

Tema 1.3

Repaso de Redes Neuronales

Deep Learning

Máster Oficial en Ingeniería Informática

Universidad de Sevilla

Contenido

- Regresión lineal
- El perceptrón
- Inspiración biológica
- Neurona artificial
- Redes multicapa
- Aplicaciones

Regresión lineal

- Cómo depende el **valor** de una **variable continua (objetivo)** con respecto variables de entrada (**características**).
- La función f aproxima y a través de la entrada x como:

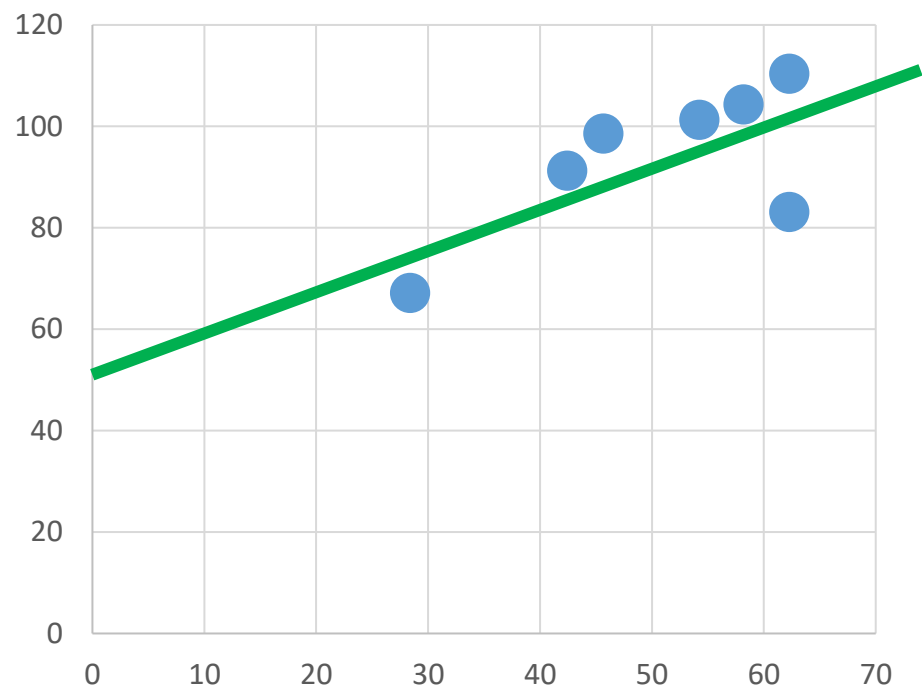
$$y \approx f = Wx$$

- Siendo $x \in \mathbb{R}^n$ y $W \in \mathbb{R}^n$
- Si pensamos en 1 dimensión, es la ecuación de la recta.
- Pero estaría pasando siempre por el punto (0,0). Por tanto, añadimos un sesgo (**bias**):

$$f = Wx + b$$

Regresión lineal

Tamaño (m ²)	Precio (€)
42,45	91241
54,25	101251
32,5	83051
62,3	110341
28,4	67124
45,69	98525
58,2	104251



x

y

$$y \approx f = Wx + b = 0,8233x + 52,096$$

Regresión lineal

X_1	X_2	X_3	X_4	Y
Tamaño (m ²)	Número habitaciones	Número plantas	Años construido	Precio (€)
42,45	2	1	10	91241
54,25	3	2	23	101251
32,5	2	1	5	83051
62,3	4	3	41	110341
28,4	1	1	24	67124

- Número características: $n = 4$
- y^i valor de la variable y para el ejemplo i
- x^i las características del ejemplo i
- x_j^i el valor de característica j del ejemplo i

$$f = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$

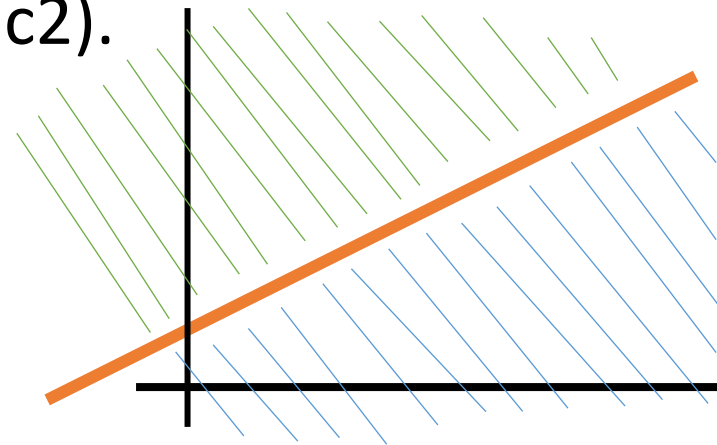
$$f = 80 + 0,9x_1 + 0,5x_2 + 3x_3 - 2x_4$$

