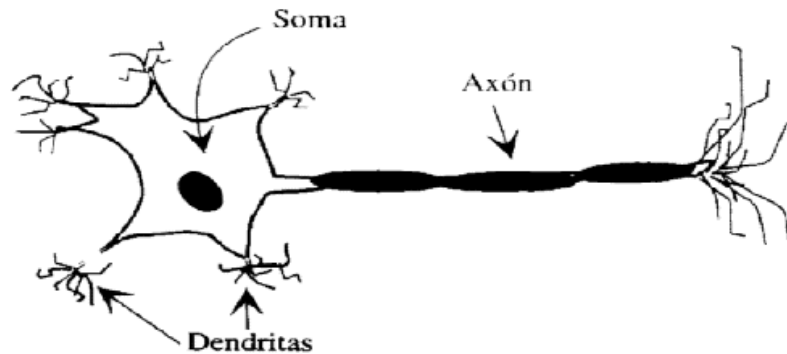


REDES NEURONALES

REDES NEURONALES

- Neurona: base del funcionamiento del cerebro.
- Sistema de procesamiento cerebral de la información:
 - Complejo, No lineal y Paralelo.

Fisiología de una neurona elemental



- Elementos de que consta: sinapsis, axón, dendritas y soma o cuerpo

NEURONA ARTIFICIAL

- Neurona artificial: unidad de procesamiento de la información, es un dispositivo simple de cálculo que ante un vector de entradas proporciona una única salida.

- Elementos:

- Conjunto de entradas, x_j

- Pesos sinápticos, w_i

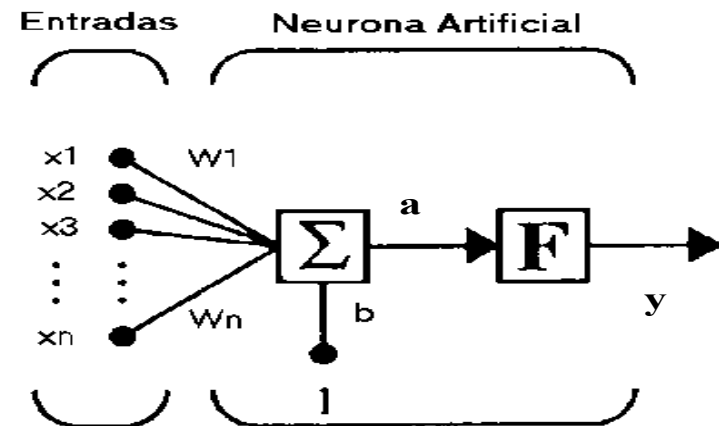
- Función de activación:

$$w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = a$$

- Función de transferencia:

$$y = F(w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n)$$

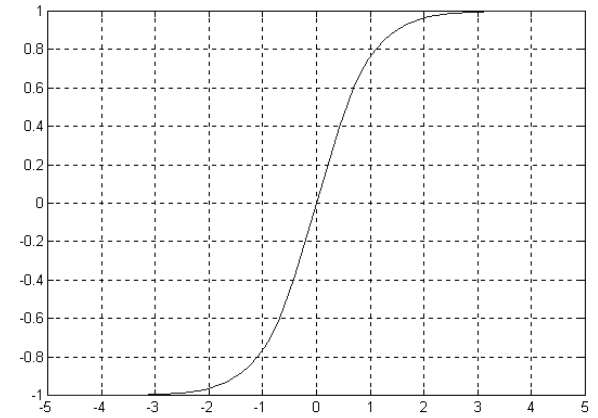
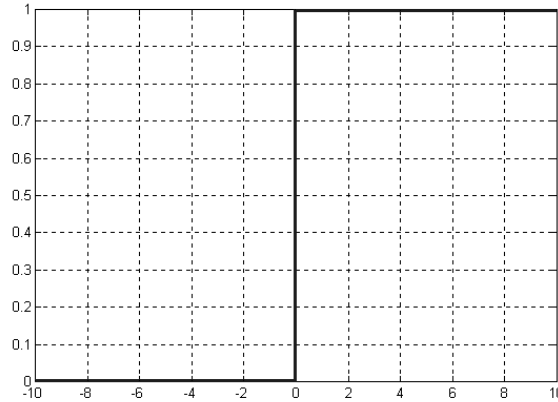
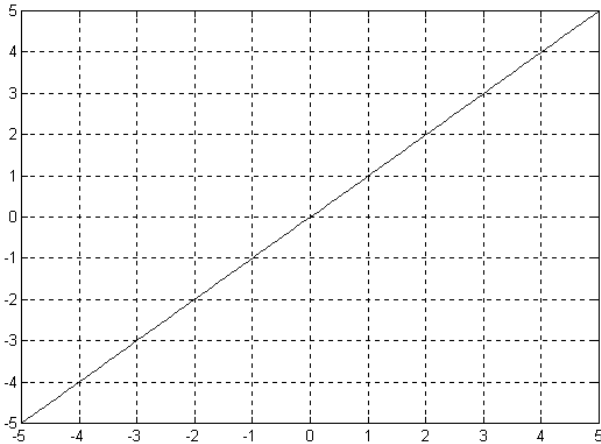
- Bias o polarización: entrada constante de magnitud 1, y peso b que se introduce en el sumador



