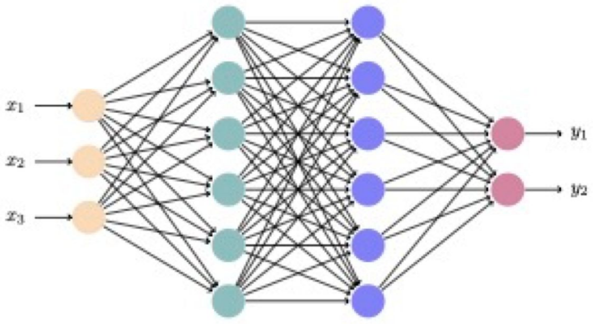
Fundamentos Matemáticos del Aprendizaje Profundo

1er cuat. 2025
Clase 5

Repaso



Una red neuronal codifica una función. Esta red se compone de

Neuronas

En esta clase estudiaremos el concepto de NEURONA ABSTRACTA

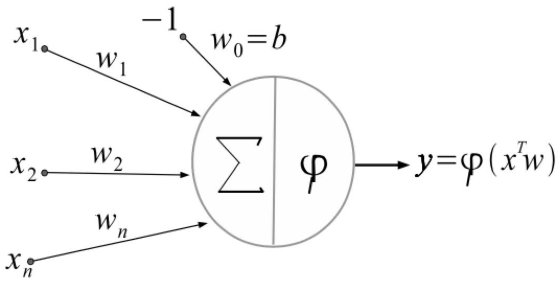
Repaso (2)

Una red precisa de *pesos* y *sesgos.*
Estos se inicializan aleatoriamente y luego se optimizan minimizando un *costo*

Algoritmos de minimización

- Descenso de gradiente (Gradient descent)
- Descenso estocástico (Stochastic gradient descent)
- Otros: Line search method, Adaptive gradient (AdaGrad), Root Mean Square Propagation (RMSProp), Adam, AdaMax, etc (ver [Calin], cap. 4)

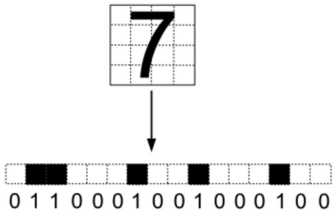
Neurona abstracta



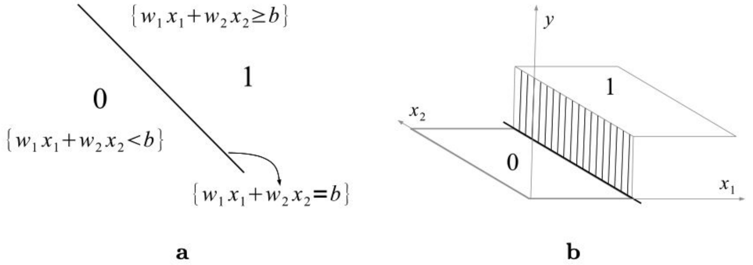
DEFINICIÓN 5.1.1. Una neurona abstracta es un cuádruple $(\mathbf{x}, \mathbf{w}, \varphi, y)$, donde $\mathbf{x} = (x_0, \ldots, x_n)$ es el vector de entrada, $\mathbf{w} = (w_0, \ldots, w_n)$ es el vector de pesos, con $x_0 = -1$ y $w_0 = b$, el sesgo, y φ es la función de activación que define la función de salida $y = \varphi(\mathbf{w} \cdot \mathbf{x}) = \varphi(\sum_{i=0}^{n} w_i x_i)$.

Tipos de datos:

- *binario* si $x_i \in \{0, 1\}$ para $i = 1, \ldots, n$;
- *signo* si $x_i \in \{-1, 1\}$ para $i = 1, \ldots, n$;
- *dígito* si $x_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ para $i = 1, \ldots, n$;
- *números reales arbitrarios* si $x_i \in \mathbb{R}$ para $i = 1, \ldots, n$;
- *números reales acotados* si $x_i \in [a, b]$ para $i = 1, \ldots, n$.