Perceptrón simple

- *Frank Rosenblatt, ~1957*
- Clasificación **binaria** (dos clases, c1 y c2).
- La función f retorna valores **0** (para c1) o **1** (para c2).

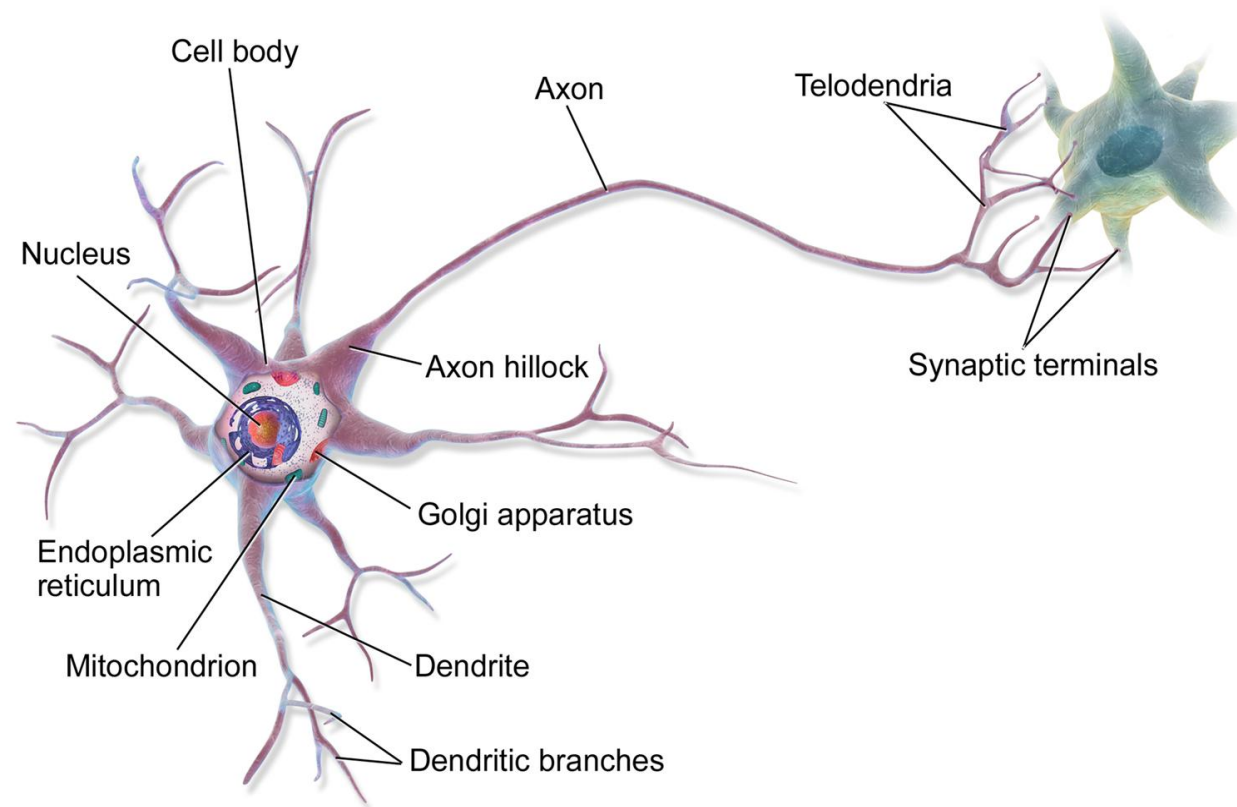
$$f = S(Wx + b)$$

- Donde la función S , o umbral, es:
 - 1, si $Wx + b > \text{valor_umbral}$
 - 0 en otro caso
- Si pensamos en 2 dimensiones, sería partir el plano mediante una recta.



