Escuela Técnica Superior de Ingeniería Informática

Máster Universitario en Ingeniería Informática

Tema 1.3
Introducción a las Redes Neuronales

Deep Learning

Máster Oficial en Ingeniería Informática

Universidad de Sevilla

Contenido

- Regresión lineal
- El perceptrón
- Inspiración biológica
- Neurona artificial
- Redes multicapa
- Formulación matricial
- 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 $\mathrm{x} \in \mathbb{R}^n$ y $\mathrm{W} \in \mathbb{R}^n$
- Si pensamos en 1 dimensión, es la ecuación de la recta.
  - Pero estaría pasando por el punto (0,0)
- Añadimos **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

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

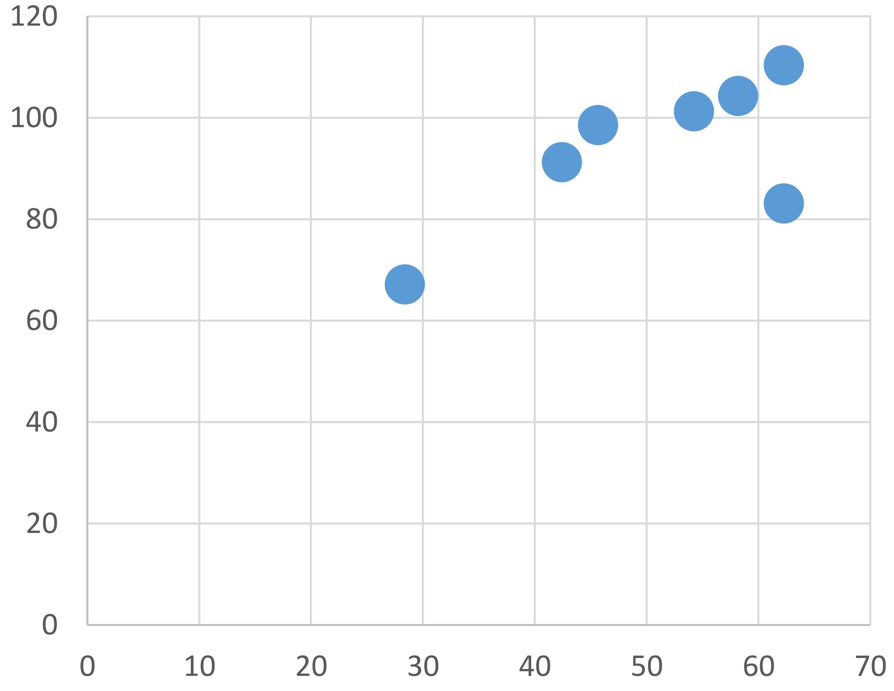
Regresión lineal

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

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

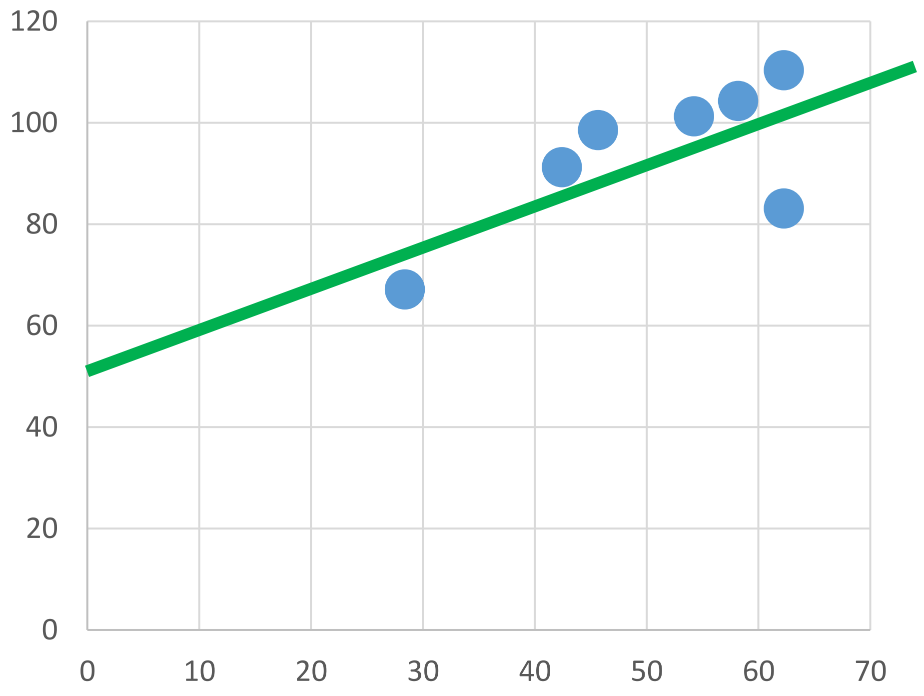


x y

Regresión lineal

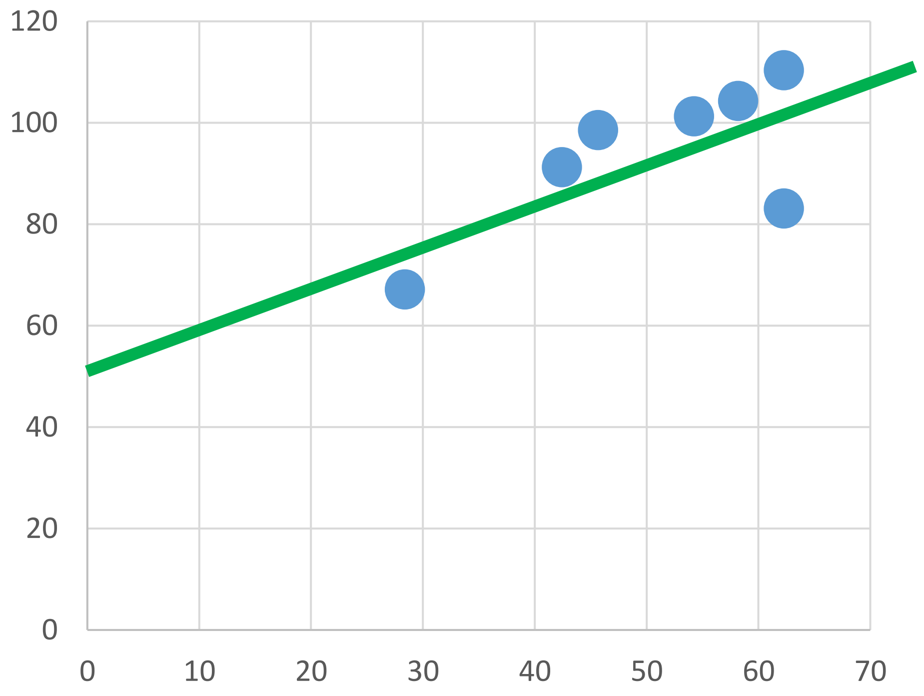
