

Ayudantía 9: Redes Neuronales

Bernardita Morris
José Domínguez

Características Generales

¿Qué son las redes neuronales?

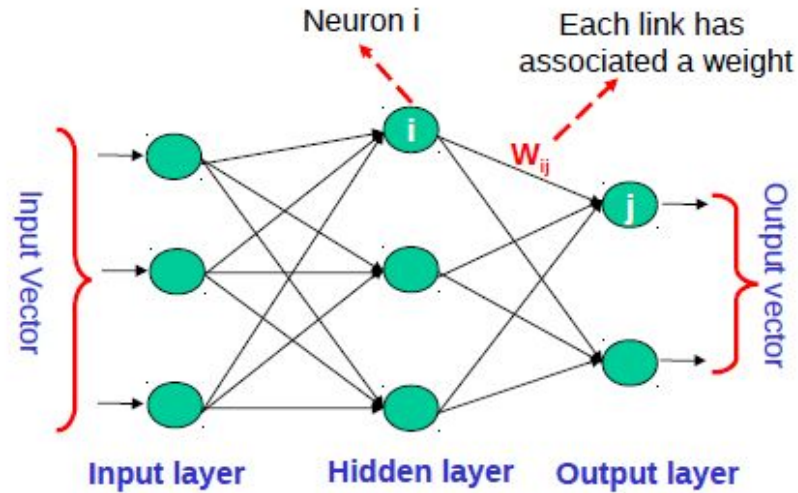
- Es una técnica de Machine Learning
- Su nombre y estructura están inspirados en el cerebro humano

¿Para qué se usan las redes neuronales?

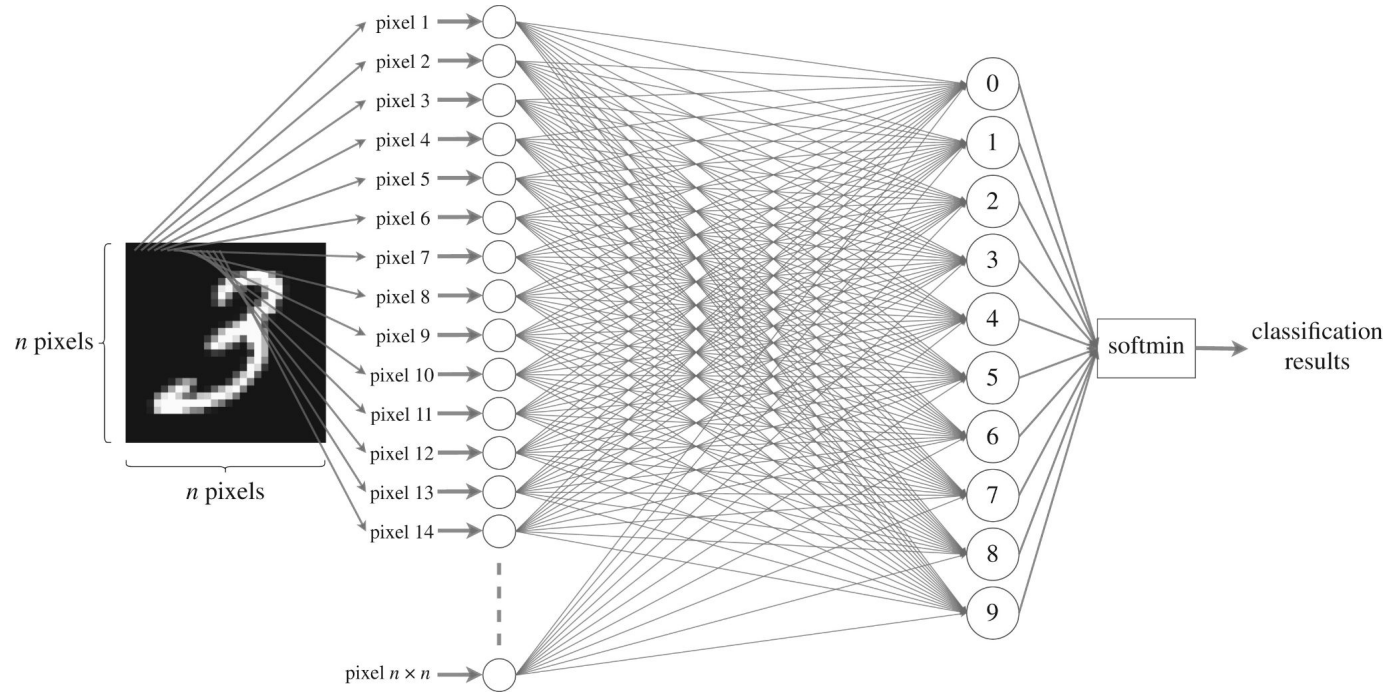
- Para clasificaciones y regresiones
- Se usan para aproximar funciones complejas