Inspiración biológica

- La **neurona**, 1888, Santiago Ramón y Cajal
- Podemos reconocer diferentes partes:
 - El cuerpo central, llamado **soma**, que contiene el núcleo celular.
 - Una prolongación del soma, el **axón**.
 - Una ramificación terminal, **dendritas**.
 - Una zona de conexión entre una neurona y otra, conocida como **sinapsis**.

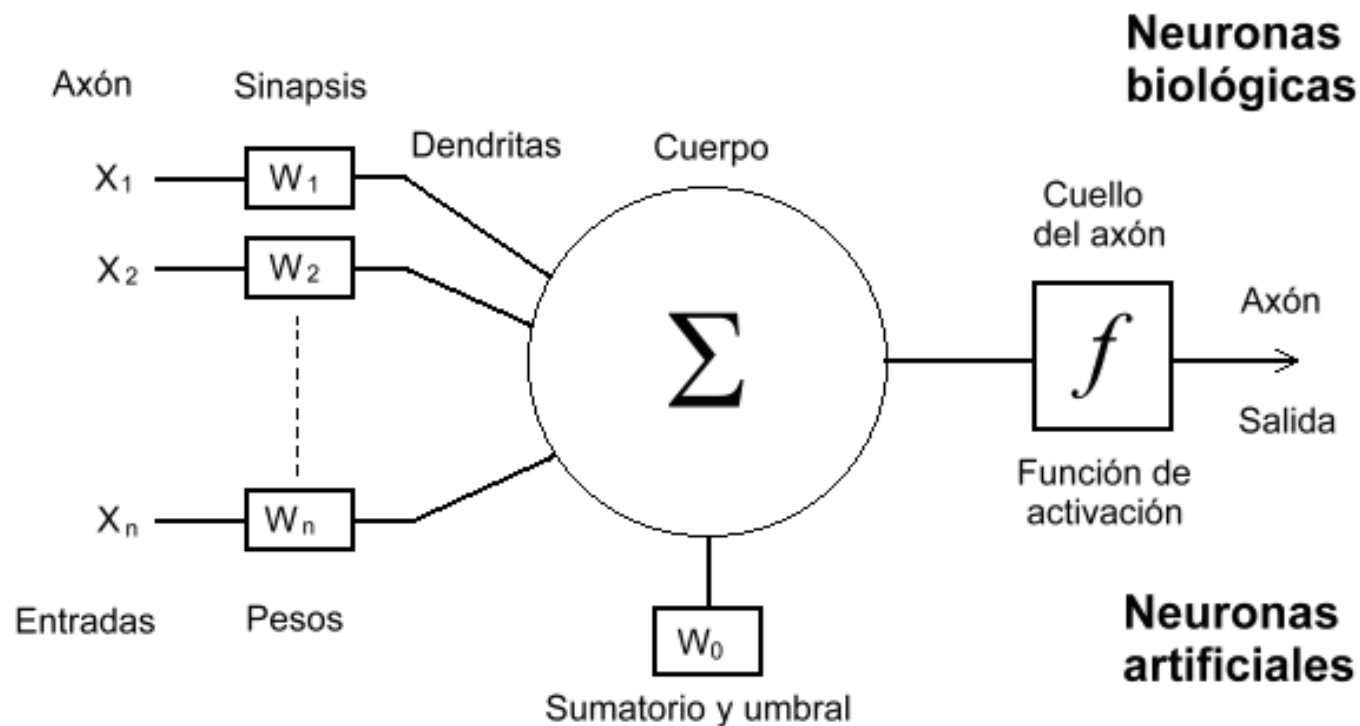


Neurona artificial

Perceptrón (1943, McCulloch y Pitts), modelo neuronal con n entradas, que consta de:

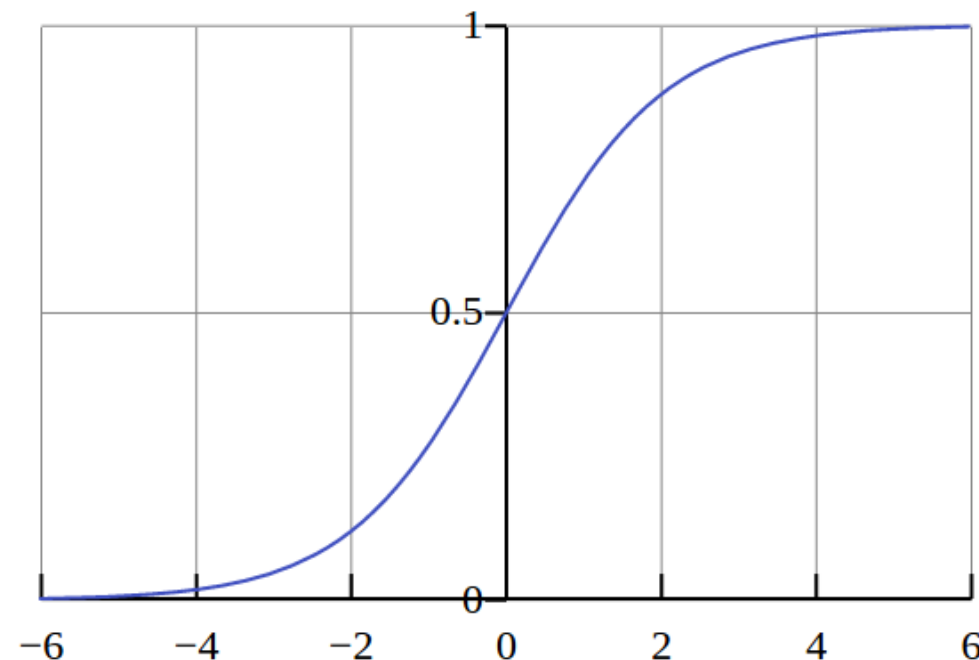
- Un conjunto de **entradas** x_1, \dots, x_n
- Los **pesos** sinápticos w_1, \dots, w_n , correspondientes a cada entrada
- Una entrada de **umbral**, con w_0 y $x_0=1$
- Una función de **agregación**, Σ
- Una función de **activación**, f
- Una **salida**, o