Tamaño (m²)	Precio (€)
42,45	91241
54,25	101251
32,5	83051
62,3	110341
28,4	67124
45,69	98525
58,2	104251

x y



Regresión lineal

Tamaño (m²)	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

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

Regresión lineal

x_1 Tamaño (m^2)	x_2 Número habitaciones	x_3 Número plantas	x_4 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

Regresión lineal

x_1 Tamaño (m^2)	x_2 Número habitaciones	x_3 Número plantas	x_4 Años construido	y 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

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

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

- 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

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

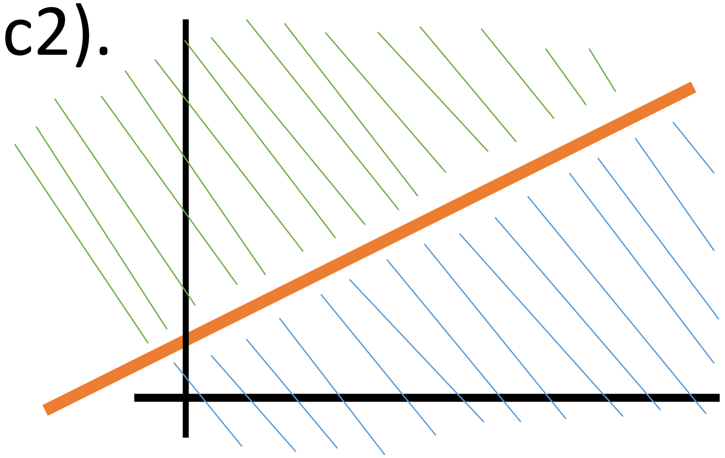
- Número características: n = 4
- y^i valor de la variable y para el ejemplo i
- x^i las características del ejemplo i
- x_j^i el valor de característica j del ejemplo i