Características Generales

¿Que compone a una red Neuronal?



Idea general del funcionamiento



¿Cómo funcionan las redes Neuronales?

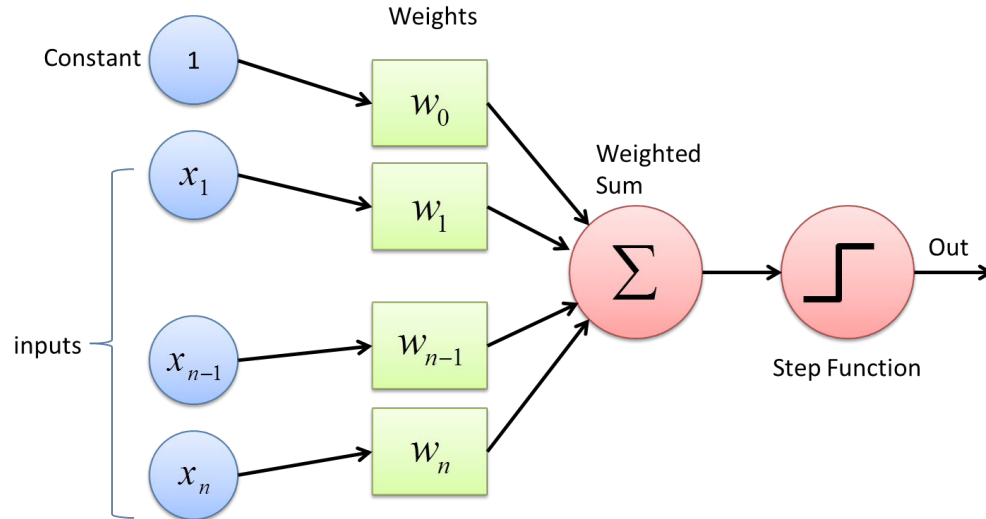
Para esto veremos primero la red más básica de todas

- El perceptrón

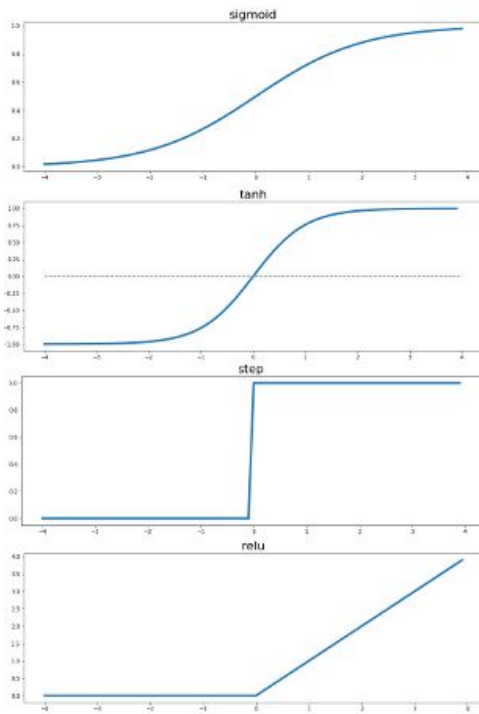
¿Cómo funcionan las redes Neuronales?

