Minería de datos aplicada

Introducción: Redes Neuronales

Intro

Una red neuronal es un procesador distribuido masivamente paralelo formado por unidad de proceso simple, que tiene propensión natural para el almacenamiento de conocimiento de la experiencia y ponerla a disposición para su uso.

Se asemeja al cerebro en dos aspectos:

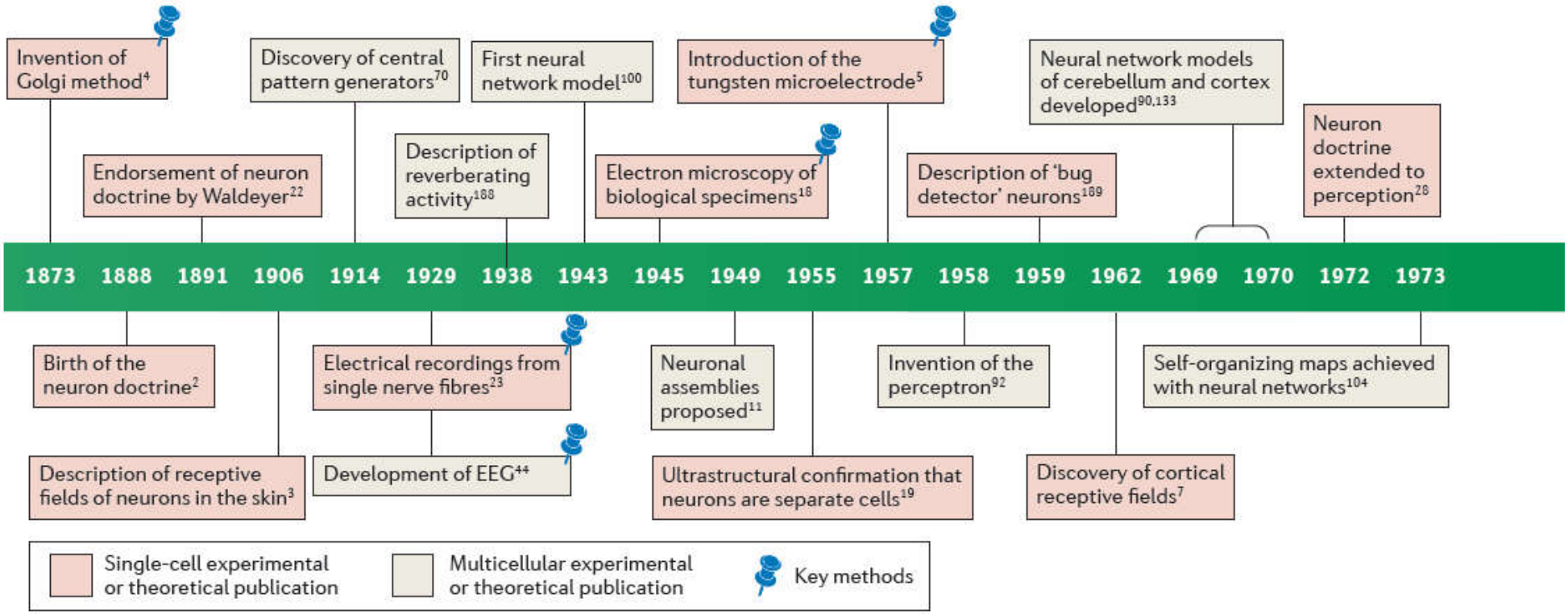
- El conocimiento se adquiere por la red de su entorno a través de un proceso de aprendizaje.
- las fuerzas de conexión interneuronas, conocidos como los pesos sinápticos, se utilizan para almacenar los conocimientos adquiridos.

Intro

1. conectividad masiva
2. No lineal, paralelo, robusto y tolerante a errores
3. La capacidad de adaptarse a un entorno
4. Capacidad de aprender y generalizar a partir de ejemplos conocidos
5. El comportamiento colectivo es diferente del comportamiento individual