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

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

Perceptrón simple

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

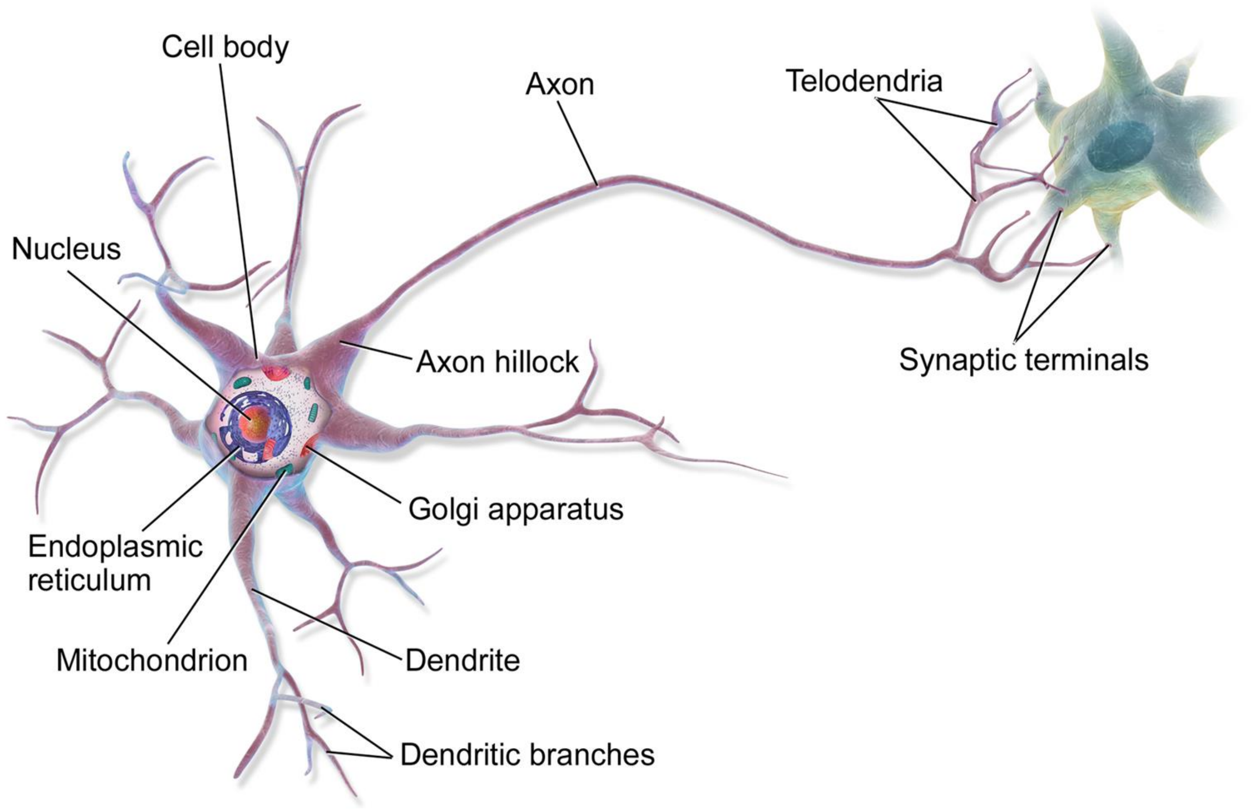


$$f = S(Wx + b)$$

  - Donde la función S, o umbral, es:
    - 1, si Wx + b > 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**.

