermedio		Patron de Salida
00	→	000	→	0
01	→	010	→	1
10	→	100	→	1
11	→	111	→	0

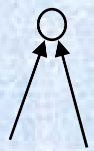
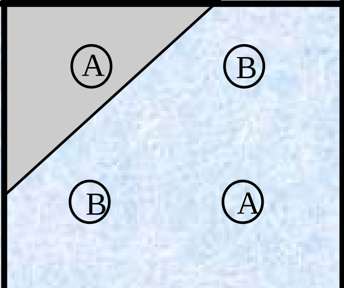
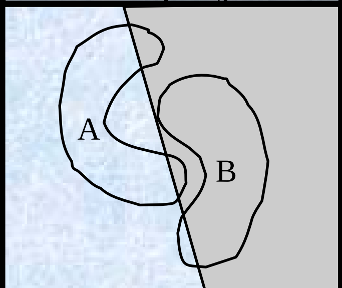
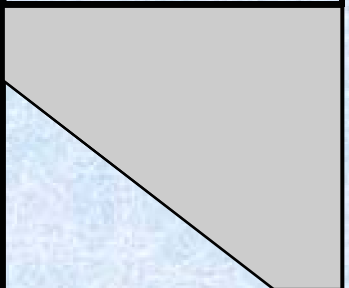
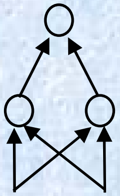
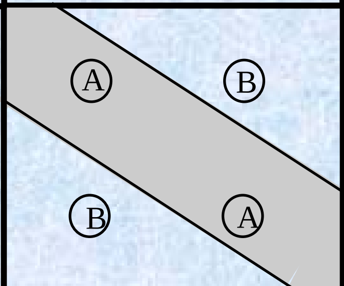
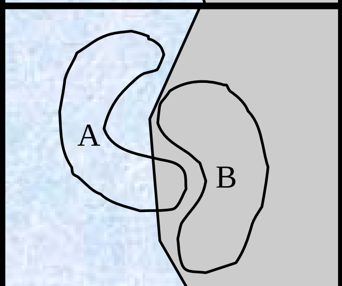
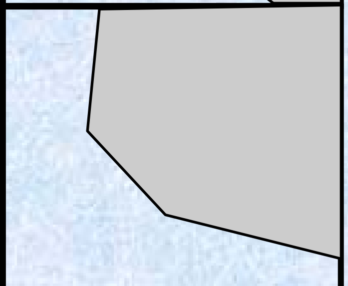
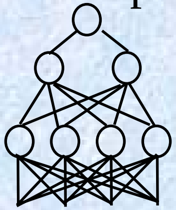
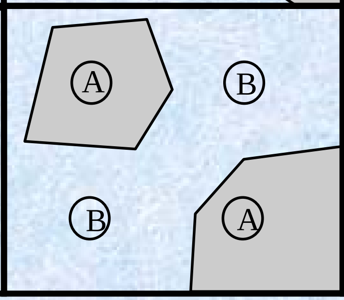
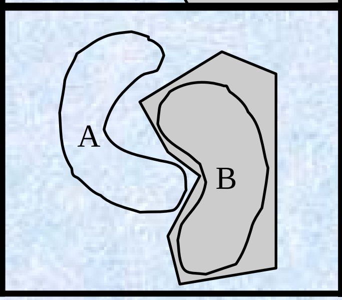
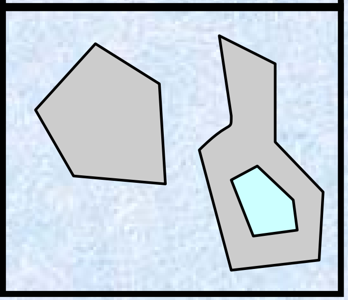
Nodos de representación interna



Indice

- Perceptrón
  - Regla delta
- Perceptrón Multicapa
  - Retropropagación
- Ejemplos