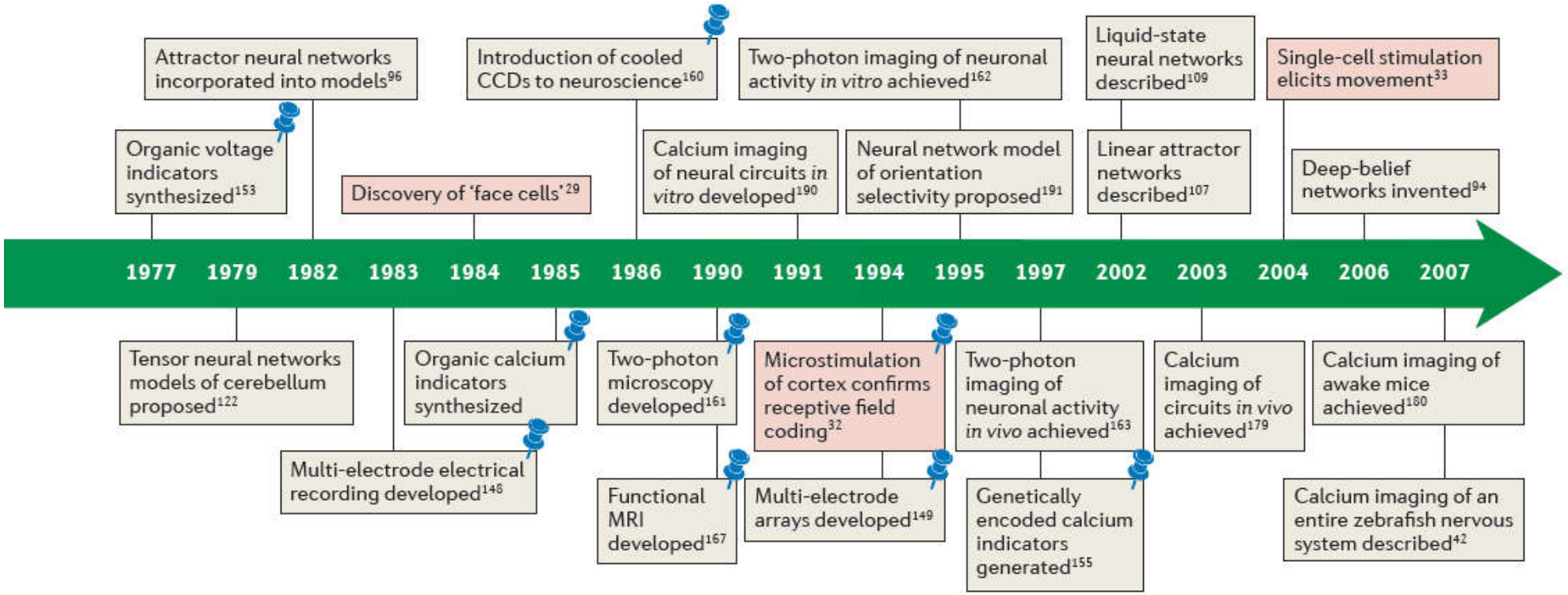
Redes Neuronales Artificiales imita algunas de las propiedades de las redes neuronales biológicas

Desarrollo Histórico



Nature Reviews Neurosciens **16**,487–497(2015)

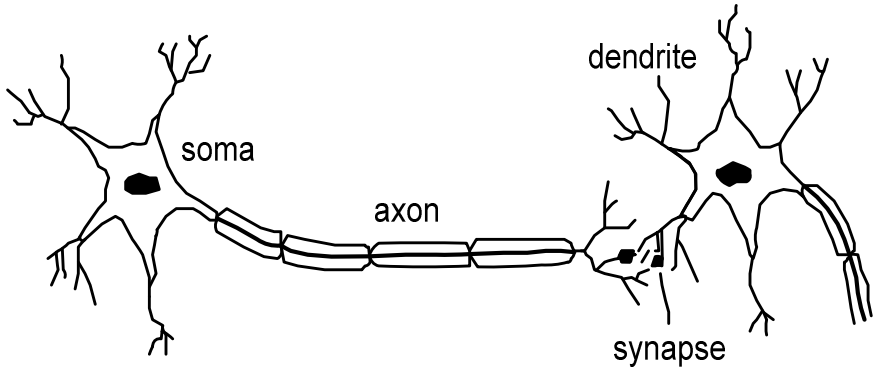
Desarrollo Histórico



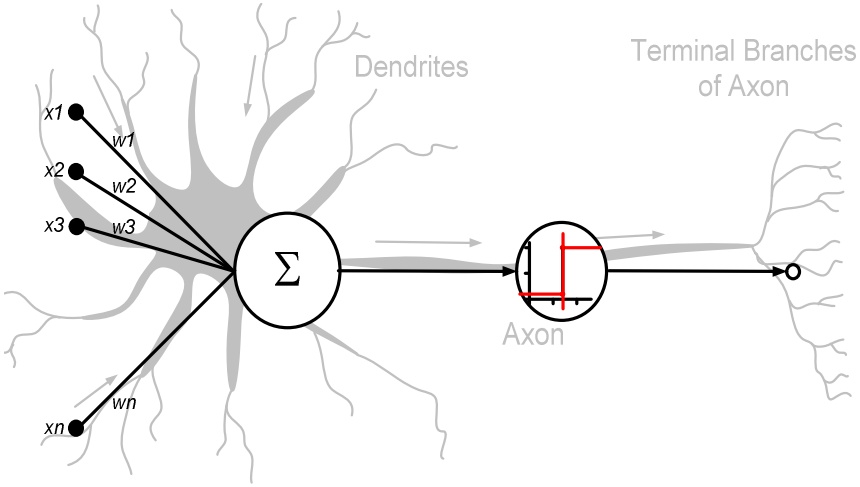
Nature Reviews Neurosciens **16**,487–497(2015)

