

Parte: 1

Teoría Deep Learning

Profesores: Javier López
Daniel Cano

Introducción

Motivación Biológica

- Se pretendía **modelar** las neuronas reales
- (hoy) Se sabe que las neuronas artificiales son aproximaciones extremadamente simples

Feed-Forward Neural Networks

- Conjunto de **unidades (Neuronas)**
- Conjunto de **conexiones dirigidas (Sinapsis)**

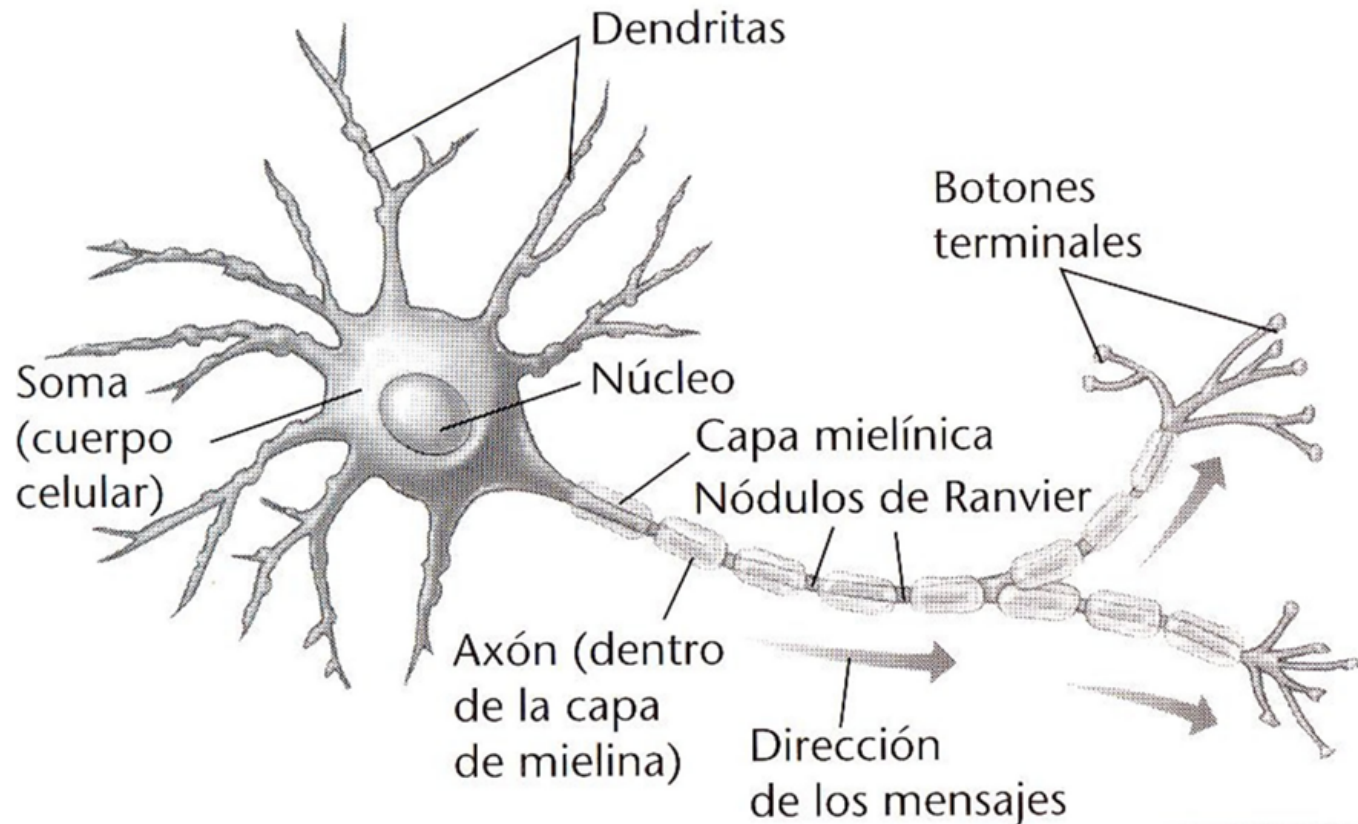
Historia

Hitos

- **1958** - Perceptron supervised learning model
- 1986 - Backpropagation & autoencoders
- 1989 - Convolutional neural network
- 1997 - Long Short-Term Memory
- **2006** - Deep Neural Networks
- 2014 - GANs