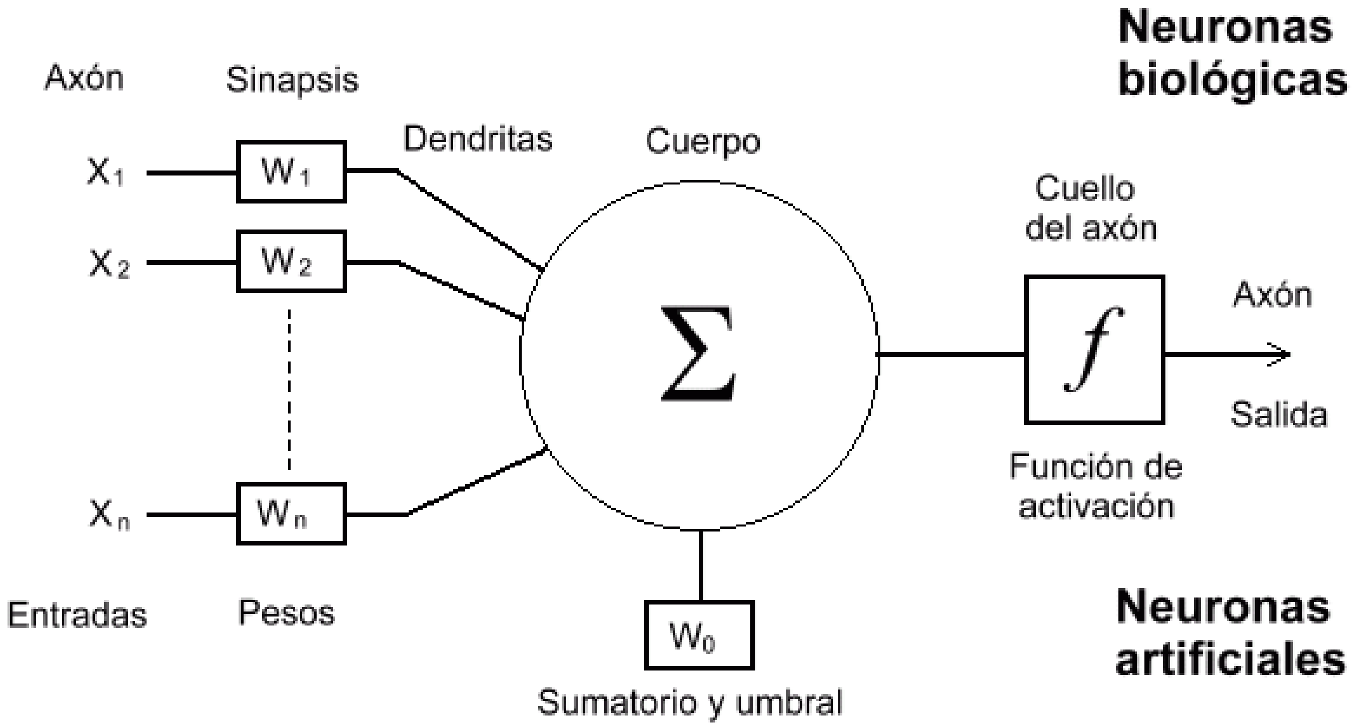


Neurona artificial

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

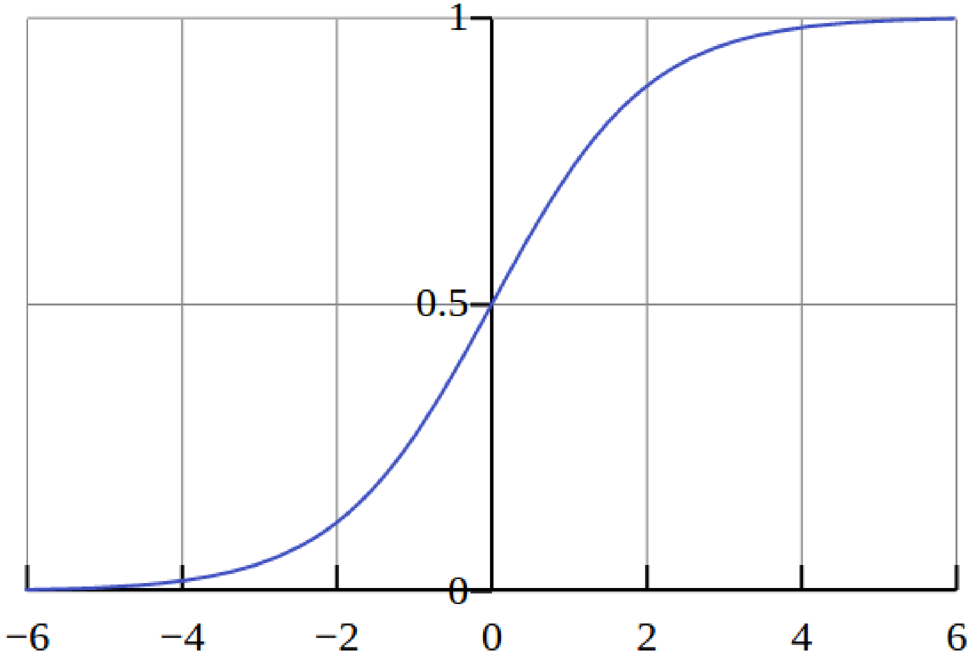
- Un conjunto de **entradas $x_1,\ldots x_n$**
- Los **pesos** sinápticos **$w_1,\ldots w_n$**, correspondientes a cada entrada
- Una entrada de **umbral**, con **w_0** y x_0=1
- Una función de **agregación, Σ**
- Una función de **activación, f**
- Una **salida**, ***o***