Redes Neuronales

Neurona Biologica

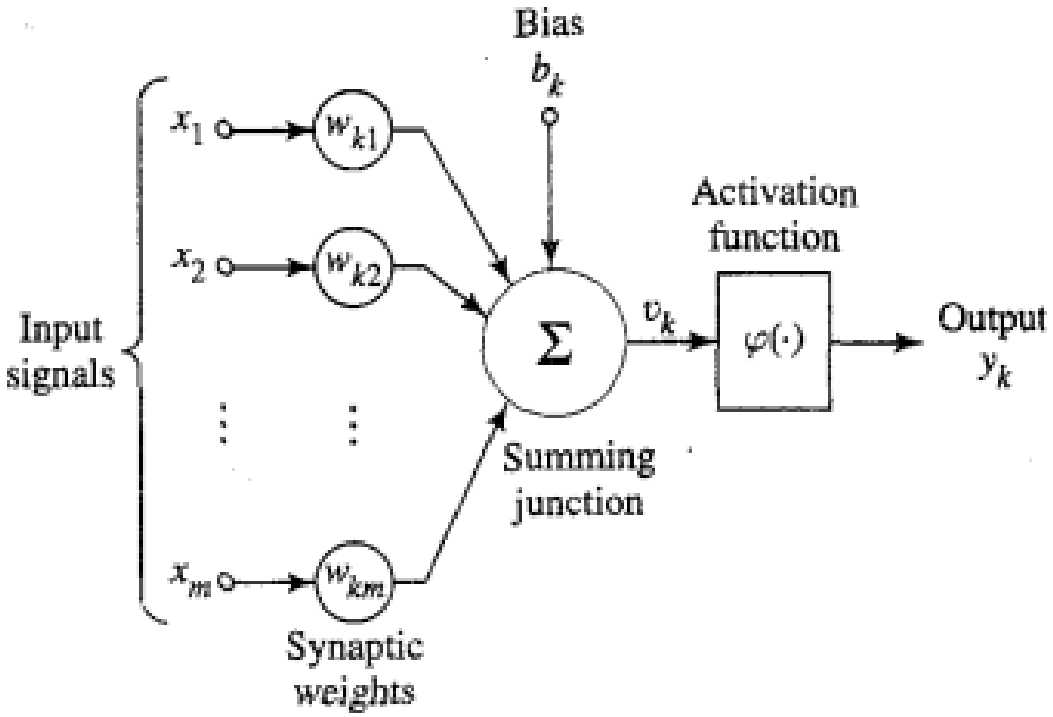


Neurona Artificial



Redes Neuronales

Neuronal Artificial



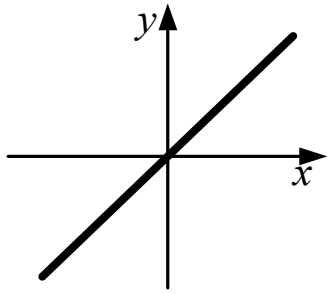
$$net = \sum_{i=1}^{m} w_{ki} x_i + b_k$$

$$y_k = f(net)$$

Redes Neuronales

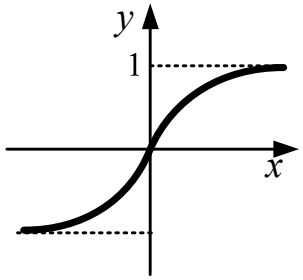
Cualquier función continua (diferenciable) se puede utilizar como una función de activación en una red neural.

El comportamiento no lineal de las redes neuronales se hereda de las funciones de activación no lineales usados.