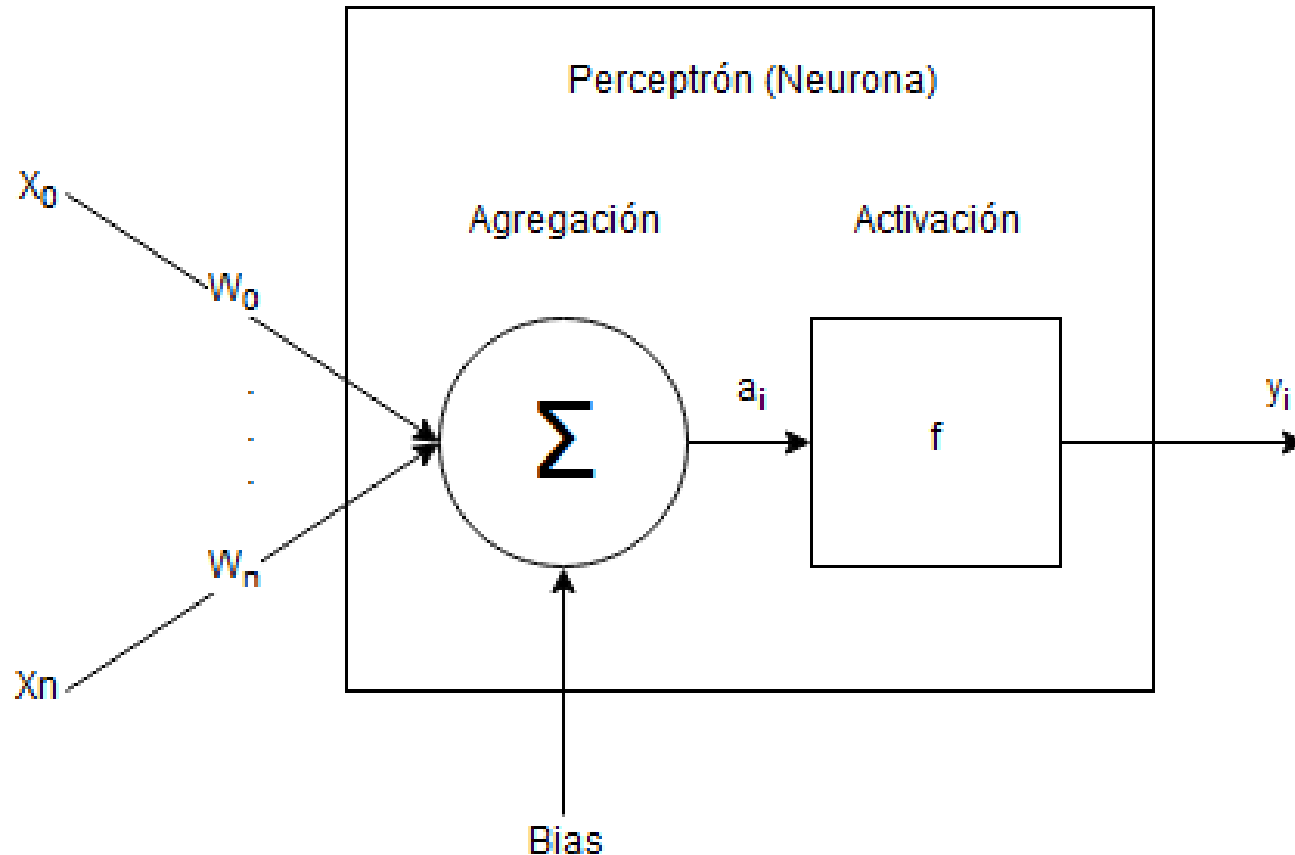
Neurona

Representación



Perceptrón

Representación