NEURONA ARTIFICIAL

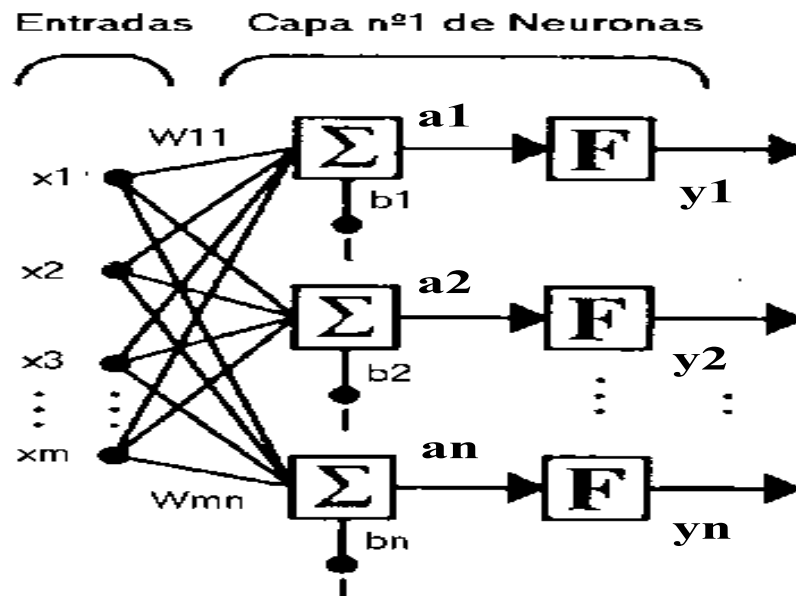
■ Principales funciones de transferencia:

- Lineal: $y=ka$
- Escalón: $y = 0$ si $a < 0$; $y=1$ si $a \geq 0$
- Sigmoide
- Gaussiana.



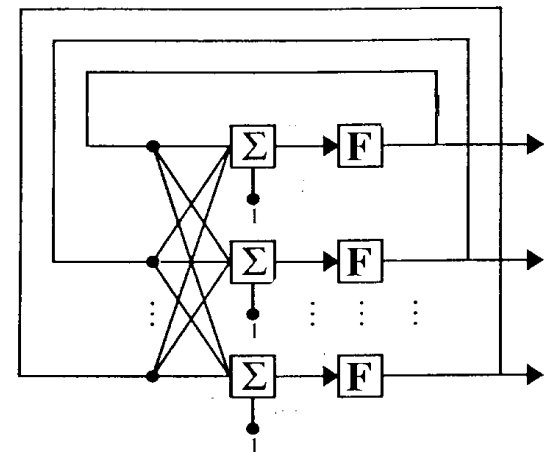
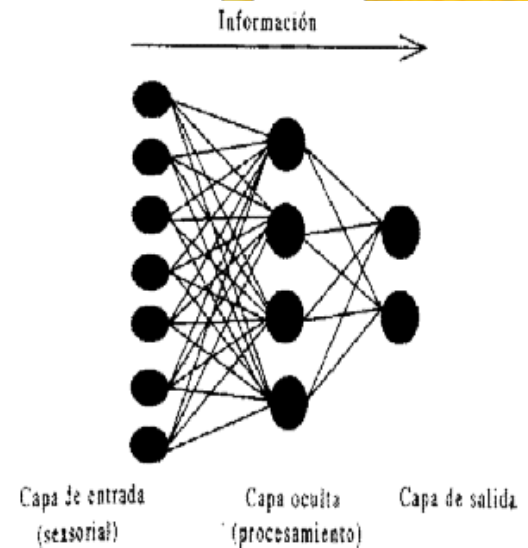
RNA de una capa