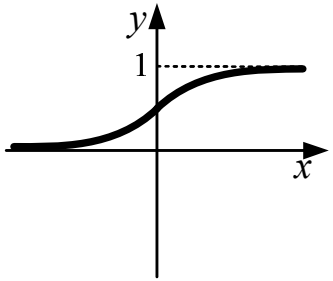
$$y = f(x) = x$$

Función Lineal



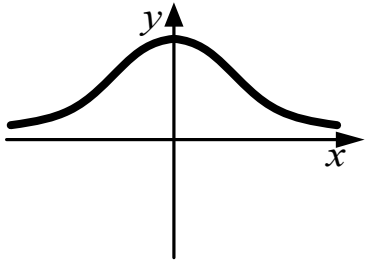
$$y = f(x) = \frac{2}{1+e^{-2x}} - 1$$

Sigmoide



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

Logaritmica

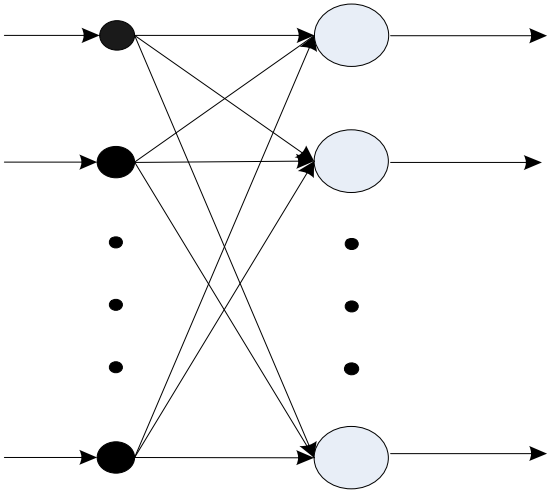


$$y = f(x) = e^{-ax^2}$$

Radial Basica

Arquitecturas de Redes

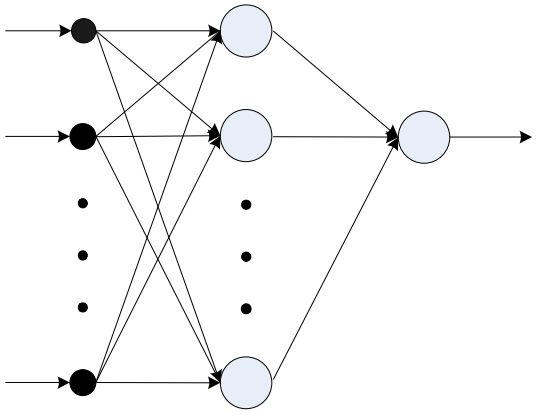
Single layer feedforward network (Single layer perceptron)



Input layer

Output layer

Multilayer feedforward network (Multilayer perceptron)