Perceptrón

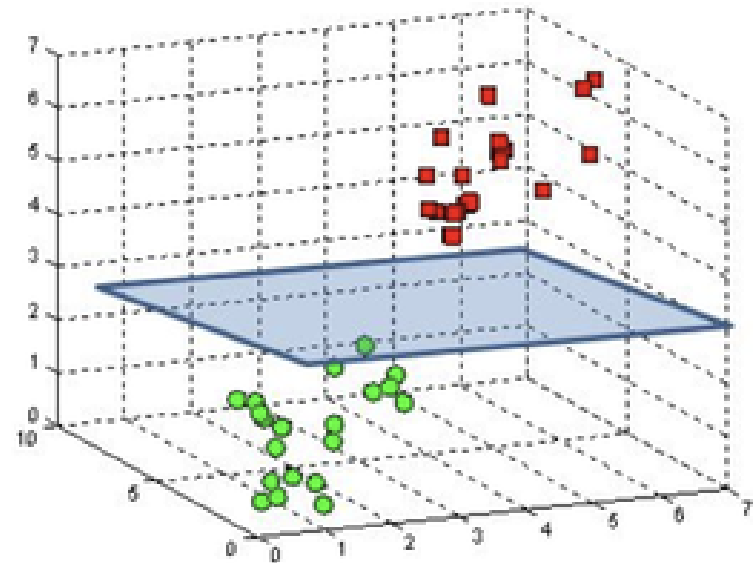
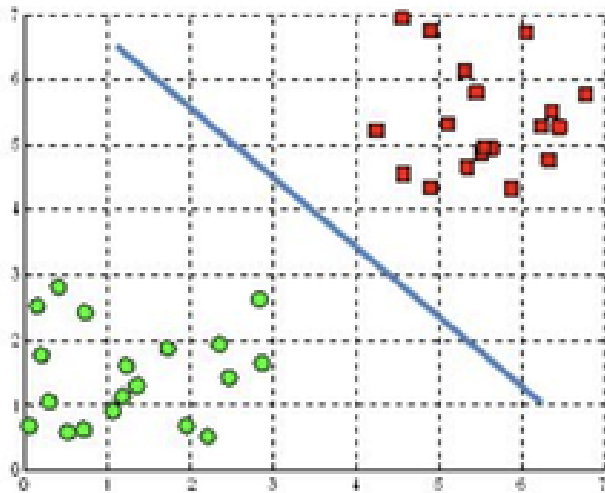
Ecuación