- Una neurona aislada dispone de poca potencia de cálculo.
- Los nodos se conectan mediante la sinapsis
- Las neuronas se agrupan formando una estructura llamada capa.
- Los pesos pasan a ser matrices W ($n \times m$)
- La salida de la red es un vector: $Y = (y_1, y_2, \dots, y_n)^T$
- $Y = F(W \cdot X + b)$



RNA Multicapa

- Redes multicapa: capas en cascada.
- Tipos de capas:
 - Entrada
 - Salida
 - Oculta
- No hay realimentación => *red feedforward*
 - Salida depende de entradas y pesos.
- Si hay realimentación => *red recurrente*
 - Efecto memoria
 - Salida depende también de la historia pasada.
- Una RNA es un aproximador general de funciones no lineales.

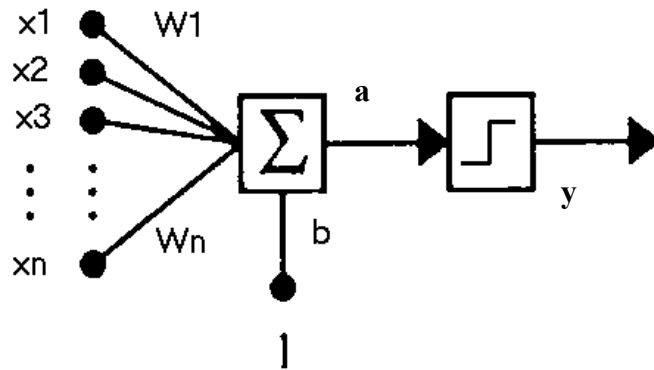


Entrenamiento I

- Entrenamiento: proceso de aprendizaje de la red.
- Objetivo: tener un comportamiento deseado.
- Método:
 - Uso de un algoritmo para el ajuste de los parámetros libres de la red: los pesos y las bias.
 - Convergencia: salidas de la red = salidas deseadas.
- Tipos de entrenamiento:
 - Supervisado.
 - | Pares de entrenamiento: entrada - salida deseada.
 - | Error por cada par que se utiliza para ajustar parámetros
 - No-supervisado.
 - | Solamente conjunto de entradas.
 - | Salidas: la agrupación o clasificación por clases
 - Reforzado.

Perceptrones

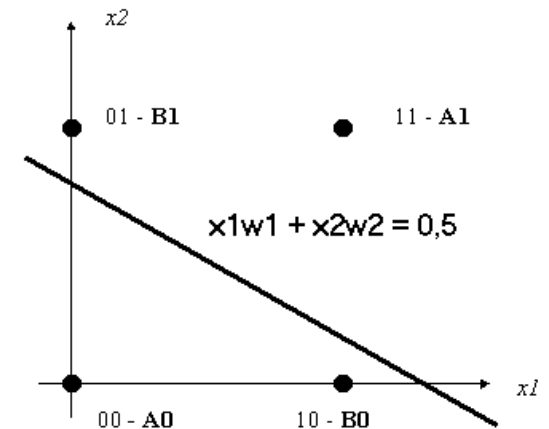
- McCulloch y Pitts, en 1943, publicaron el primer estudio sobre RNA.
- El elemento central: perceptrón.



- Solo permite discriminar entre dos clases linealmente separables: XOR.

- $0.5 = a = w_1 x_1 + w_2 x_2$
- No hay combinación de x_1 y x_2 que resuelva este problema.

- Solución: más capas o funciones de transferencia no lineales.



Aprendizaje del Perceptrón.