Tipos de neuronas: Perceptrón



Tipos de datos: binario $x_i \in \{0, 1\}$

Función de activación: Heaviside $$\varphi(x) = \begin{cases} 1, & \text{si } x < 0 \\ 0, & \text{si } x \geq 0. \end{cases}$$

Salida: *puerta umbral* $$y = \varphi(\mathbf{w} \cdot \mathbf{x}) = \begin{cases} 1, & \text{si } \sum_{i=1}^{n} w_i x_i < b \\ 0, & \text{si } \sum_{i=1}^{n} w_i x_i \geq b. \end{cases}$$

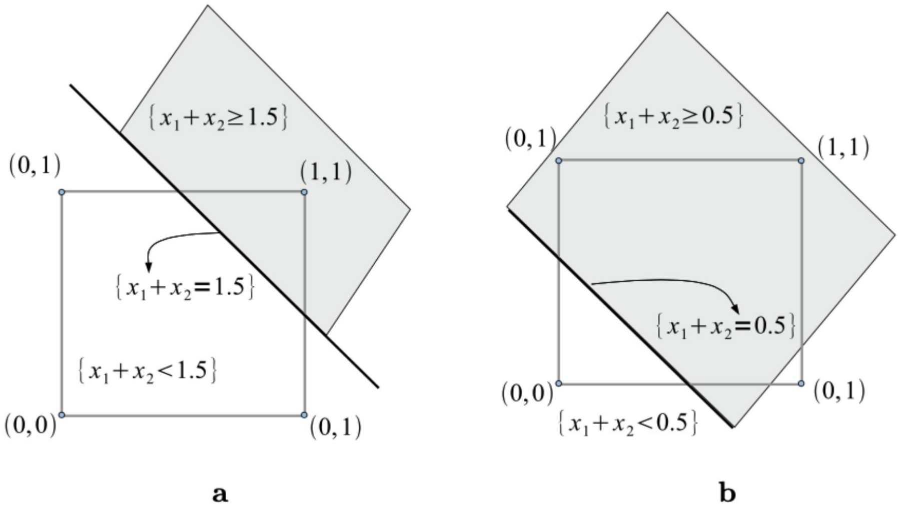
Implementando AND y OR

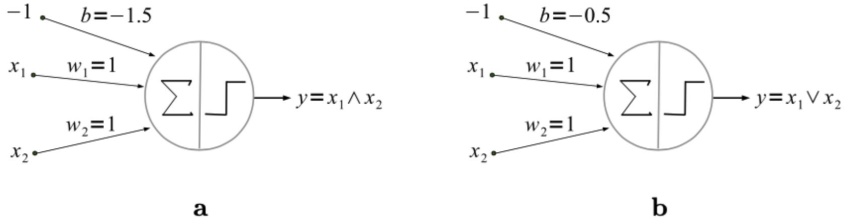
AND:

x_1	x_2	$y = x_1 \land x_2$
0	0	0
0	1	0
1	0	0
1	1	1

OR:

x_1	x_2	$y = x_1 \lor x_2$
0	0	0
0	1	1
1	0	1
1	1	1

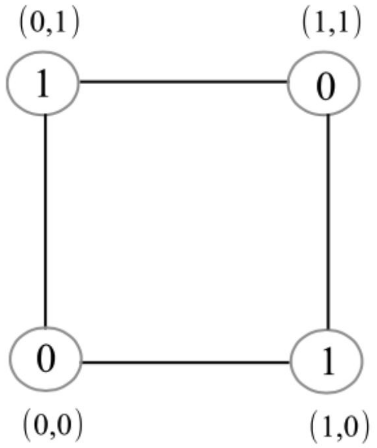




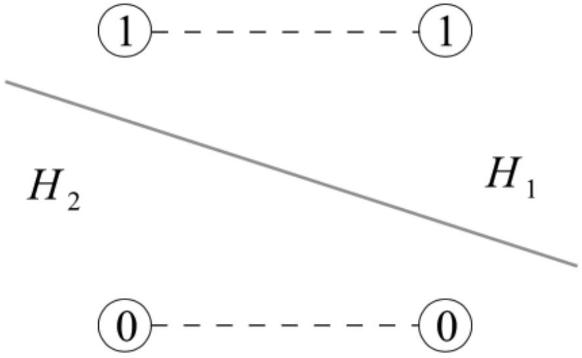
No se puede implementar XOR

XOR:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

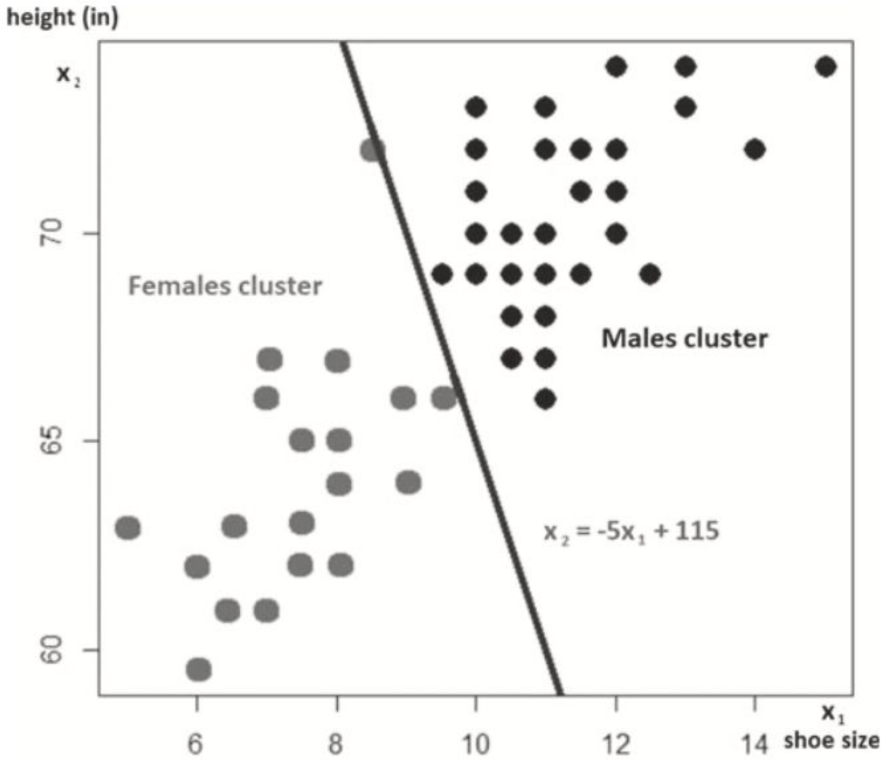


a

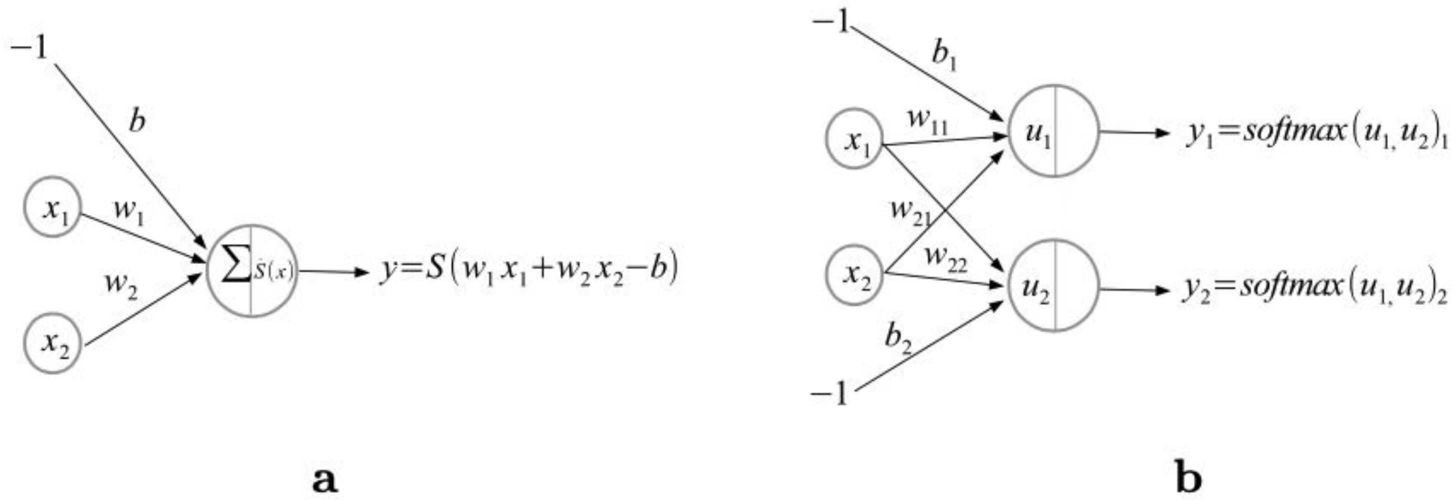


b

División de clústeres



División de clústeres (2)



$$(y_1, y_2) = \mathrm{softmax}(u_1, u_2) = \left(\frac{e^{u_1}}{e^{u_1} + e^{u_2}}, \frac{e^{u_2}}{e^{u_1} + e^{u_2}} \right)$$