$$Y = f\left(\sum_{i=1}^n w_i x_i\right)$$

$$Y = X_n * W_n + \dots + X_1 * W_1 + W_0$$

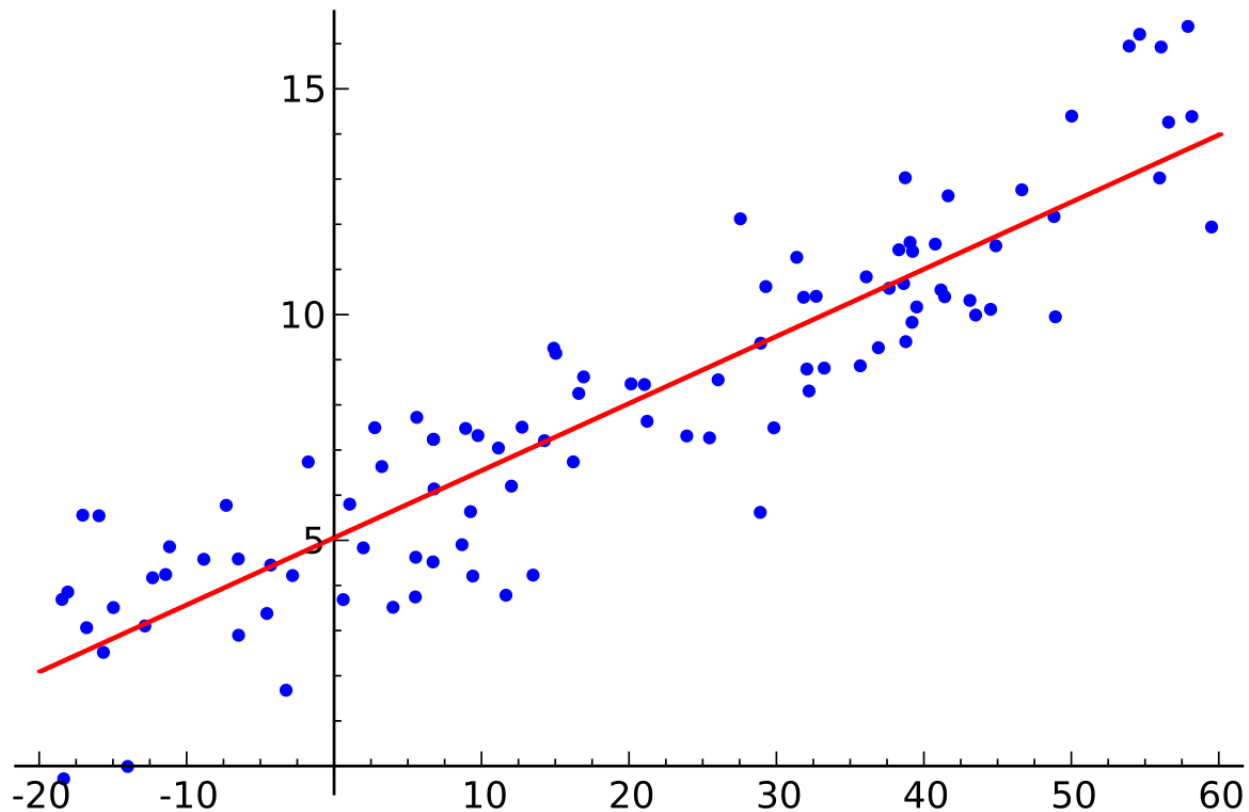
Perceptrón

Separar conjuntos



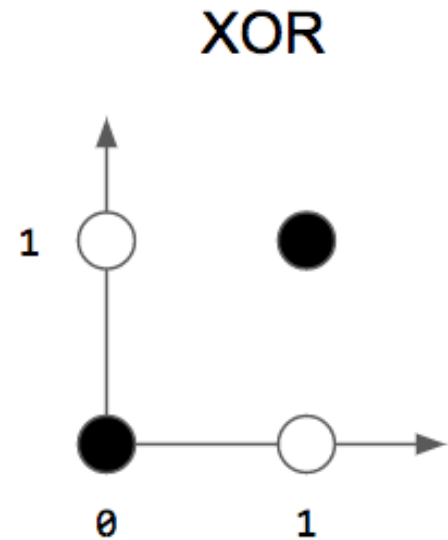
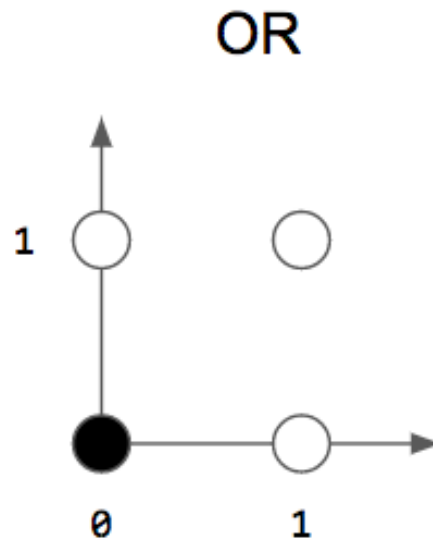
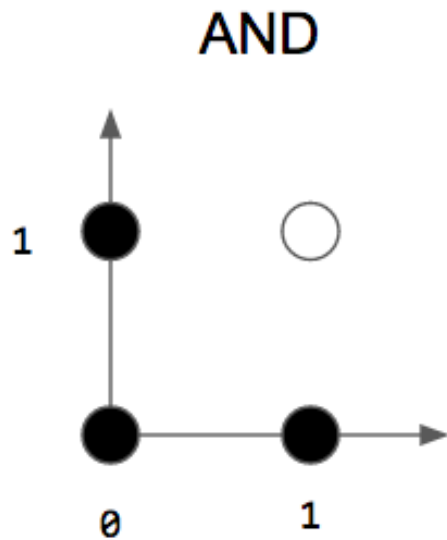
Perceptrón