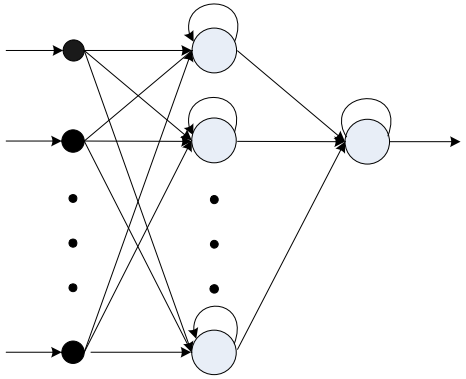
Input layer

Hidden layer

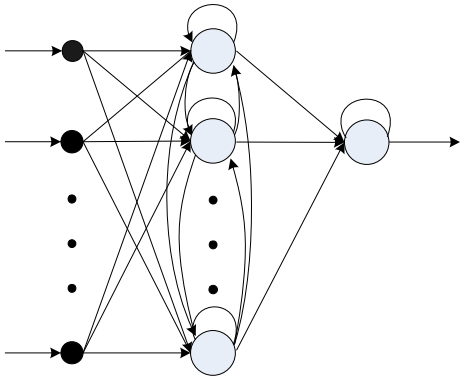
Output layer

Arquitecturas de Redes

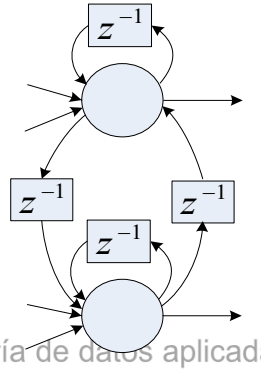
Diagonal recurrent networks



Fully recurrent networks

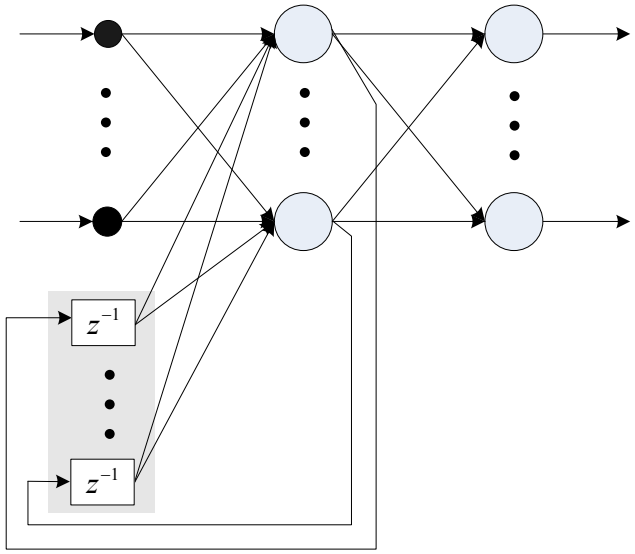


Delay element in a recurrent network

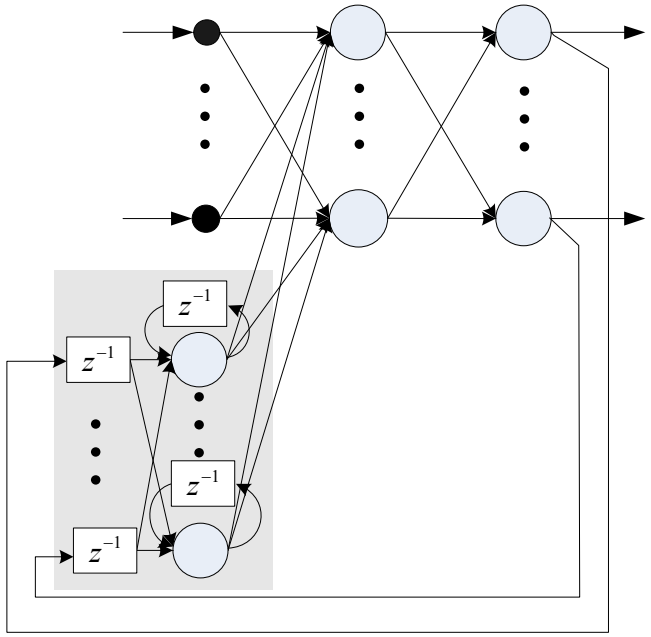


Arquitecturas de Redes

Elman's recurrent networks



Jordan's recurrent networks



El Perceptron (Single layer perceptron)

Metáfora : Aprobación de crédito

Información del aplicante :

Edad	32
Genero	Masculino
Salario mensual	3.700.000
Años en residencia	5
Años trabajando	6
Deuda corriente	7.000.000
…	…

¿Se aprueba el crédito?

El Perceptron (Single layer perceptron)

Para las entradas $\mathbf{x} = (x_1, \cdots, x_d)$ "Atributos del cliente"

Se aprueba el crédito si $\sum_{i=1}^{d} w_i x_i > \text{threshold}$

Se niega si $\sum_{i=1}^{d} w_i x_i < \text{threshold}$

Esta formula lineal es escrita como:

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

El Perceptron (Single layer perceptron)

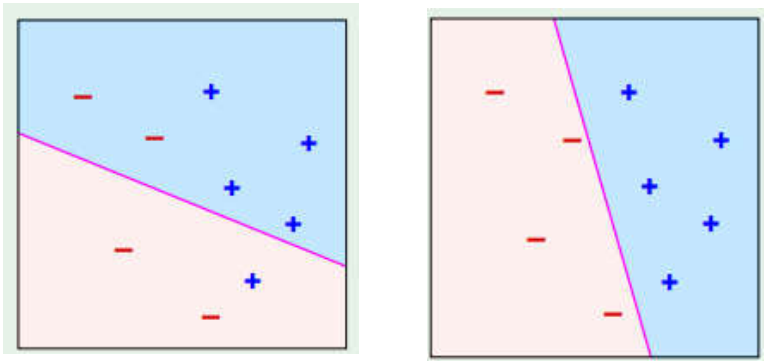
$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i\, x_i\right) + \quad w_0 \quad\right)$$

Si introducimos una coordinada artificial $X_0=1$

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i\, x_i\right)$$

En forma del vector:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\intercal}\mathbf{x})$$

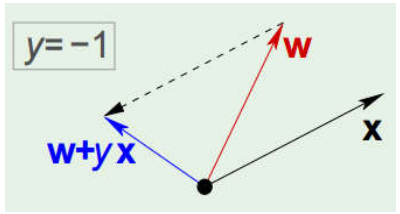


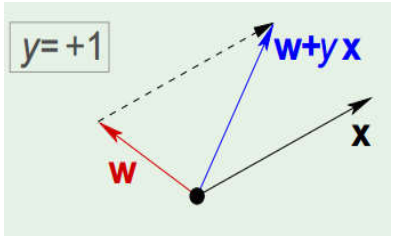
Datos “linealmente separables”

El Perceptron (Single layer perceptron)

La representación del perceptron:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x})$$





Dado un conjunto de entrenamiento:

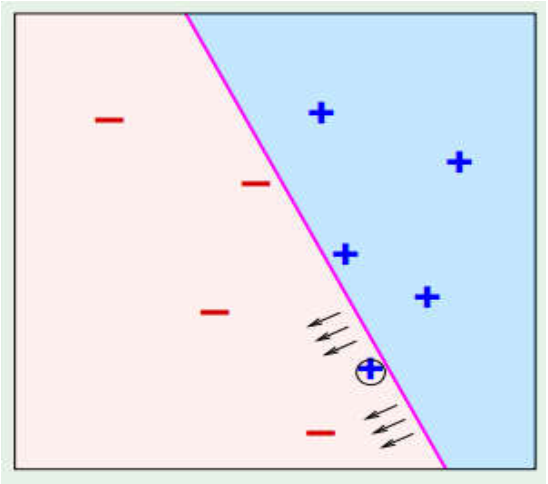
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

Elegir un punto mal clasificado :

$$\text{sign}(\mathbf{w}^\intercal \mathbf{x}_n) \neq y_n$$

Y actualizar la ponderación del vector:

$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$$



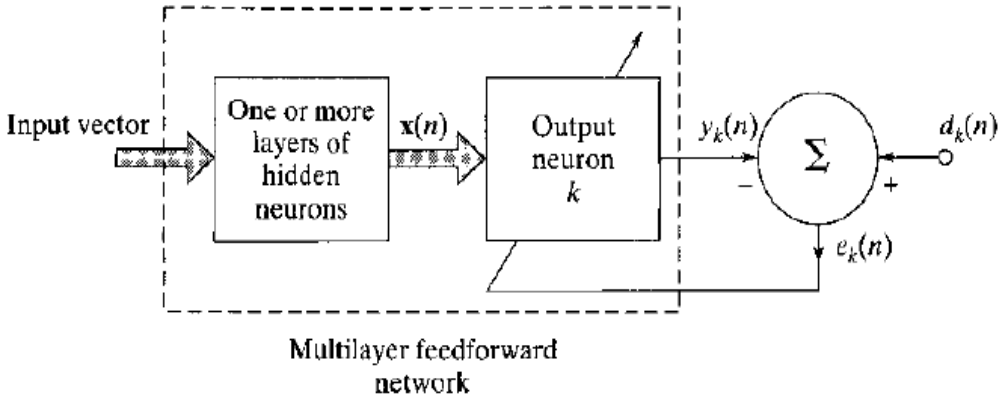
Regla de entrenamiento delta (Widrow-Hoff Rule)

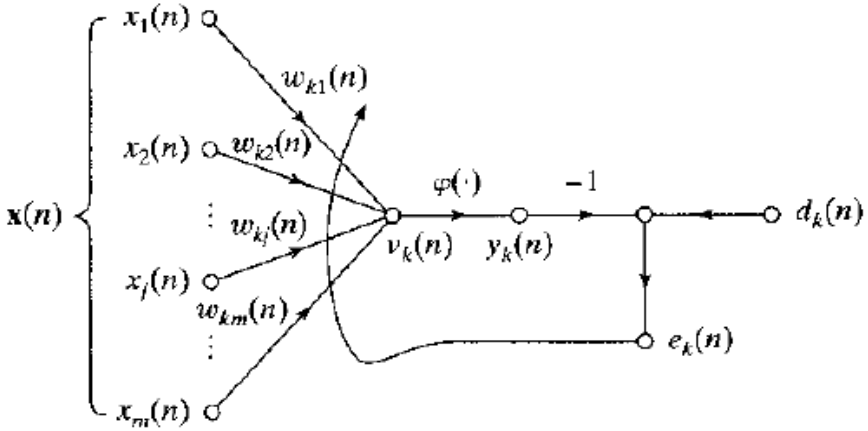
- Soluciona el problema de la convergencia anterior
- Basada en gradiente descendente
- Se ajusta asintóticamente a la representación deseada
- Medida de error:

$$E(\vec{w}) = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2$$

- D es el conjunto de ejemplos de entrenamiento,
- td es la salida de la función buscada
- od es la salida de la red.

Regla de entrenamiento delta (Widrow-Hoff Rule)



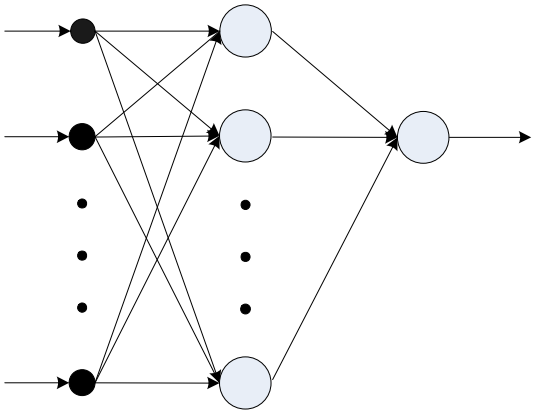


Minimización de la función de costo)

$$E(\vec{w}) = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2$$

(Multilayer perceptron)

¿Cómo podemos aumentar la capacidad de representación de un perceptrón?