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



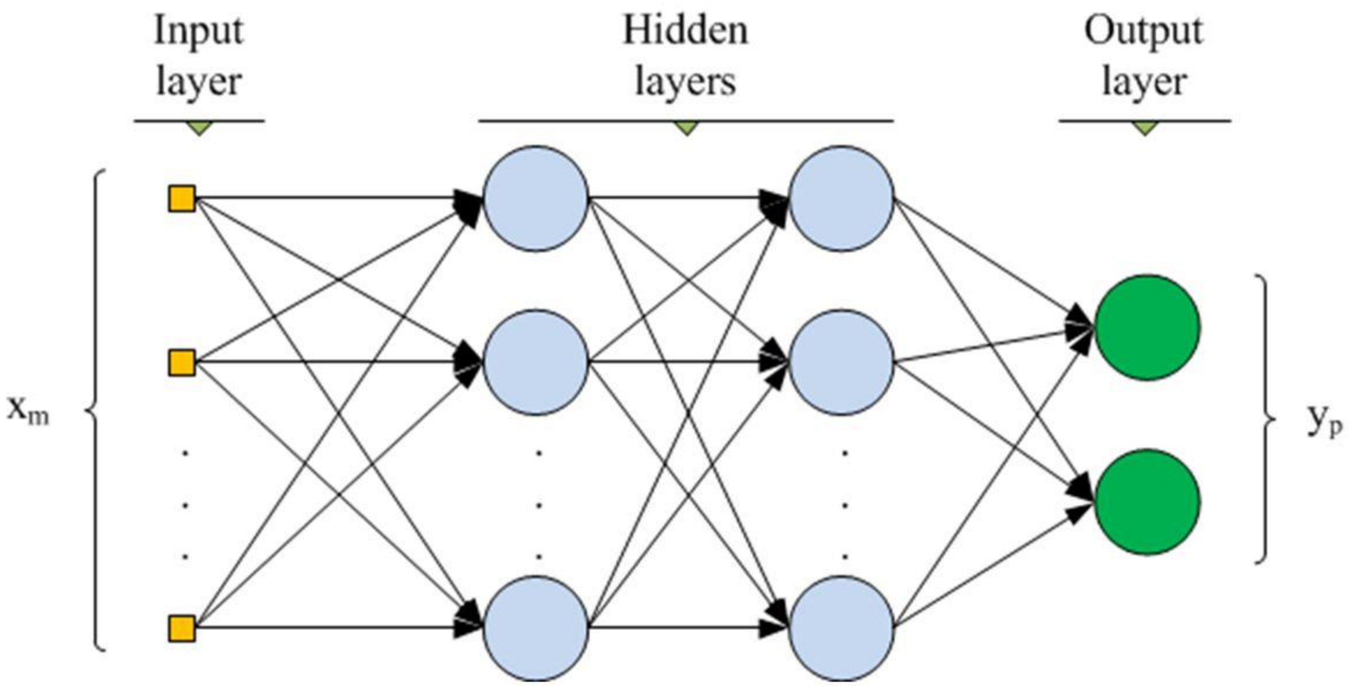
Neurona artificial

- La Función de Activación busca:
  - **Normalizar** valores ([0,1], [-1,1],…)
  - Preferiblemente, funciones **no lineales**.
  - **Funciones** de activación clásicas:
    - Bipolar: $sgn(x) = \begin{cases} 1, si\ x > 0 \\ -1, si\ x \leq 0 \end{cases}$
    - Umbral: $umbral(x) = \begin{cases} 1, si\ x > 0 \\ 0, si\ x \leq 0 \end{cases}$
    - 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