Regresión Lineal



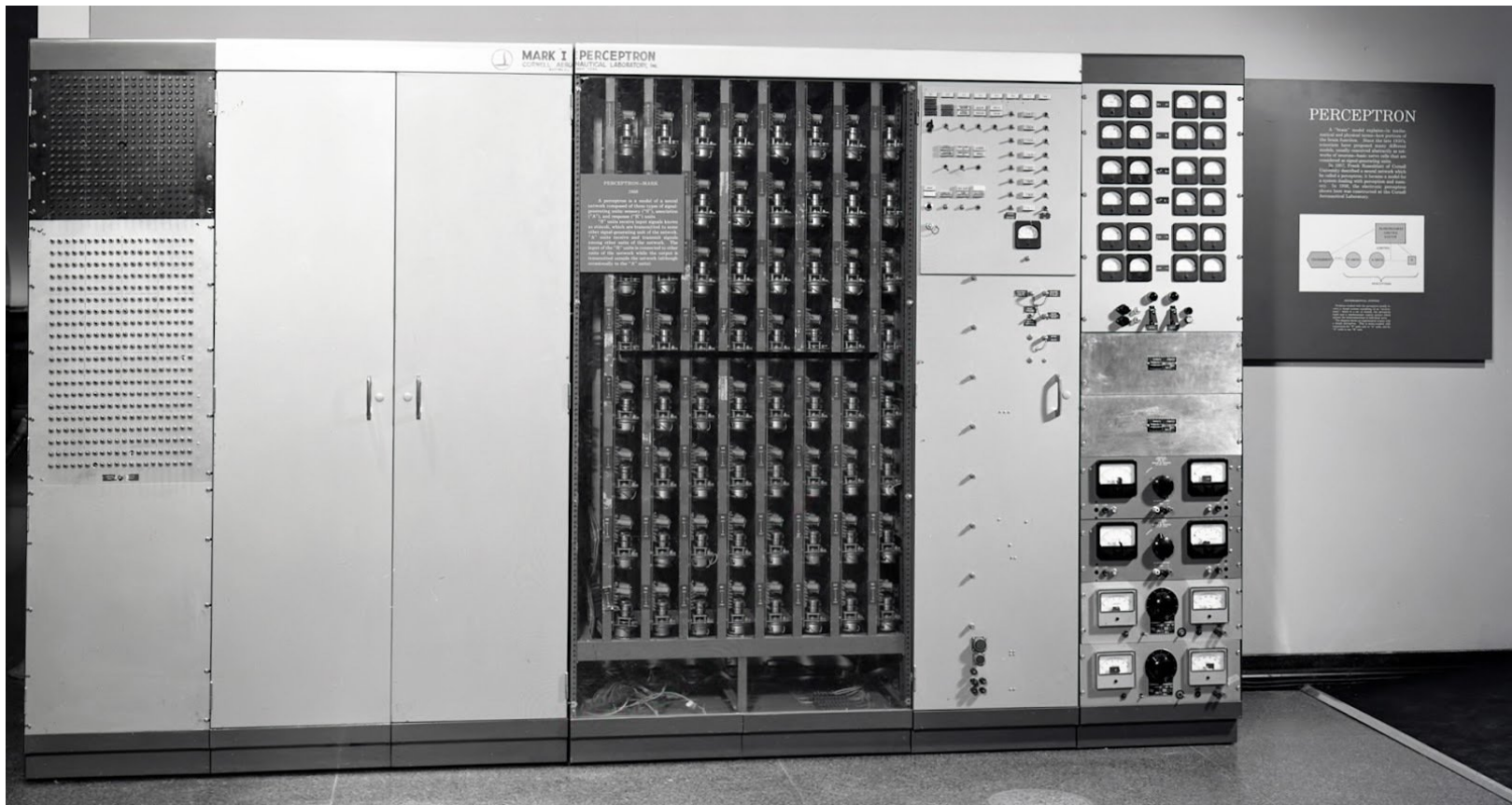
Perceptrón

Limitaciones



Perceptrón

Limitaciones



Perceptrón

Inicialización

- Random init (mejor en general) (dist. normal)
- 0 init (W en bias = 0)
- He init
- Xavier init