Para esto veremos primero la red más básica de todas

- El perceptrón



Función Activación



Activation Functions

Sigmoid

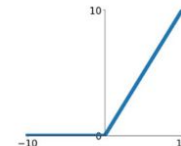
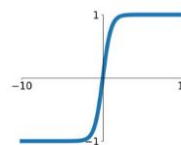
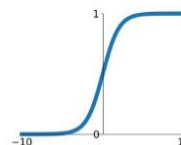
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

tanh

$$\tanh(x)$$

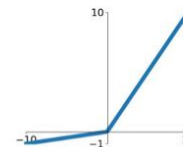
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

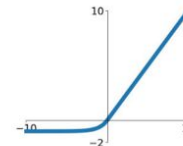


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



¿Cómo funcionan las redes Neuronales?

$$\sum_{i=1}^m w_i x_i + \textit{bias} = w_1 x_1 + w_2 x_2 + w_3 x_3 + \textit{bias}$$

$$\text{output} = f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0 & \text{if } \sum w_1 x_1 + b < 0 \end{cases}$$

Ejemplo: ¿Salir o no salir?

bias: 3

x1: ¿Tengo prueba mañana? (si : 1, no: 0)

x2: ¿Dormí lo suficiente anoche? (si : 1, no: 0)

x3: ¿Es viernes? (si : 1, no: 0)

w1: -10

w2: -1

w3: 6

$$y = (0 \cdot -5) + (1 \cdot -1) + (1 \cdot 6) + 3 = 8$$

función activación: $f(8) = 1$

resultado: Hoy se Sale!

Ejemplo ¿salir o no salir?

- Qué pasa si ahora tenemos más factores que afectan nuestra decisión?
- Qué pasa si no sabemos cuáles son los pesos al momento de tomar esta decisión
 - Qué es más importante nuestro estado de ánimo, cuanto dormimos, que nos dijo un amigo, etc.
- Como sabemos como estos distintos factores interactúan entre sí

Prueba	Dormir	Ánimo	...	viernes	¿Salir?
1	0.3345	15.34		1	si
0	1.564	23.5		0	no
1	3.6255	-4.3		0	no
...
1	3.212	40.123		1	no