$$o = f\left(\sum_{i=0}^n w_i x_i\right)$$



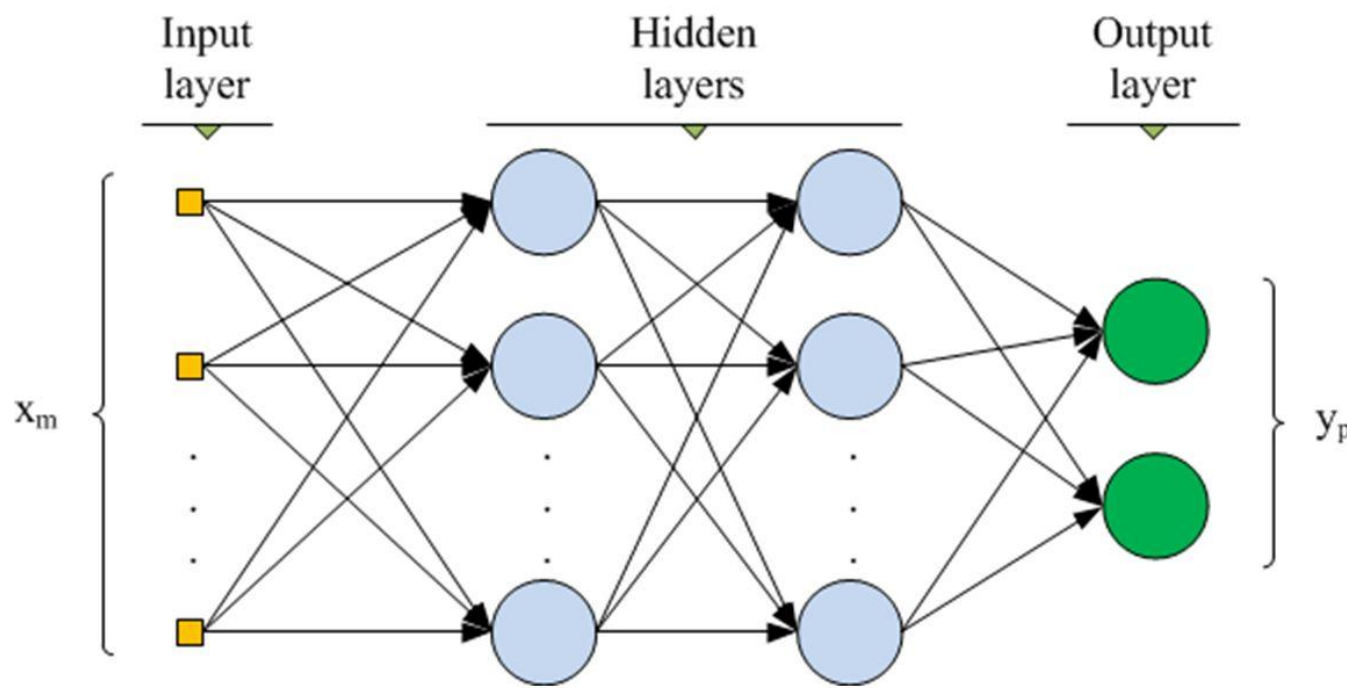
Neurona artificial

- La Función de Activación busca:
 - **Normalizar** valores ($[0,1]$, $[-1,1]$,...)
 - Preferiblemente, funciones **no lineales**.
 - **Funciones** de activación clásicas:
 - Bipolar: $\text{sgn}(x) = \begin{cases} 1, & \text{si } x > 0 \\ -1, & \text{si } x \leq 0 \end{cases}$
 - Umbral: $\text{umbral}(x) = \begin{cases} 1, & \text{si } x > 0 \\ 0, & \text{si } x \leq 0 \end{cases}$
 - Sigmoide o logística: $S(x) = \frac{1}{1+e^{-x}}$
 - **Sigmoide** es muy común (**regresión logística**). Al devolver valores entre 0 y 1, se suele usar cuando se quiere representar una probabilidad



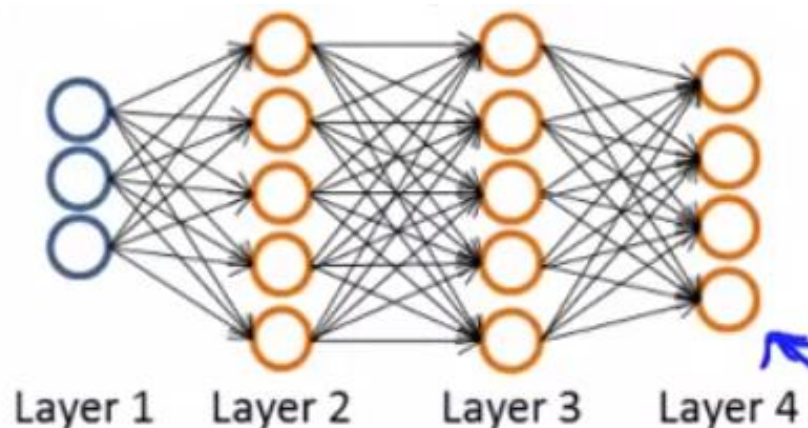
Redes multicapa

- Organizando perceptrones simples en múltiples capas (MLP):
 - Capa de **neuronas** de entrada
 - Capa/s de **neuronas** ocultas
 - Capa de **neuronas** de salida
- Cada neurona de una capa conectada con todas de la capa anterior (**fully connected**)
- Capa de **entrada** sin pesos