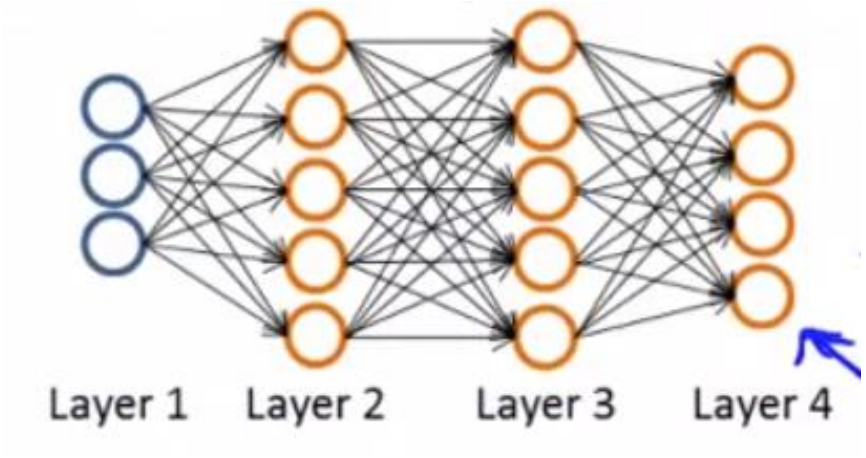
Redes multicapa

- 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



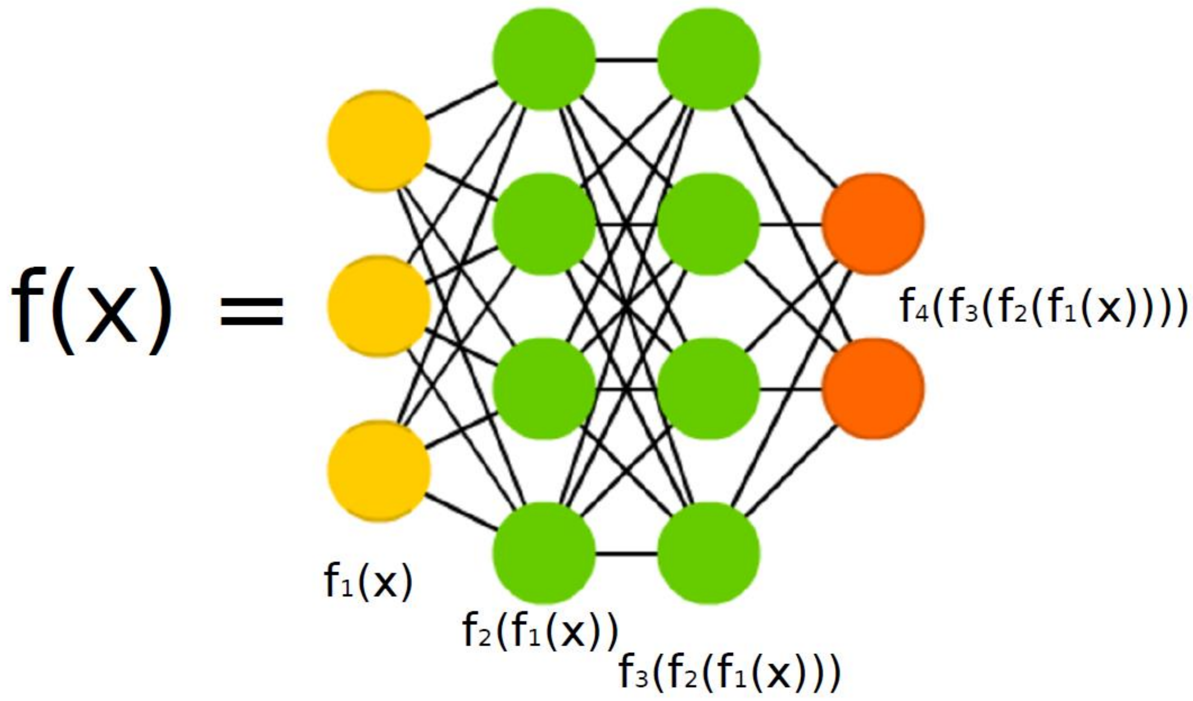
Redes multicapa

- ¿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≥3 clases
  - K variables de salida, , $y_1..y_k$
  - K unidades de salida
  - Se considera la más alta