$$W^{[l]} = np.random.randn(size_l, size_l-1) * np.sqrt{\frac{2}{size^{[l-1]}}}$$

$$W^{[l]} = np.random.randn(size_l, size_l-1) * np.sqrt{\frac{1}{size^{[l-1]}}}$$

Perceptrón

Funciones de Activación

Identidad:

$$f(x) = x$$

Logística:

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

Rectifier Linear (ReLU):

$$f(x) = \max(0, x)$$

Tangente Hiperbólica:

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

Softmax:

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

*Muchas otras

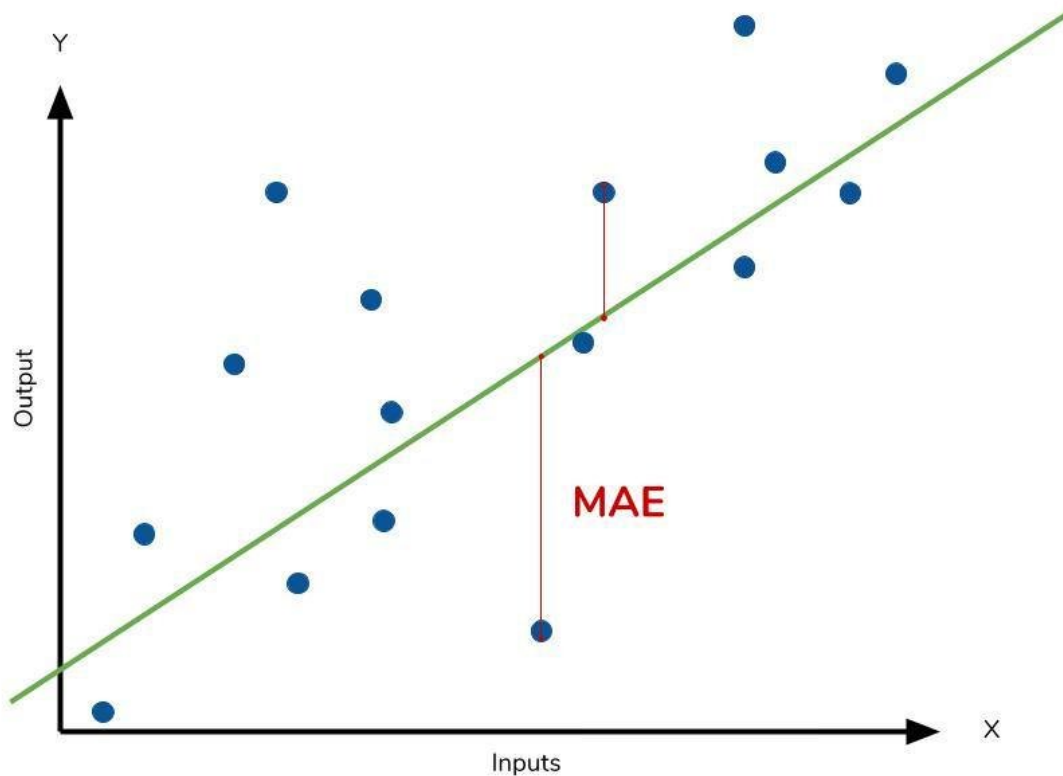
Perceptrón

Funciones de Coste

- Necesitamos evaluar el modelo
- Tiene que contar con ciertas propiedades
- Seleccionar adecuadamente para cada problema

Perceptrón

Noción de error



Manos a la obra

Notebooks

- Funciones de Activación
- Funciones de Coste