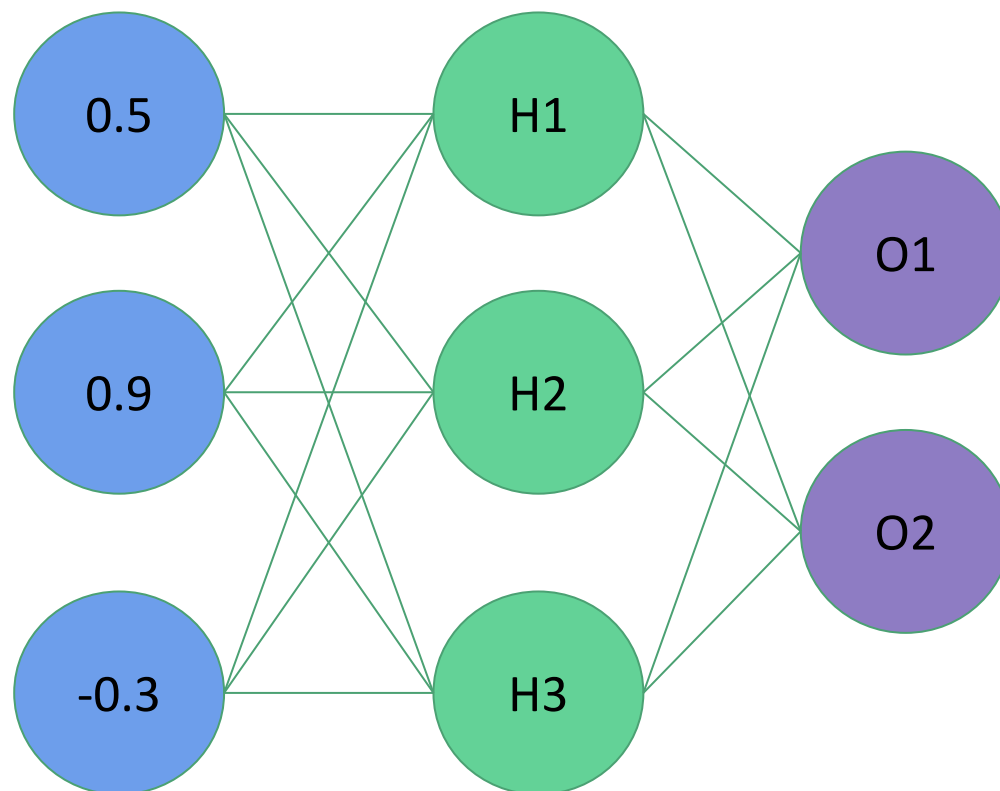


Redes multicapa

- ¿Cuántas neuronas en capa de **salida**?
- Clasificación binaria:
 - $K=2$ clases
 - Variable de salida es $y=0$ o 1
 - 1 unidad de salida
- Clasificación multiclase:
 - $K \geq 3$ clases
 - K variables de salida, $y_1 \dots y_k$
 - K unidades de salida
 - Se considera la más alta



Redes multicapa



H1 Weights = (1.0, -2.0, 2.0)

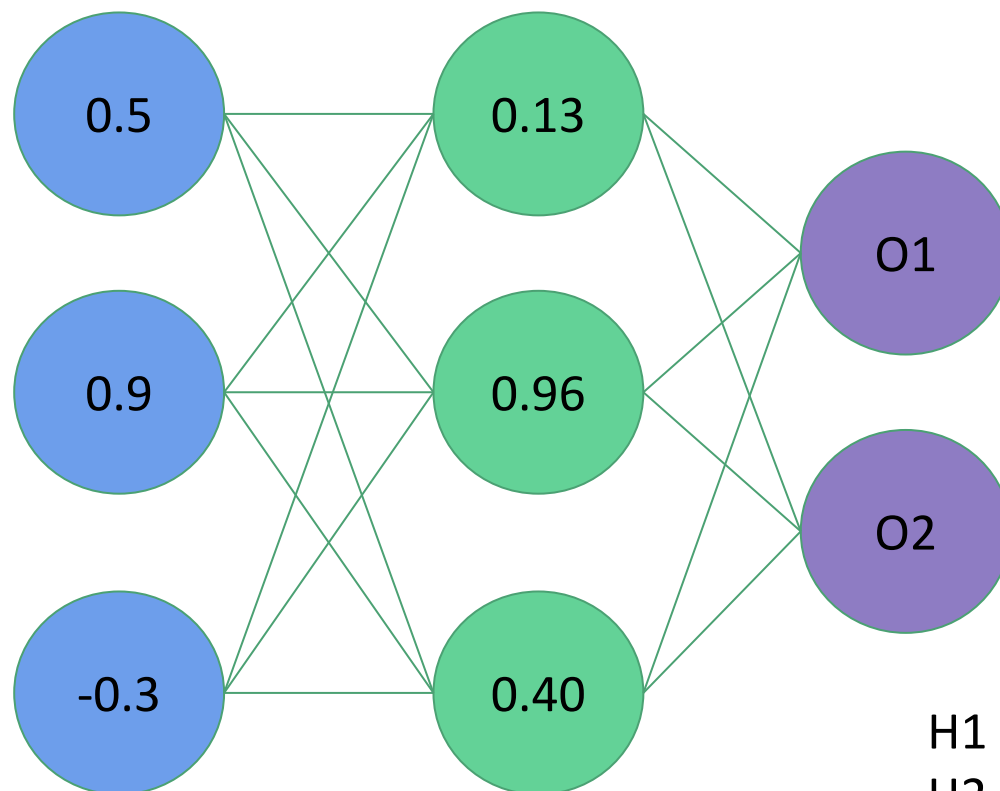
H2 Weights = (2.0, 1.0, -4.0)

