UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Licenciatura en Ciencia de Datos

Introducción al Aprendizaje Profundo

Perceptrón

Profesores:

Berenice & Ricardo Montalvo Lezama

Febrero 2021

Contenido basado en el curso de AP del Dr. Gibran Fuentes Pineda del PCIC

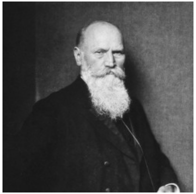
Breve historia

- 1871 Joseph von Gerlach propone la teoría reticular.
- 1871 Camillo Golgi propone el método de la tintura.
- 1888 Santiago Ramón y Cajal propone la doctrina de la neurona.
- 1891 H. Waldeyer-Hartz y otros consolidan la doctrina de la neurona.
- 1906 Nobel de medicina a Golgi y Cajal.
- 1950s Se verifica la doctrina de la neurona con el microscopio electrónico.
- Deep Learning in Neural Networks: An Overview, Schmidhuber, 2014.

Joseph von Gerlach

Camillo Golgi

Santiago Cajal



H. Waldeyer-Hartz

Imágenes tomadas de la Wikipedia.

Neurona biológica

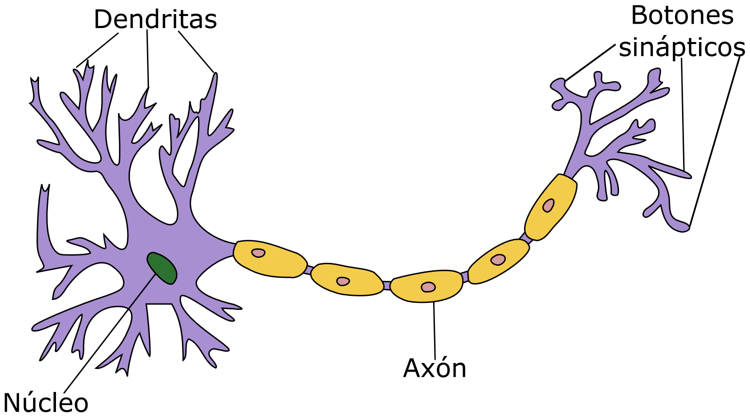
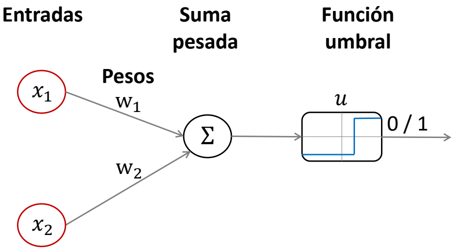


Imagen del usuario Quasar de Wikipedia, traducida al español (CC BY-SA 3.0)

Neurona artificial

- Unidad de umbral lineal propuesta por Warren McCulloch y Walter Pitts en 1943.

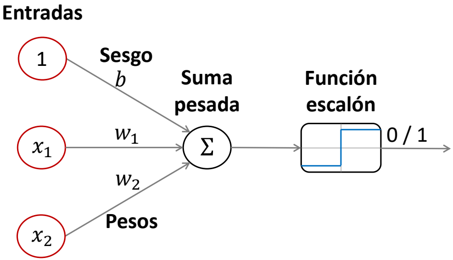


- Elementos básicos
  1. Pesos sinápticos
  2. Estímulo cumulativo
  3. Todo o nada (activación)

$$\hat{y} = a = \phi(b + \sum_{i=1}^{m} w_i \cdot x_i)$$
$$= \phi(b + \mathrm{w}^\top \cdot \mathrm{x})$$

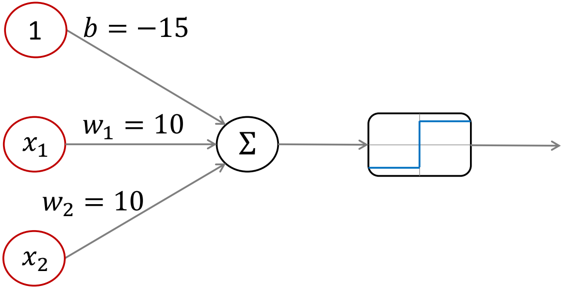
Forma general de neurona artificial

- En la forma general el sesgo es el peso de una entrada con valor 1.