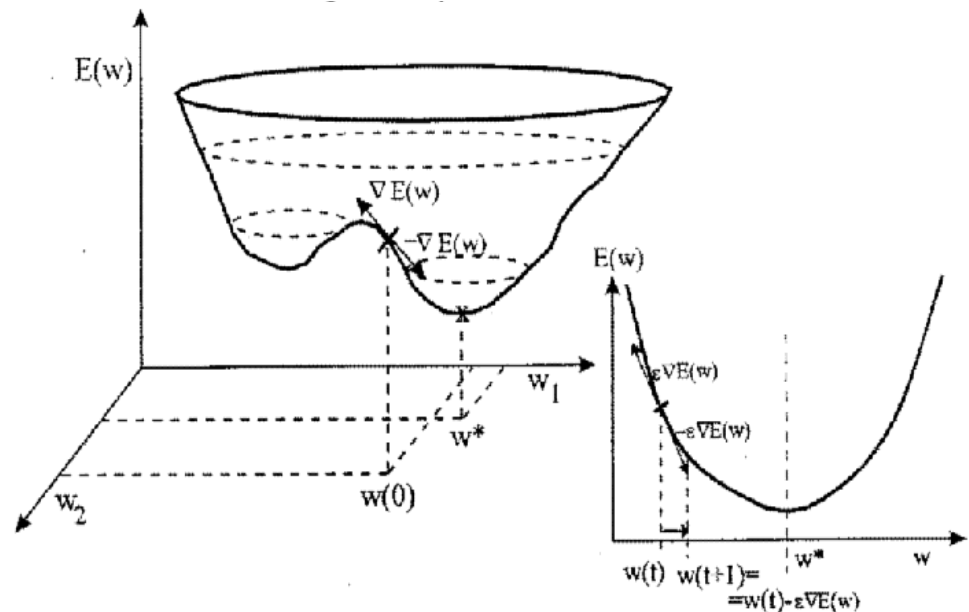
- Algoritmo supervisado:
 - ① Aplicar patrón de entrada y calcular salida de la red
 - ② Si salida correcta, volver a 1
 - ③ Si salida incorrecta
 - | 0 → sumar a cada peso su entrada
 - | 1 → restar a cada peso su entrada
 - ④ Volver a 1
- Proceso iterativo, si el problema es linealmente separable este algoritmo converge en un tiempo finito.
- Nos da los pesos y las bias de la red que resuelve el problema.

Regla delta

- Generalización del algoritmo del perceptrón para sistemas con entradas y salidas continuas.
- Se define: $\delta = \mathbf{T} - \mathbf{A} = e_k(n)$ (salidas deseadas - salidas de la red).
- Minimiza una función de coste basada en ese vector de error:

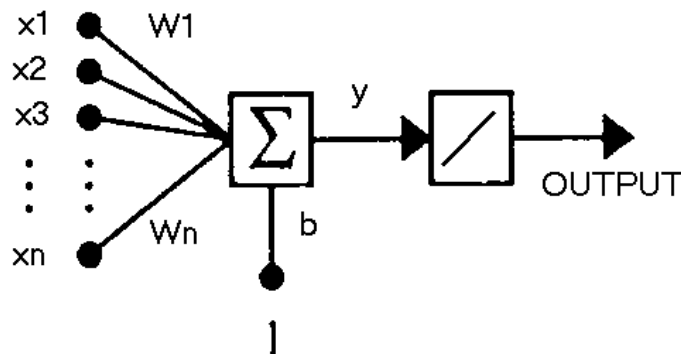
$$J = E \left[\frac{1}{2} \sum_k e_k^2(n) \right]$$

- $\Delta_i = \delta \cdot l_r \cdot x_i$
- $W_i(n+1) = W_i(n) + \Delta_i$
- Razón de aprendizaje l_r
- Si las neuronas son lineales => **un único mínimo**



Redes Neuronales Lineales.

- Función de transferencia lineal.
- Algoritmo de entrenamiento de Widrow-Hoff o Delta, tiene en cuenta la magnitud del error.
- Entrenamiento:
 - Suma de los cuadrados de los errores sea mínima.
 - Superficie de error con mínimo único.
 - Algoritmo tipo gradiente.