Perceptrón Multicapa

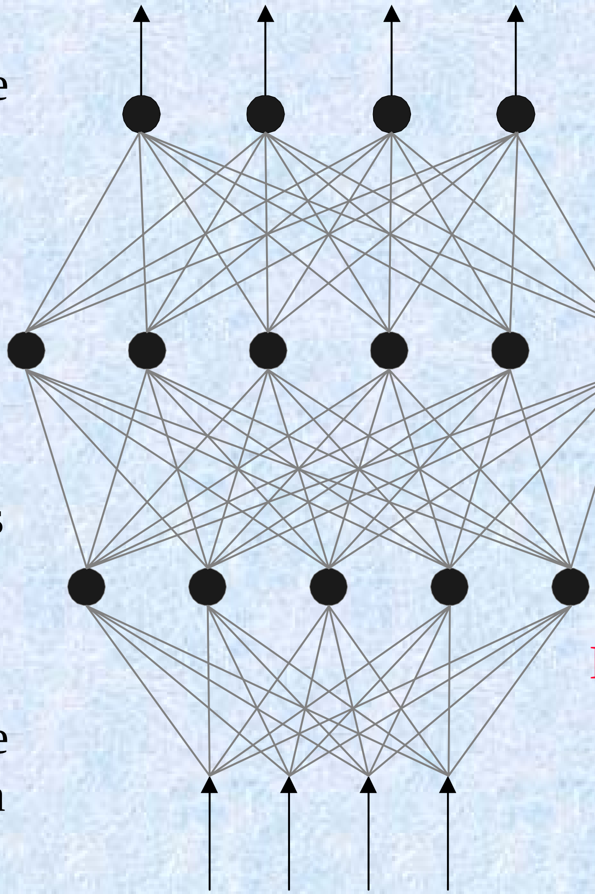
Estructura	Regiones de Decisión	Problema XOR	Clases Complejas	Regiones de Decisión Generales
Una capa 	Semiespacios limitados por hiperplanos			
Dos capas 	Regiones convexas abiertas o cerradas			
Tres capas 	Arbitrarias. Complejidad limitada por el número de nodos			

Perceptrón Multicapa

Capa de Salida

Capas Ocultas

Capa de Entrada



OR

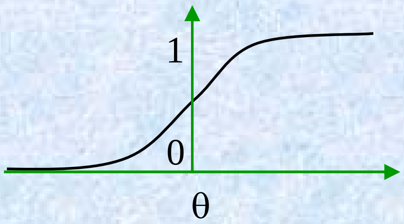
AND

HIPERPLANOS

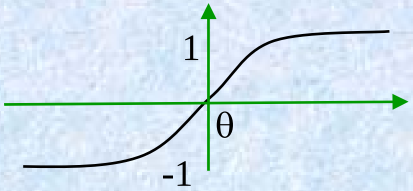
$$o_i = f(net_i - \theta) \qquad net_i = \sum_{j=1}^{n} w_{ij} i_j$$

$$o_i = f(net_i)$$

$$f(x) = \frac{1}{1+e^{-x}}$$



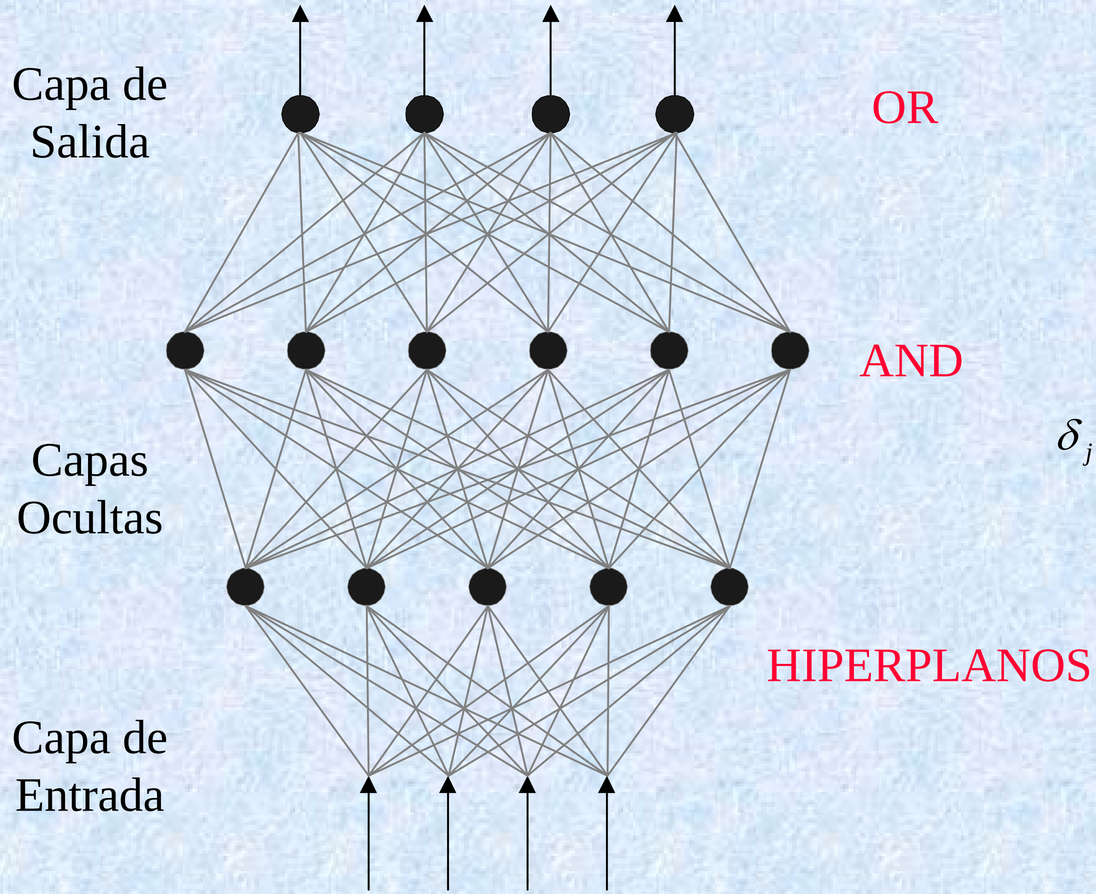
$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}$$