Redes Neuronales

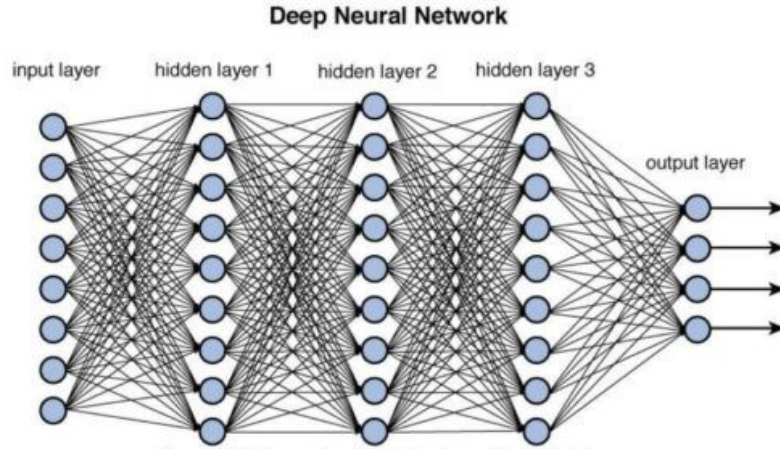
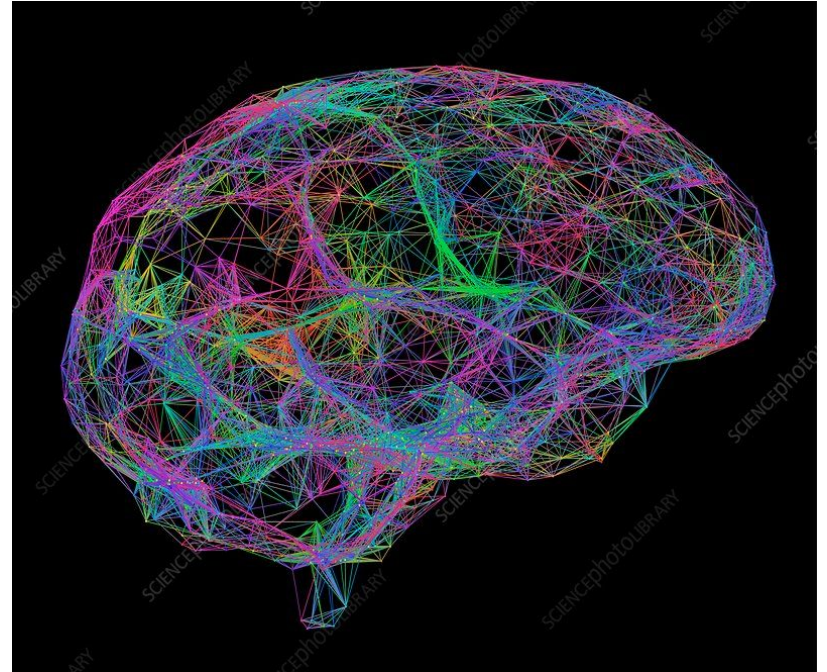


Figure 12.2 Deep network architecture with multiple layers.



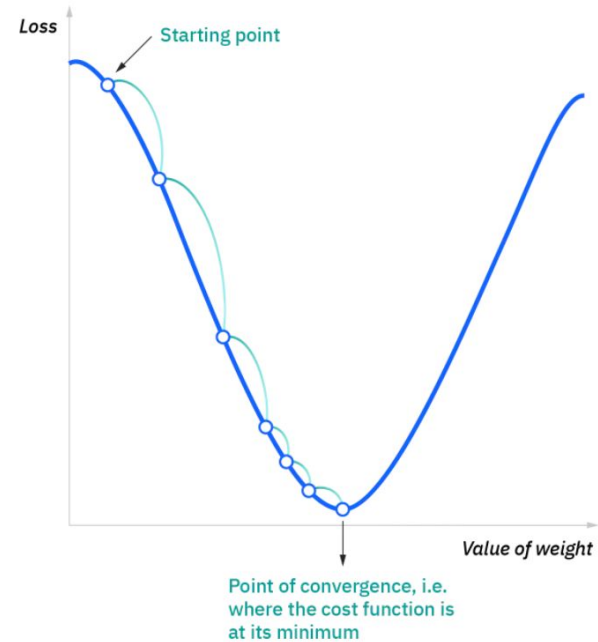
Redes neuronales

In Binary Classification problem

$$\text{If } y = 0 \quad \text{Loss} = -\frac{1}{N} \sum_{n=1}^N \left[\log(1 - \hat{y}_n) \right]$$