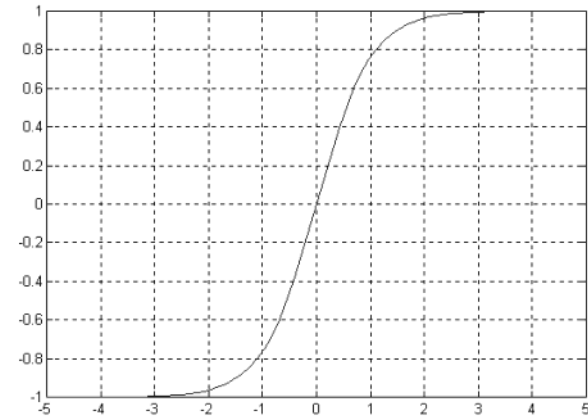
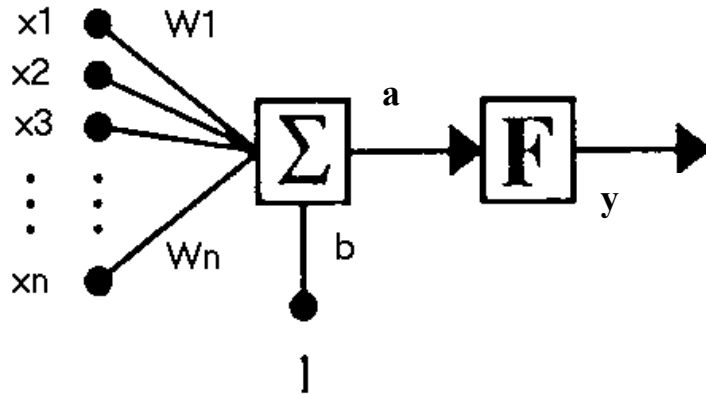
- Aproximan funciones lineales.

Backpropagation

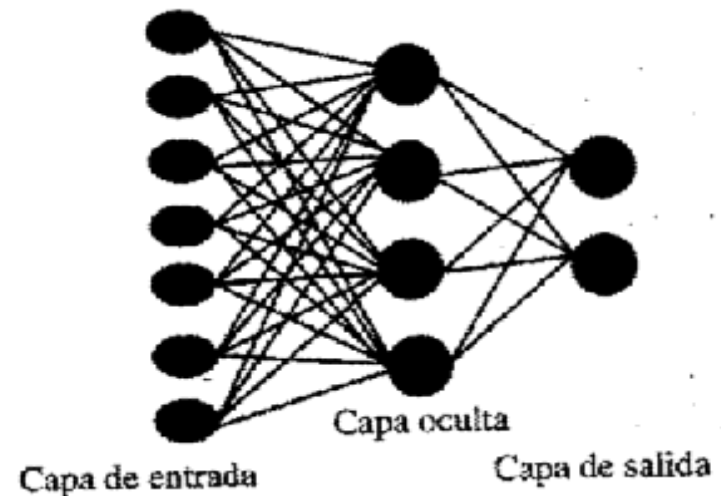


- Clave en el resurgimiento de las redes neuronales.
- Primera descripción del algoritmo fue dada por Werbos en 1974
- Generalización del algoritmo de Widrow-Hoff para redes multicapa con funciones de transferencia no-lineales y diferenciables.
- 1989 Hornik, Stinchcombe y White
 - Una red neuronal con una capa de sigmoides es capaz de aproximar cualquier función con un número finito de discontinuidades
- Propiedad de la generalización.
- La función de transferencia es no-lineal, la superficie de error tiene varios mínimos locales.

Red Perceptron Multicapa (MLP)



- Función acotada, monótona creciente y diferenciable.
- Red de tipo feedforward.
- Suficiente con dos capas.



Algoritmo backpropagation I

■ Descripción:

Adelante

- Tras inicializar los pesos de forma aleatoria y con valores pequeños, seleccionamos el primer par de entrenamiento.
- Calculamos la salida de la red
- Calculamos la diferencia entre la salida real de la red y la salida deseada, con lo que obtenemos el vector de error

Atrás

- Ajustamos los pesos de la red de forma que se minimice el error
- Repetimos los tres pasos anteriores para cada par de entrenamiento hasta que el error para todos los conjuntos de entrenamiento sea aceptable.

■ Descenso por la superficie del error

■ Cálculo de derivadas del error respecto de los pesos y de las bias.