Input layer **Hidden layer** **Output layer**

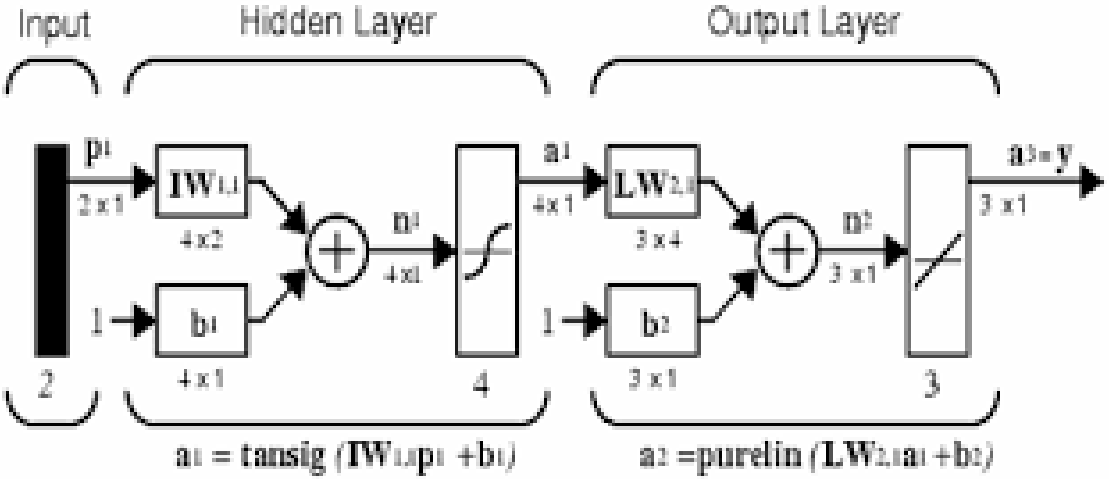
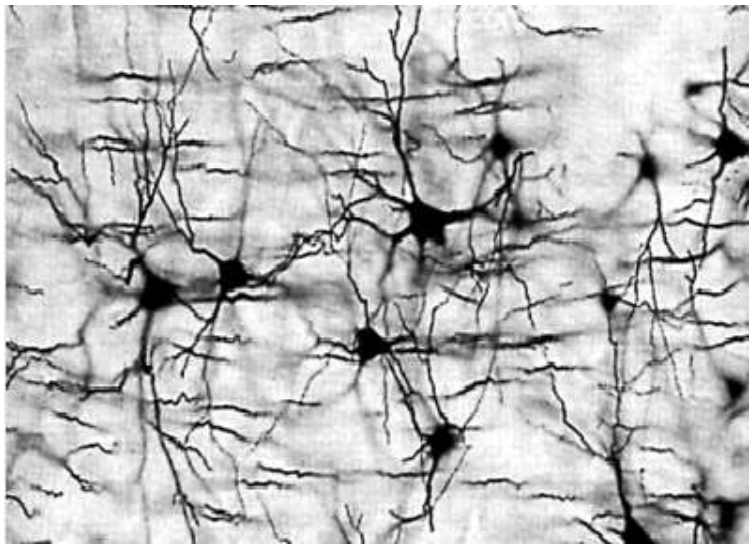


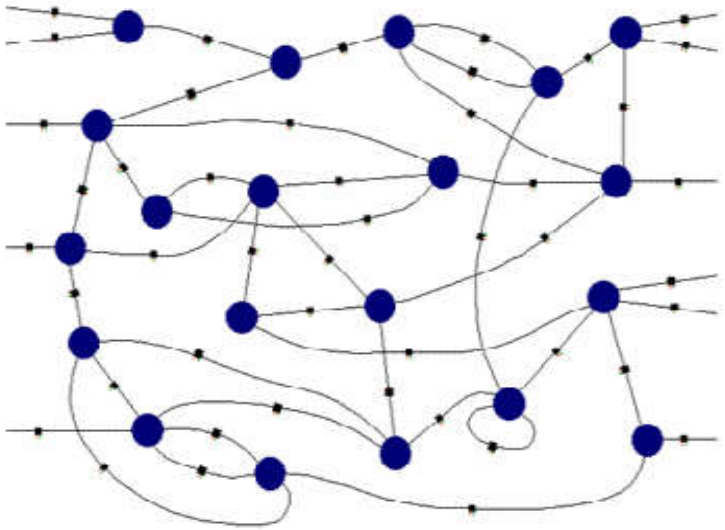
Fig. 4. Ejemplo de topología de red Feedforward.

(Multilayer perceptron)

Red neuronal



Estructura Neuronal



Algoritmo Backpropagation

- Aprende los wi para una red neuronal multi-capa, con funciones de activación derivables
- Versión fully-connected y conexión estricta
- Función de error:

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in O} (t_{kd} - o_{kd})^2$$

en donde D es el conjunto de ejemplos y O el conjunto de nodos de salida.

- tkd será la salida esperada para el ejemplo d en el nodo k,
- okd la salida obtenida para el nodo k y el ejemplo d. Convergencia a m´ınimos locales

Algoritmo Backpropagation

Similitud con regla delta

- En regla delta, wi se actualizan con ηxi(t − o).
- En backpropagation se suma a los wij el producto ηxij δi .
- En realidad δi es el producto de (t − o) por σ()0 en los nodos de salida En los nodos de la capa oculta el producto es de σ()0 y P k∈O whk δk