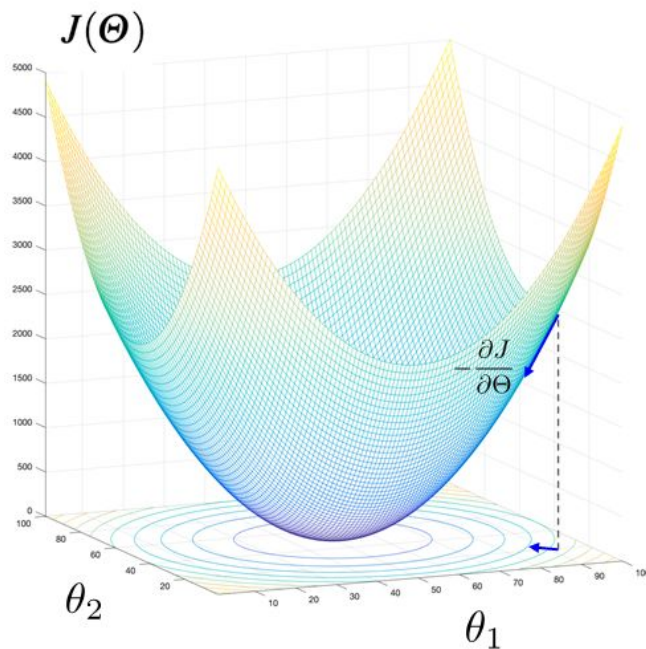
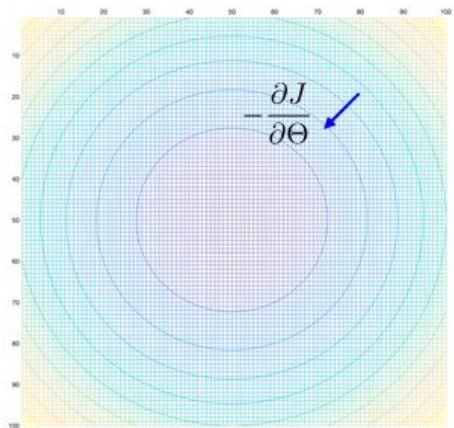
$$\text{If } y = 1 \quad \text{Loss} = -\frac{1}{N} \sum_{n=1}^N \left[y_n \log \hat{y} \right]$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_{\text{true}} - y_{\text{predicted}})^2$$

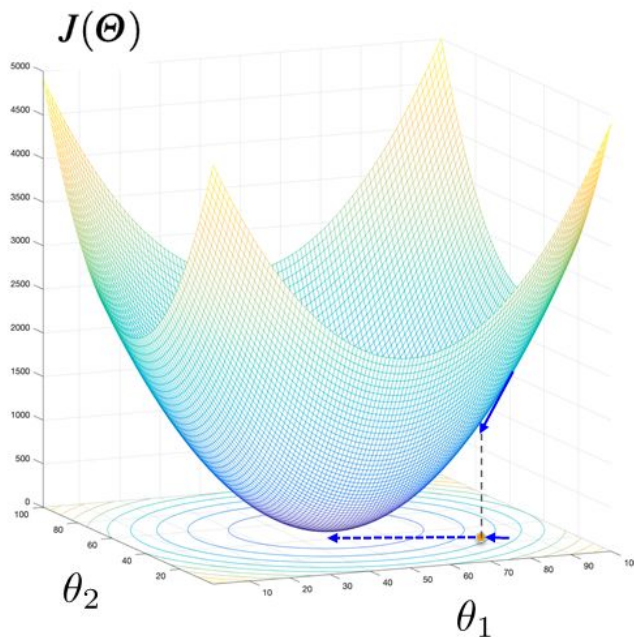
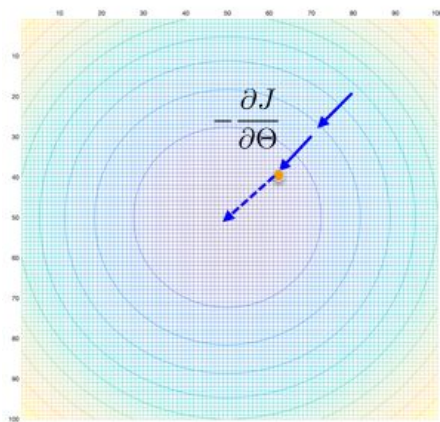