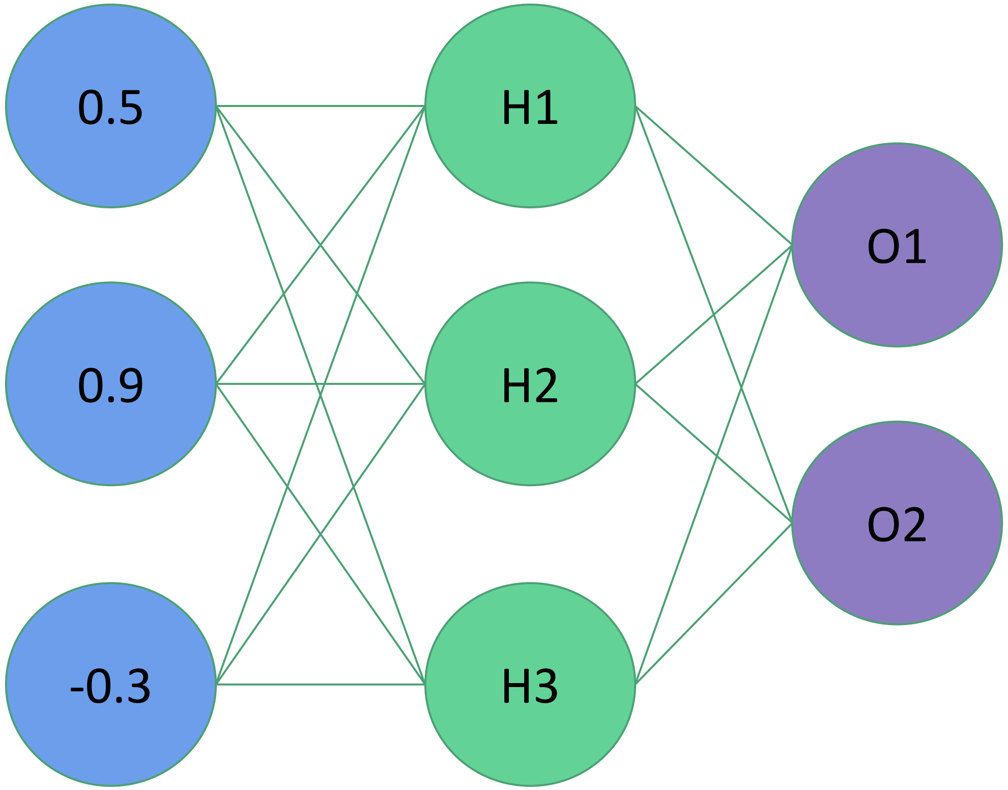


Redes multicapa

- También se les denomina redes **feed-forward**, diferentes **funciones** que **componen** a **f** hasta dar la salida **o**:



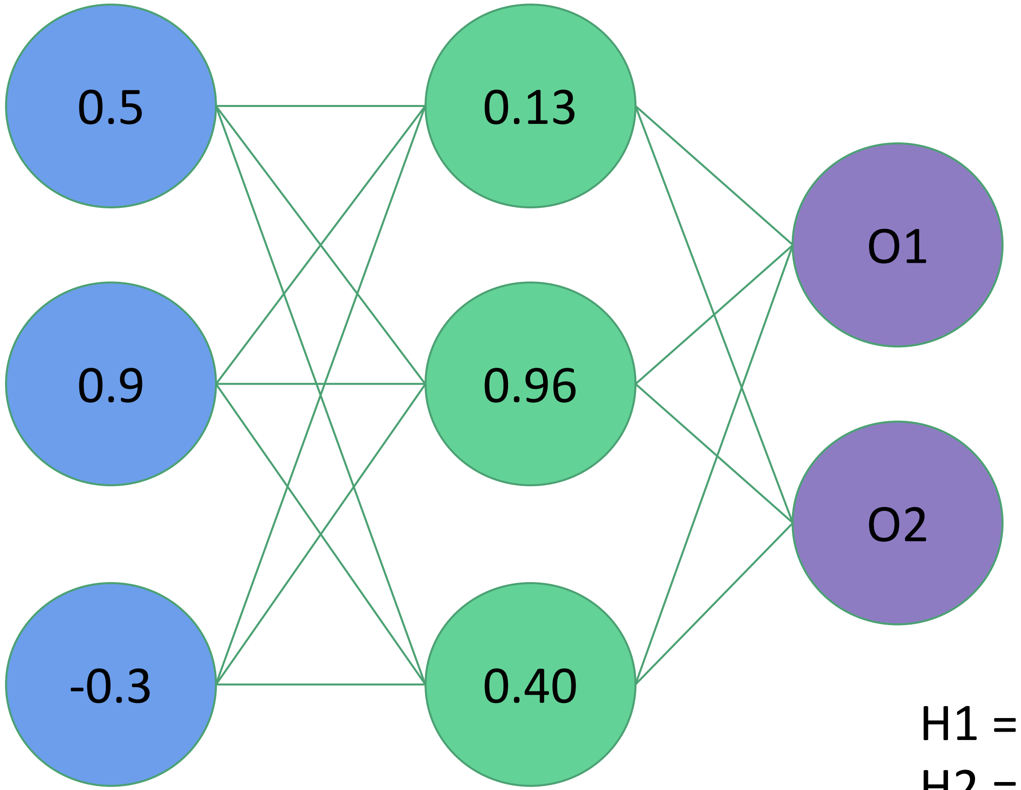
Redes multicapa



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

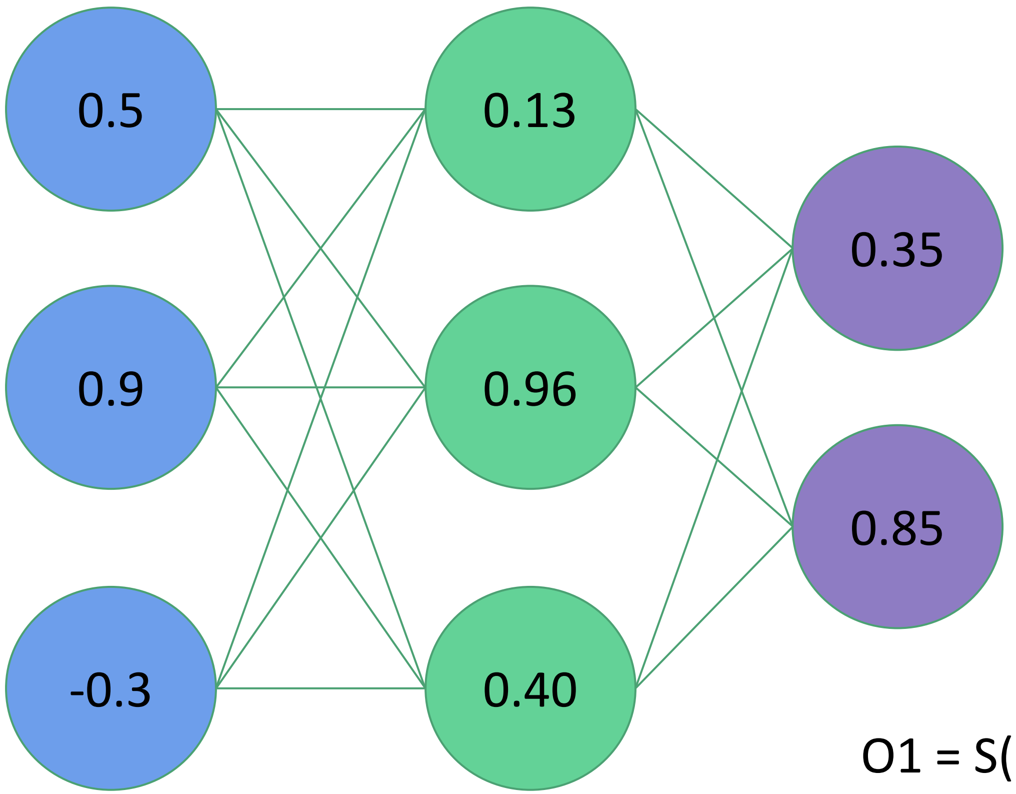


H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

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

Formulación matricial

- Una capa oculta (k neuronas)

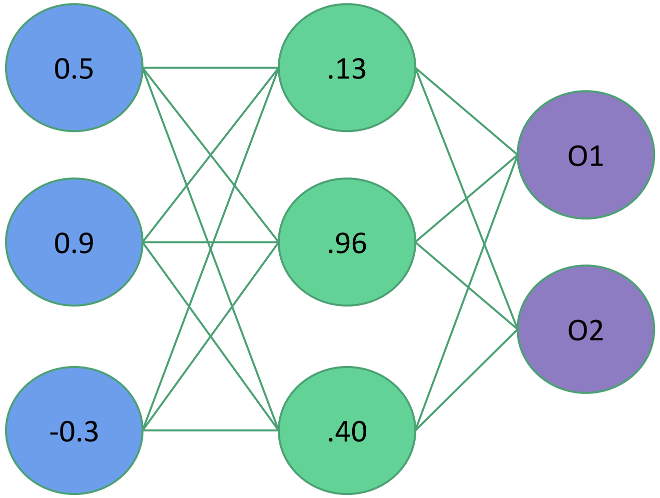
$$o = f\left(\sum_{i=0}^{n}\sum_{j=0}^{k_h} w_{ij}\, x_i\right)$$

w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}
w_{20}	w_{21}	w_{22}	w_{23}	w_{24}	w_{25}	w_{26}
w_{30}	w_{31}	w_{32}	w_{33}	w_{34}	w_{35}	w_{36}
w_{40}	w_{41}	w_{42}	w_{43}	w_{44}	w_{45}	w_{46}

x_0	x_1	x_2	x_3	x_4	x_5	x_6

S(W * X) = o

Formulación matricial



H1 Weights = (1.0, -2.0, 2.0)
H2 Weights = (2.0, 1.0, -4.0)
H3 Weights = (1.0, -1.0, 0.0)

Hidden Layer Weights, Inputs, Hidden Layer Outputs

$$S\left(\begin{bmatrix} 1.0 & -2.0 & 2.0 \\ 2.0 & 1.0 & -4.0 \\ 1.0 & -1.0 & 0.0 \end{bmatrix} * \begin{bmatrix} 0.5 \\ 0.9 \\ -0.3 \end{bmatrix}\right) = S\left(\begin{bmatrix} -1.9 & 3.1 & -0.4 \end{bmatrix}\right) = \begin{bmatrix} 0.13 & 0.96 & 0.4 \end{bmatrix}$$

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 neuronal artificial se puede implementar mediante una formulación **matricial**.