Algoritmo Backpropagation

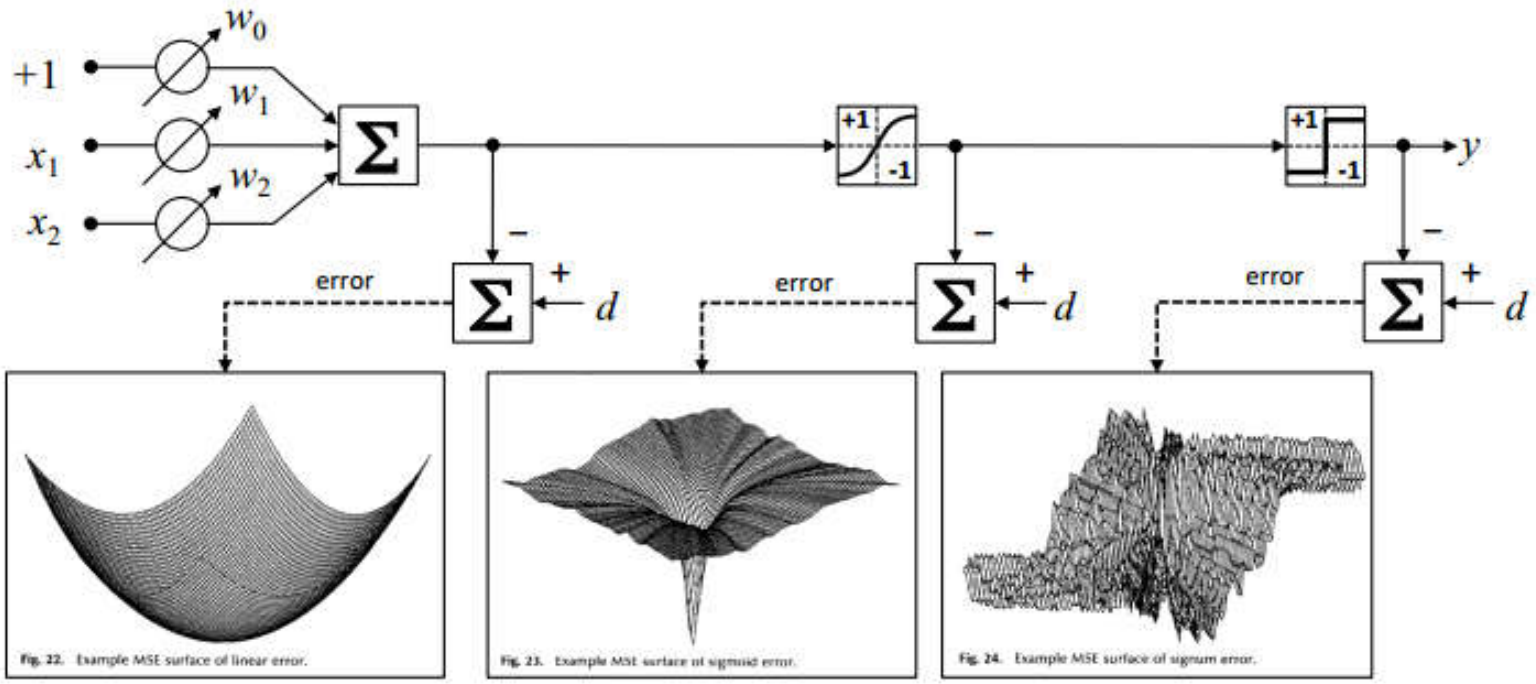
Ventajas:

1. Es fácil de usar con pocos parámetros para ajustar.
2. Es un algoritmo fácil de implementar.
3. Es aplicable a un amplio rango de problemas
4. Es capaz de formar arbitrariamente complejos mapeos no lineales

Desventajas:

1. Hay una incapacidad para saber cómo generar con la máxima precisión un procedimiento de mapeo arbitrario.
2. Es difícil saber cuantas neuronas y capas son necesarias.
3. El aprendizaje puede ser lento.
4. Los nuevos aprendizajes sustituyen a los aprendizajes antiguos a menos que los viejos patrones se repitan en el proceso de formación.
5. No tiene la capacidad de detectar cosas nuevas, sólo cosas iguales o parecidas a las que se han usado para su entrenamiento.

Complejidad de la función de error.



Redes Neuronales

Ventajas:

1. Pueden adaptarse a problemas de clasificación y numéricos
2. Flexibilidad en el modelamiento.
3. Capases de modelar patrones mas complejos que casi cualquier algoritmo
4. Pocos supuestos sobre las elaciones subyacentes de los datos

Desventajas:

1. Extremadamente intensivo computacionalmente (depende de la topología de la red)
2. Muy propensos al sobreajuste.
3. Resultados complejos , difíciles , sino imposibles, de interpretar

THANK YOU!

ANY QUESTIONS?

Jun Akizaki - http://thepopp.com

Used Font: Roboto Light & Roboto Condensed Light

Icon: Font generated by flaticon.com under CC BY. The authors are: Stephen Hutchings.
Changed the color by Photoshop

World Map: http://www.tutsking.com/vectors/world-dots-map
Changed the color by Photoshop