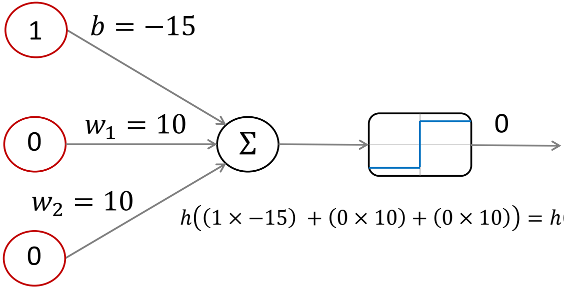
- Elementos básicos
  1. Pesos sinápticos
  2. Estímulo cumulativo
  3. Todo o nada (activación)

$$\hat{y} = a = \phi(b + \sum_{i=1}^{m} w_i \cdot x_i)$$
$$= \phi(b + \mathrm{w}^\top \cdot \mathrm{x})$$



x_1	x_2	AND (∧)
0	0	0
0	1	0
1	0	0
1	1	1

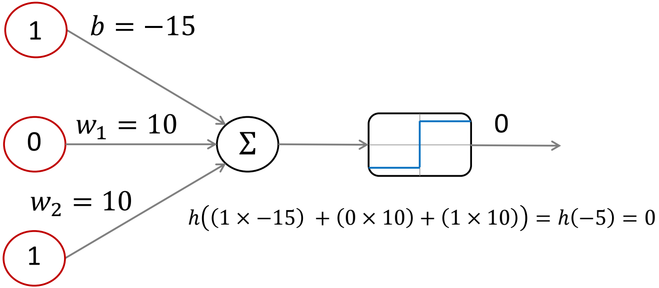
Compuerta AND (∧)



$$h\big((1 \times -15) + (0 \times 10) + (0 \times 10)\big) = h(-15) = 0$$

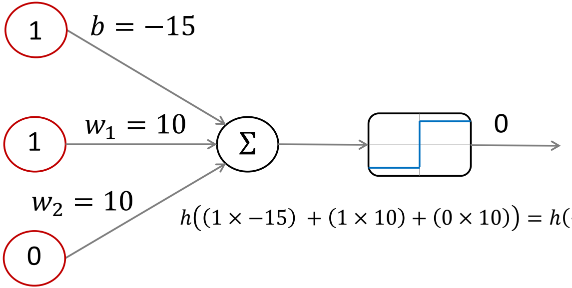
x_1	x_2	AND (∧)
0	**0**	**0**
0	1	0
1	0	0
1	1	1

Compuerta AND ($\wedge$)



x_1	x_2	AND ($\wedge$)
0	0	0
0	**1**	**0**
1	0	0
1	1	1

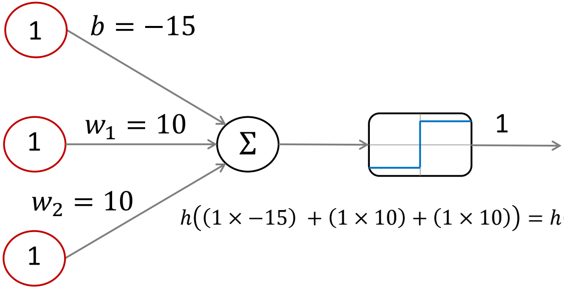
Compuerta AND ($\wedge$)



$$h\big((1 \times -15) + (1 \times 10) + (0 \times 10)\big) = h(-5) = 0$$

x_1	x_2	AND ($\wedge$)
0	0	0
0	1	0
1	**0**	**0**
1	1	1

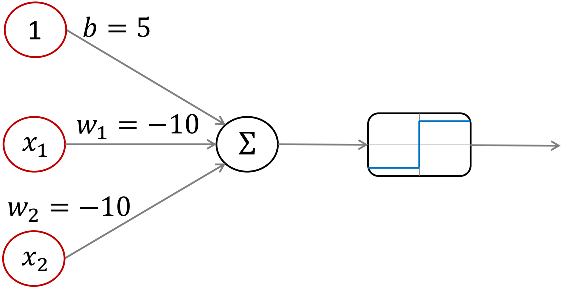
Compuerta AND ($\wedge$)



$$h\big((1 \times -15) + (1 \times 10) + (1 \times 10)\big) = h(5) = 1$$

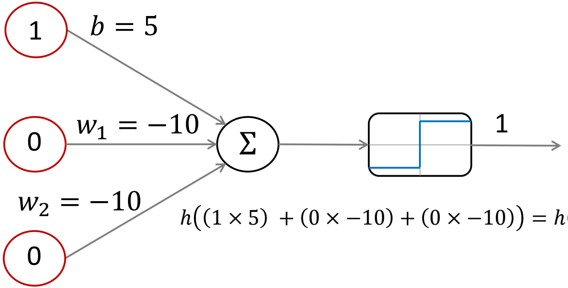
x_1	x_2	AND ($\wedge$)
0	0	0
0	1	0
1	0	0
1	**1**	**1**