Algoritmo backpropagation II

■ Detalles:

- SSE: $E = \sum E_p = \sum (y_{pk} - o_{pk})^2$
- $\Delta W_{ij} = -\eta \partial E / \partial W_{ij}$

■ Pasos:

- Inicialización:
 - | Construcción de la red.
 - | Inicialización aleatoria de pesos y umbrales (-0.5, 0.5)
 - | Criterio de terminación (número máximo de iteraciones,...).
 - | Contador de iteraciones $n=0$.
- Fase hacia delante:
 - | Calcular la salida de la red para cada patrón de entrada.
 - | Calcular el error total cometido (SSE)
 - | Si la condición de terminación se satisface, parar
- Fase hacia atrás:

Algoritmo backpropagation III

- Fase hacia atrás:

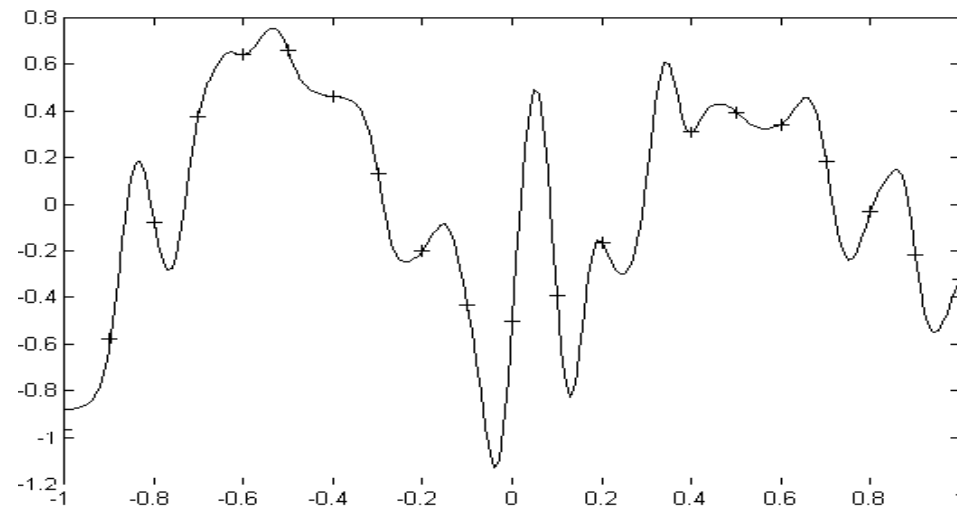
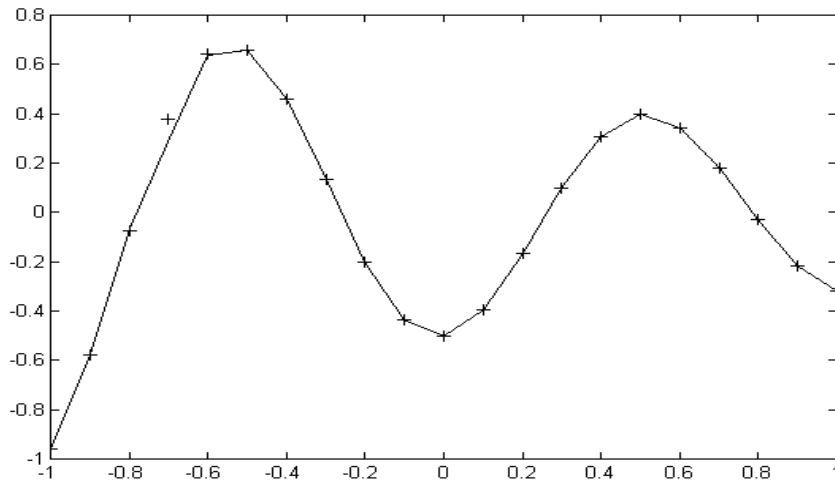
- | Incrementar el contador $n=n+1$.
- | Para cada neurona de salida calcular: $\delta_k = (o_k - y_k)f'(net_k)$
donde $net_j = \sum_i w_{ij}x_i + b_j$
- | Para cada unidad oculta, calcular $\delta_j = f'(net_j) \sum_k \delta_k w_{jk}$
- | Actualizar pesos: $\Delta w_{ij}(n+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(n)$
- | Volver a la fase hacia delante.

- Inconvenientes del algoritmo backpropagation:

- Tiempo de entrenamiento no acotado.
- Dependiente de las condiciones iniciales:
 - | Parálisis de la red.
 - | Mínimos locales.

Algoritmo Backpropagation IV

- Underfitting.
- Memorización o Sobreaprendizaje.
- Caracterización de la red. ¿Cuántas capas, cuántas neuronas en cada capa,...?