Entrenamiento de NN: Método del gradiente

- Definir parámetros (pesos y bias) de forma que se minimice el error



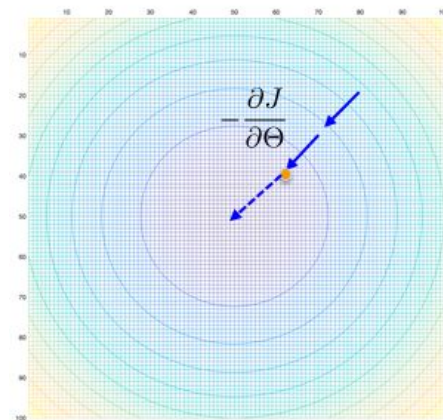
Entrenamiento de NN: Método del gradiente



Entrenamiento de NN: Método del gradiente

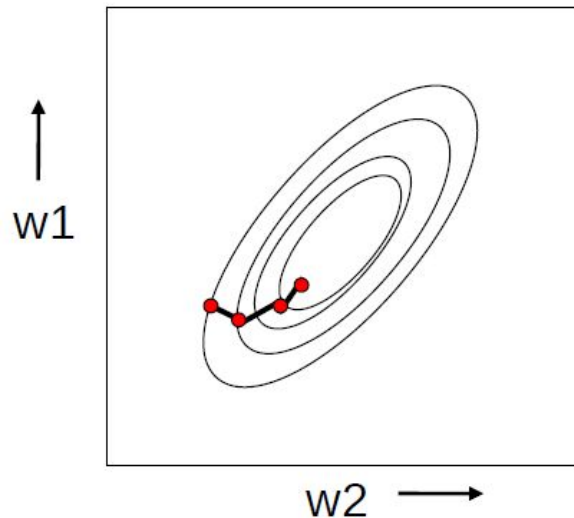
- Pasos se van definiendo según un learning rate η

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$



Entrenamiento de NN: Batch mode

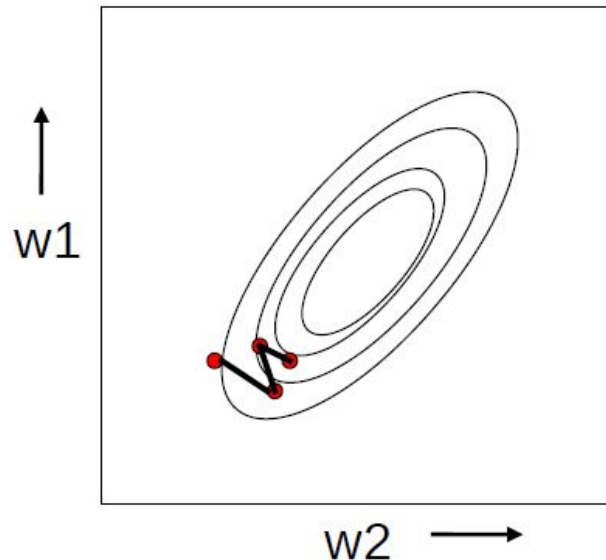
- Se usan todos los datos de entrenamiento
- **Ventaja:** permite hacer estimaciones más precisas del gradiente al tener mayor cantidad de datos
- **Desventaja:** tiempo de cómputo mayor



Entrenamiento de NN: Incremental mode

- Se usa un subconjunto de los datos de entrenamiento para estimar el gradiente
- **Ventaja:** permite realizar los cálculos más rápido (particularmente ventajoso para redes muy grandes)
- **Desventaja:** estimaciones que realiza sobre el gradiente son menos precisas
- Stochastic gradient descent:
 - Selección de mini-batch de forma aleatoria

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



Backpropagation

- Permite ampliar la idea del perceptrón a redes multicapas
 - Nos permite entrenar redes con capas ocultas
- Idea: Red calcula output final y va propagando error hacia atrás en las capas
- Implementación de la regla de la cadena con derivadas por capas de la red

$$\frac{dNN}{dw} = \frac{dNN}{dLoss} \cdot \frac{dLoss}{dLayer_N} \cdot \frac{dLayer_N}{dLayer_{N-1}} \cdots \frac{dLayer_W}{dw}$$

¿Cómo detectamos overfitting?