Indice

- Perceptrón
  - Regla delta
- Perceptrón Multicapa
  - Retropropagación
- Ejemplos

Perceptrón Multicapa



Retropropagación

(Regla δ generalizada)

$$\Delta_p w_{ij}(t+1) = \eta \delta_j i_i \qquad \text{capa i previa a j}$$

$$\delta_j = \begin{cases} f'_j(net_{pj})(d_j - o_j) & \text{Capa de Salida} \\ f'_j(net_{pj}) \sum_k \delta_k w_{ij} & \text{Capas Ocultas} \end{cases}$$

Perceptrón Multicapa

Retropropagación: Demostración

Error Cuadrático Médio $$E \equiv \sum_p E_p = \frac{1}{2}\sum_p \sum_j (d_{pj} - o_{pj})^2$$

Regla de la cadena $$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} \quad (1)$$

Cambio de la excitación la variar el peso de conexión

Cambio del error al variar la excitación en j

Segundo Término

$$\frac{\partial net_{pj}}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} (\sum_i w_{ji} o_{pi}) = o_{pi} \quad (2)$$

Primer Término

Definamos $\delta_{pj} \equiv -\dfrac{\partial E_p}{\partial net_{pj}}$ (3) Sustituyendo (2) y (3)en (1) $\Rightarrow$ $\boxed{\Delta_p w_{ji} = \eta \delta_{pj} o_{pj}}$

Queda determinar qué es δ_{pj} para cada neurona

Perceptrón Multicapa

Regla de la cadena $\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pi}}$

Cambio de la salida por cambiar la excitación

Cambio del error al variar la salida

Segundo Término

Dado que $o_{pj} = f(net_{pj}) \quad \Rightarrow \frac{\partial o_{pj}}{\partial net_{pj}} = f'(net_{pj})$ Derivada de la función de salida

Primer Término

Para neurona de salida. Dado que $E_p = \frac{1}{2}\sum_j (d_{pj} - o_{pj})^2 \Rightarrow \frac{\partial E_p}{\partial o_{pj}} = -(d_{pj} - o_{pj})$

$$\Rightarrow \boxed{\delta_{pj} = f'(net_{pj})\left(d_{pj} - o_{pj}\right)}$$

Perceptrón Multicapa

Para neurona oculta.

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial net_{pk}}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial net_{pk}} \frac{\partial}{\partial o_{pj}} \left(\sum_i w_{ki} o_{pi} \right) = \sum_k \frac{\partial E_p}{\partial net_{pk}} w_{kj} = -\sum_k \delta_{pk} w_{kj}$$

Cómo afecta a la capa previa

δ de la capa previa

Conexión con la neurona previa

$$\Rightarrow \boxed{\delta_{pj} = f'(net_{pj}) \sum_k \delta_{pk} w_{kj}}$$

Particularizando $o_{pj} = \frac{1}{1+e^{-net_j}} \Rightarrow o'_{pj} = f'(net_j) = \frac{e^{-net_j}}{\left(1+e^{-net_j}\right)^2} = o_{pj}(1-o_{pj})$

$$\Rightarrow \delta_j = \begin{cases} o_j(1-o_j)(d_j - o_j) & \text{Capa de Salida} \\ i_j(1-i_j) \sum_k \delta_k w_{ij} & \text{Capas Ocultas} \end{cases}$$

Indice

- Perceptrón
  - Regla delta
- Perceptrón Multicapa
  - Retropropagación
- <u>Ejemplos</u>