Tipos de neurona: Neurona sigmoide

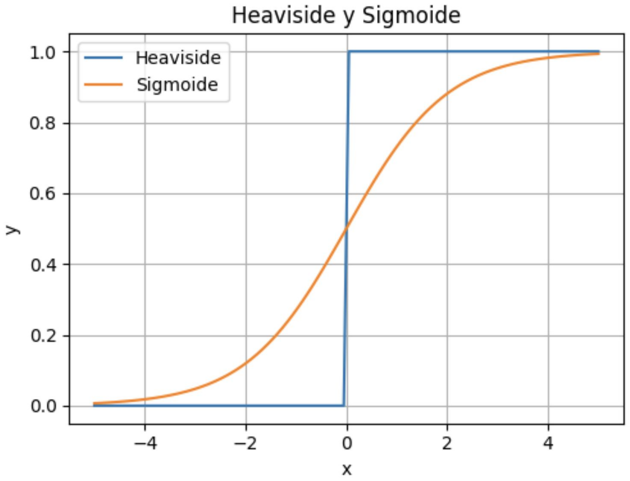
Tipo de dato: número real

Función de activación: sigmoide

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

Salida: número entre 0 y 1

$$y = \sigma(\mathbf{w} \cdot \mathbf{x} - b) = \frac{1}{1+e^{-\mathbf{w}\cdot\mathbf{x}+b}}$$



Ejemplo: Probabilidad de default de una compañía

Queremos predecir la probabilidad de que una compañía entre en default en un período de tiempo T. Vamos a usar una neurona sigmoide para esto.

Entrada de la neurona: $\mathbf{x} \in \mathbb{R}^m$ (reservas, ingresos, costos, gastos, etc)

Conjunto de entrenamiento: $\{(\mathbf{x}_i, z_i)\}_{1 \leq i \leq n}, \ z_i = \pm 1$

Tenemos (X, Z) v.a., donde Z toma valores 1 y -1. La probabilidad condicional P(Z|X) viene dada por

z	-1	1
$\mathbb{P}(z\|\mathbf{x})$	$1 - h(\mathbf{x})$	$h(\mathbf{x})$

Elegimos $h \colon \mathbb{R}^m \to [0,1],\ h(\mathbf{x}) = \sigma(\mathbf{w}\cdot\mathbf{x})$

$\mathbf{w}\cdot\mathbf{x}$ = puntaje por defecto de la compañía

Como $\sigma(-x) = 1 - \sigma(x)$ tenemos que $\mathbb{P}(z|\mathbf{x}) = \sigma(z\mathbf{w}\cdot\mathbf{x})$

El peso óptimo es el de *máxima verosimilitud*

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \mathbb{P}(z_1,\ldots,z_n|\mathbf{x}_1,\ldots,\mathbf{x}_n)$$

Asumiendo independencia…

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-z_i \mathbf{w}\cdot\mathbf{x}_i})$$

Descenso de gradiente

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-z_i \mathbf{w}\cdot\mathbf{x}_i})$$

$$\nabla F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} z_i \mathbf{x}_i(\sigma(z_i \mathbf{w}\cdot\mathbf{x}_i) - 1) = -\frac{1}{n}\sum_{i=1}^{n} z_i \mathbf{x}_i \sigma(-z_i \mathbf{w}\cdot\mathbf{x}_i)$$

$$\mathbf{w}_{j+1} = \mathbf{w}_j - \eta\nabla F(\mathbf{w}_j) = \mathbf{w}_j + \frac{\eta}{n}\sum_{i=1}^{n} z_i \mathbf{x}_i \sigma(-z_i \mathbf{w}_j\cdot\mathbf{x}_i)$$

$\eta > 0$ tasa de aprendizaje