Redes Neuronales no supervisadas I

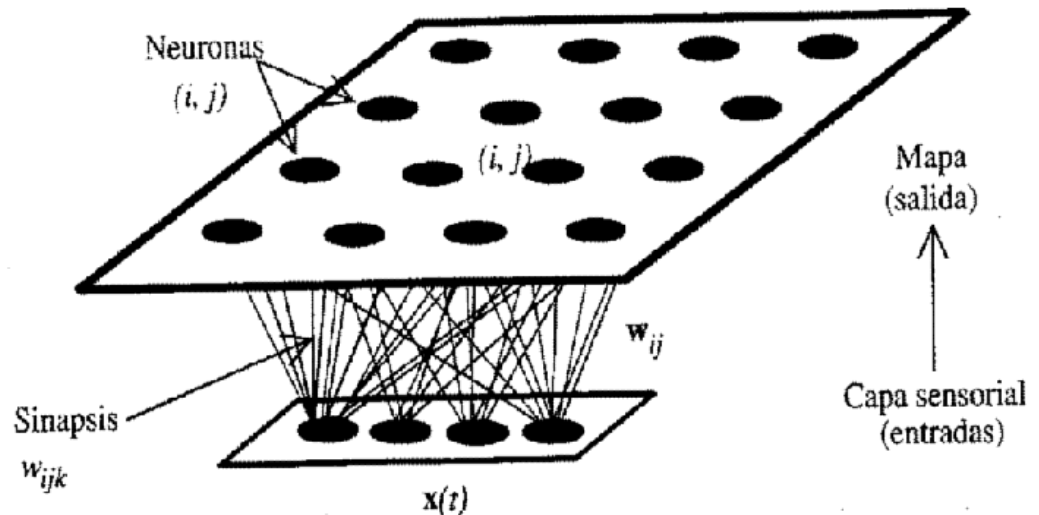
- Autoorganizativas: durante el proceso de aprendizaje la red debe descubrir por si misma regularidades o categorías => la red debe autoorganizarse en función de las señales procedentes del entorno.
- Mapa de Rasgos Autoorganizados, SOM (Kohonen, 80)
- Características:
 - Red competitiva
 - Arquitectura unidireccional de dos capas:
 - | Capa de entrada: m neuronas una por cada vector de entrada.
 - | Capa segunda se realiza el procesamiento, formando el mapa de rasgos. Tiene $n_x \times n_y$ neuronas operando en paralelo.
 - | Todas las neuronas de entrada están conectadas a las neuronas de la segunda capa, a través de los pesos w_{ij}

Redes Neuronales No-Supervisadas II

- Cada neurona (i,j) calcula la similitud entre el vector de entradas y su vector de pesos
- Vence la neurona cuyo vector de pesos es más similar al vector de entrada.

$$d(w_g, x) = \min_{ij} d(w_{ij}, x)$$

- Cada neurona sirva para detectar alguna característica del vector de entrada.
- Función de vecindad: relación entre neuronas próximas en el mapa.



RNA no supervisadas III

■ Aprendizaje:

- Inicialización de los pesos w_{ij}
- Presentación de las entradas $x(t)$
- Cada neurona calcula, la similitud entre su vector de pesos w_{ij} y el vector de entrada x , usando la distancia Euclídea

$$d^2(w_{ij}, x) = \sum_{k=1}^n (w_{ijk} - x_k)^2$$

- Determinación de la neurona ganadora: $Ganadora = \min_j d_j^2$
- Actualización de los pesos de la neurona ganadora y sus vecinas

$$w_{ijk}(t+1) = w_{ijk}(t)\alpha(t)h(|i-g|, t)(x_k(t) - w_{ijk}(t))$$

- Las demás neuronas no actualizan su peso
- Si se ha alcanzado el número de iteraciones parar, si no volver al paso 2.

VENTAJAS



■ Ventajas de las RNA:

- **Aprendizaje adaptativo**: lo necesario es aplicar un buen algoritmo y disponer de patrones (pares) de entrenamiento.
- **Auto-organización** => conduce a la generalización
- **Tolerancia a fallos**: las redes pueden aprender patrones que contienen ruido, distorsión o que están incompletos.
- **Operación en tiempo real**: procesan gran cantidad de datos en poco tiempo.
- Facilidad de inserción en tecnología ya existente.