Compuerta NOR (↓)



x_1	x_2	NOR (↓)
0	0	1
0	1	0
1	0	0
1	1	0

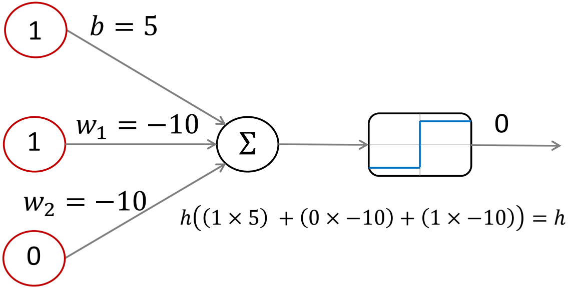
Compuerta NOR (↓)



$$h\big((1 \times 5) + (0 \times -10) + (0 \times -10)\big) = h(5) = 1$$

x_1	x_2	NOR (↓)
0	**0**	**1**
0	1	0
1	0	0
1	1	0

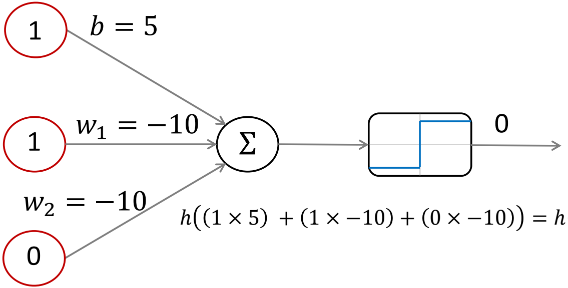
Compuerta NOR (↓)



$$h\big((1 \times 5) + (0 \times -10) + (1 \times -10)\big) = h(-5) = 0$$

x_1	x_2	NOR (↓)
0	0	1
0	**1**	**0**
1	0	0
1	1	0

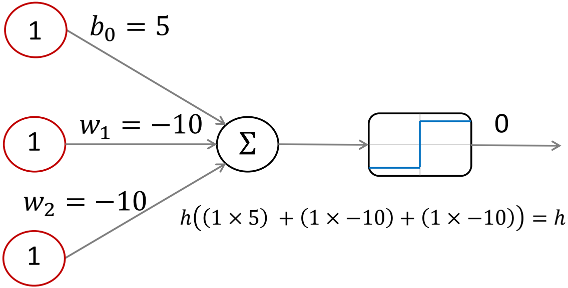
Compuerta NOR (↓)



$$h\big((1 \times 5) + (1 \times -10) + (0 \times -10)\big) = h(-5) = 0$$

x_1	x_2	NOR (↓)
0	0	1
0	1	0
1	**0**	**0**
1	1	0

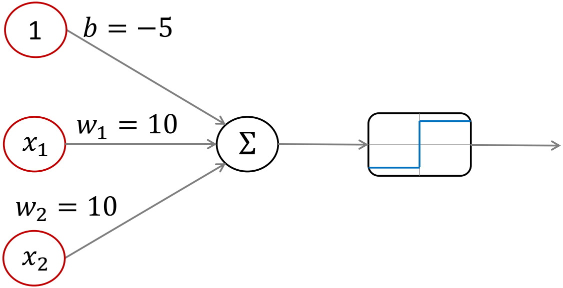
Compuerta NOR (↓)



$$h\big((1 \times 5) + (1 \times -10) + (1 \times -10)\big) = h(-15) = 0$$

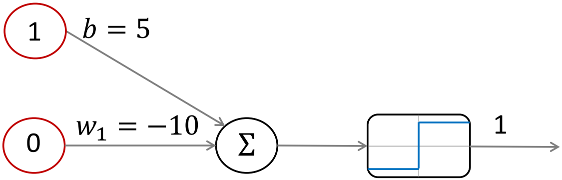
x_1	x_2	NOR (↓)
0	0	1
0	1	0
1	0	0
1	**1**	**0**