H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)

O2 Weights = (0.0, 1.0, 2.0)

Redes multicapa



H1 Weights = (1.0, -2.0, 2.0)

H2 Weights = (2.0, 1.0, -4.0)

H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)

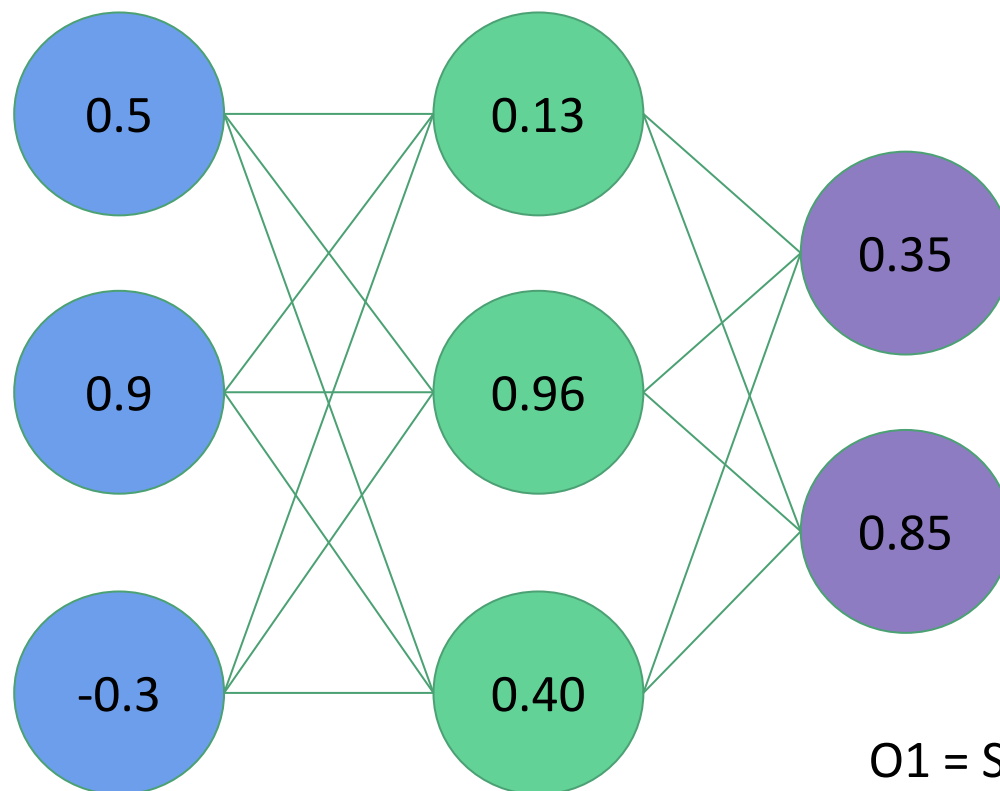
O2 Weights = (0.0, 1.0, 2.0)

$$H1 = S(0.5 * 1.0 + 0.9 * -2.0 + -0.3 * 2.0) = S(-1.9) = 0.13$$

$$H2 = S(0.5 * 2.0 + 0.9 * 1.0 + -0.3 * -4.0) = S(3.1) = 0.96$$

$$H3 = S(0.5 * 1.0 + 0.9 * -1.0 + -0.3 * 0.0) = S(-0.4) = 0.40$$

Redes multicapa



H1 Weights = (1.0, -2.0, 2.0)

H2 Weights = (2.0, 1.0, -4.0)

H3 Weights = (1.0, -1.0, 0.0)

O1 Weights = (-3.0, 1.0, -3.0)

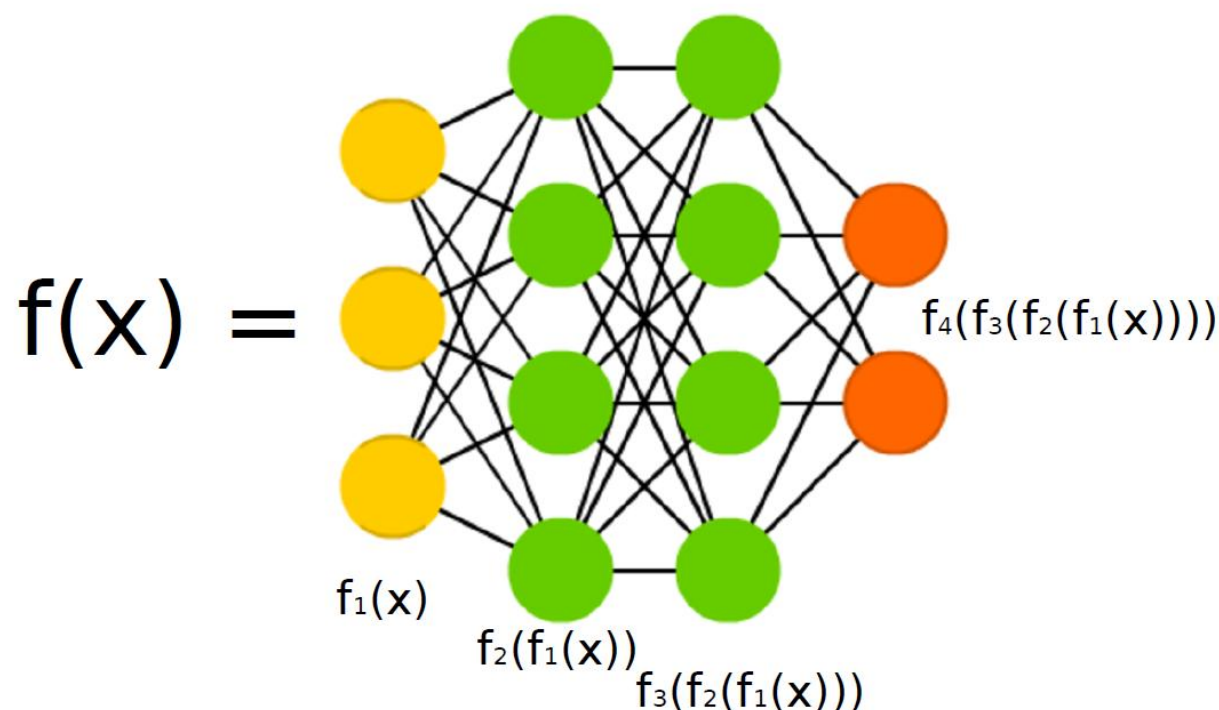
O2 Weights = (0.0, 1.0, 2.0)

$$O1 = S(0.13 * -3.0 + 0.96 * 1.0 + 0.40 * -3.0) = S(-.63) = 0.35$$

$$O2 = S(0.13 * 0.0 + 0.96 * 1.0 + 0.40 * 2.0) = S(1.76) = 0.85$$

Redes multicapa

- Una red neuronal **feed forward** representa una función $f(x)$
 - Cada capa de la red representa una función
 - Es una **composición** de diferentes funciones



Aplicaciones

- Para problemas que se pueden expresar **numéricamente** (discretos o continuos)
- Se suelen utilizar en dominios en los que el **volumen de datos** es muy alto, y puede presentar ruido: imágenes, audio, bioinformática, etc.
- En los que interesa la solución, pero **no el por qué** de la misma
- Problemas en los que es asumible que se necesite previamente un **tiempo largo de entrenamiento** de la red
- Y en los que se requieren **tiempos cortos para evaluar** una nueva instancia

Recapitulación

- Las redes neuronales son una abstracción del concepto biológico.
- El **perceptrón**, o la neurona artificial básica, procesa las entradas mediante una función de agregación y usando unos pesos, y pasando por una función de activación.
- Se busca la **no linealidad**.
- Una red multicapa representa una función, que es el resultado de la composición de las mismas capas.