Compuerta NOT (¬)



x_1	NOT (¬)
0	1
1	0

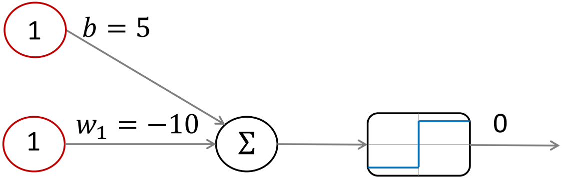
Compuerta NOT (¬)



$$h\big((1 \times 5)\ + (0 \times -10)\big) = h(5) = 1$$

x_1	NOT (¬)
0	**1**
1	0

Compuerta NOT (¬)



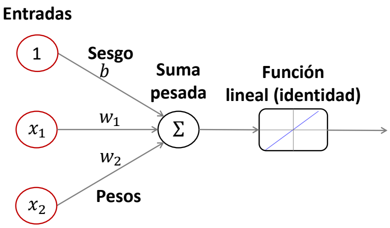
$$h\big((1 \times 5)\ + (1 \times -10)\big) = h(-5) = 0$$

x_1	NOT (¬)
0	1
1	**0**

Algoritmo del perceptrón

- Propuesto por Frank Rosenblatt en 1943[1], analizado por Minsky y Papert en 1969[2].
  1. Inicializar pesos y sesgo con ceros o un número aleatorio pequeño.
  2. Para cada ejemplo en el conjunto de entrenamiento:
     2.1 calcular la salida
     $$\hat{y}^{(i)} = \phi(\mathrm{w}[t]^\top \mathrm{x}^{(i)} + b)$$
     2.2 actualizar cada peso $w_j, j = 1, \ldots, d$ y el sesgo b
     $$\begin{aligned} w_j[t+1] =& w_j[t] + (y^{(i)} - \hat{y}^{(i)}) \cdot x_j^{(i)} \\ b[t+1] =& b[t] + (y^{(i)} - \hat{y}^{(i)}) \end{aligned}$$
  3. Repetir hasta converger o hayan pasado un número de épocas[3].
- Análisis de convergencia por Collins[4].

[1] The perceptron: A probabilistic model for information storage and organization in the brain, Rosenblat, 1943.
[2] A Step toward the Understanding of Information Processes, Minsky and Papert, 1969.
[3] Una época es una vista completa del conjunto de entrenamiento por el algoritmo.
[4] Convergence Proof for the Perceptron Algorithm, Michael Collins.



$$id(z) = z$$

$$\frac{d\,id(z)}{dz} = 1$$

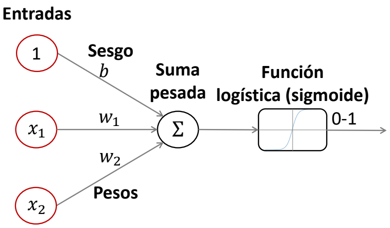
- Función de pérdida: error cuadráticos medio (ECM).

$$ECM(\mathrm{y}, \hat{\mathrm{y}}) = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)})^2$$

$$\frac{\partial ECM}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

$$\frac{\partial ECM}{\partial b} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)})$$

Neurona con activación sigmoide o logística



$$\sigma(z) = \frac{1}{1 + exp(-z)}$$

$$\frac{d\,\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))$$

- Función de pérdida: entropía cruzada binaria (ECB).

$$ECB(\mathrm{y}, \hat{\mathrm{y}}) = -\sum_{i=1}^{N} \left[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

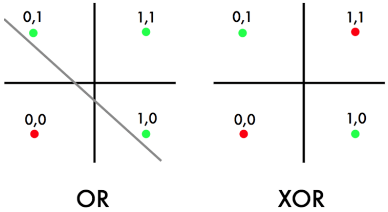
$$\frac{\partial ECB}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)}) \cdot x_j^{(i)}$$

$$\frac{\partial ECB}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}^{(i)} - y^{(i)})$$

Problemas no lineales

- ¿Cómo modelamos una computer XOR?

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0



- Minsky y Papert demostraron[5] (Necronomicón) que es imposible aprender la XOR con un perceptrón.

- Habemus el primer invierno de la IA.

[5] Perceptrons, Minsky y Papert, 1969.