



Ayudantía 13

Redes Neuronales

Por Sofía Hosiasson y

22 de Noviembre



Componentes principales

- Neuronas
- Capas
- Pesos
- Funciones de activación



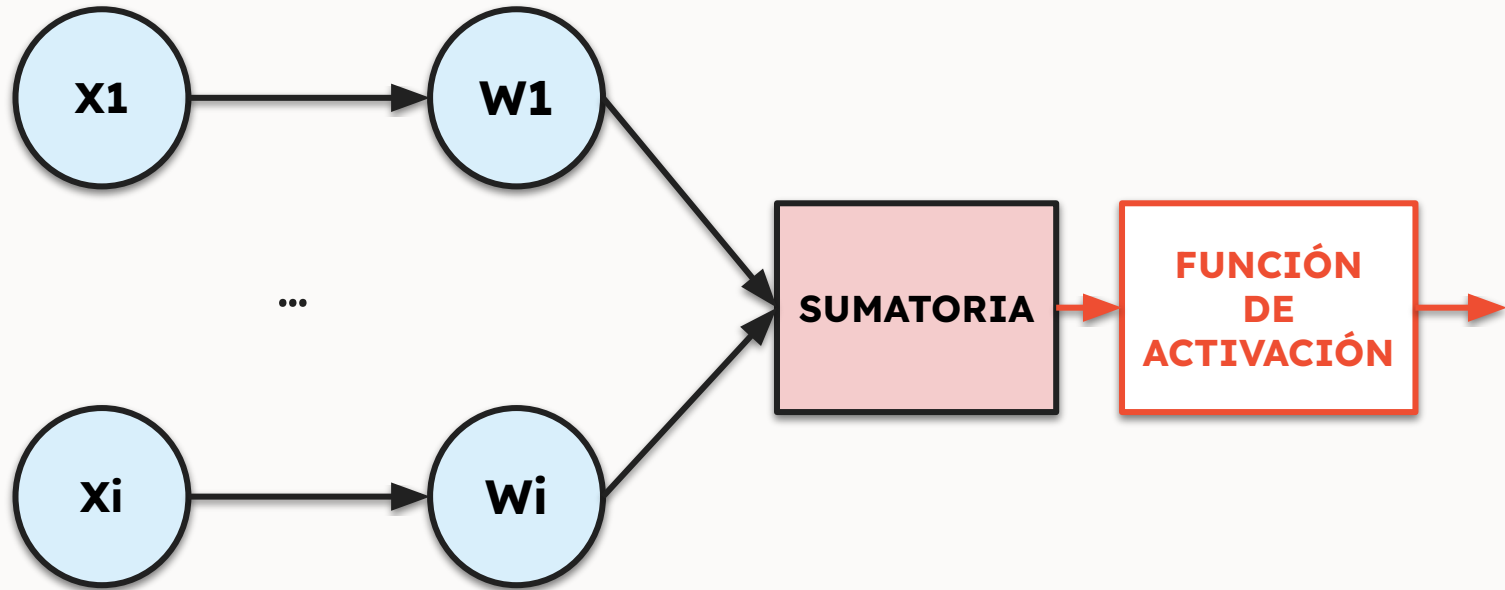


Ventajas y Desventajas

- Buenos para grandes volúmenes de datos
 - Son capaces de aprender transformaciones de los datos (embeddings), que luego pueden ser usadas como features en otros problemas.
 - Se puede usar para prácticamente cualquier problema
-
- Difíciles de entrenar
 - Interpretabilidad es compleja
 - problemas de sobre ajuste

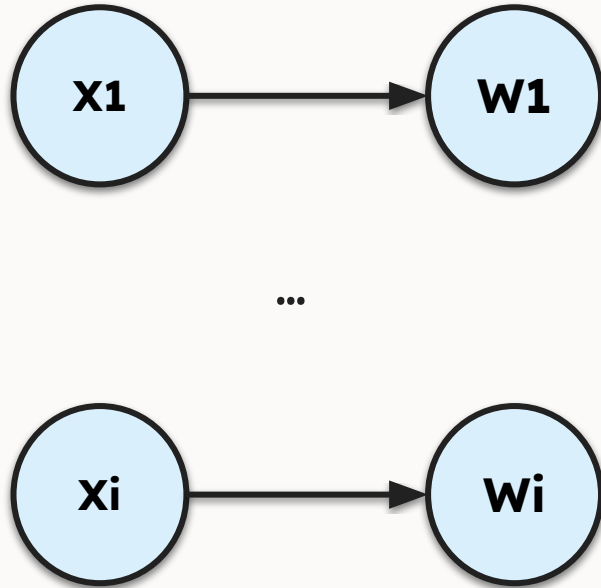


Perceptrón: la neurona básica





Perceptrón: la neurona básica

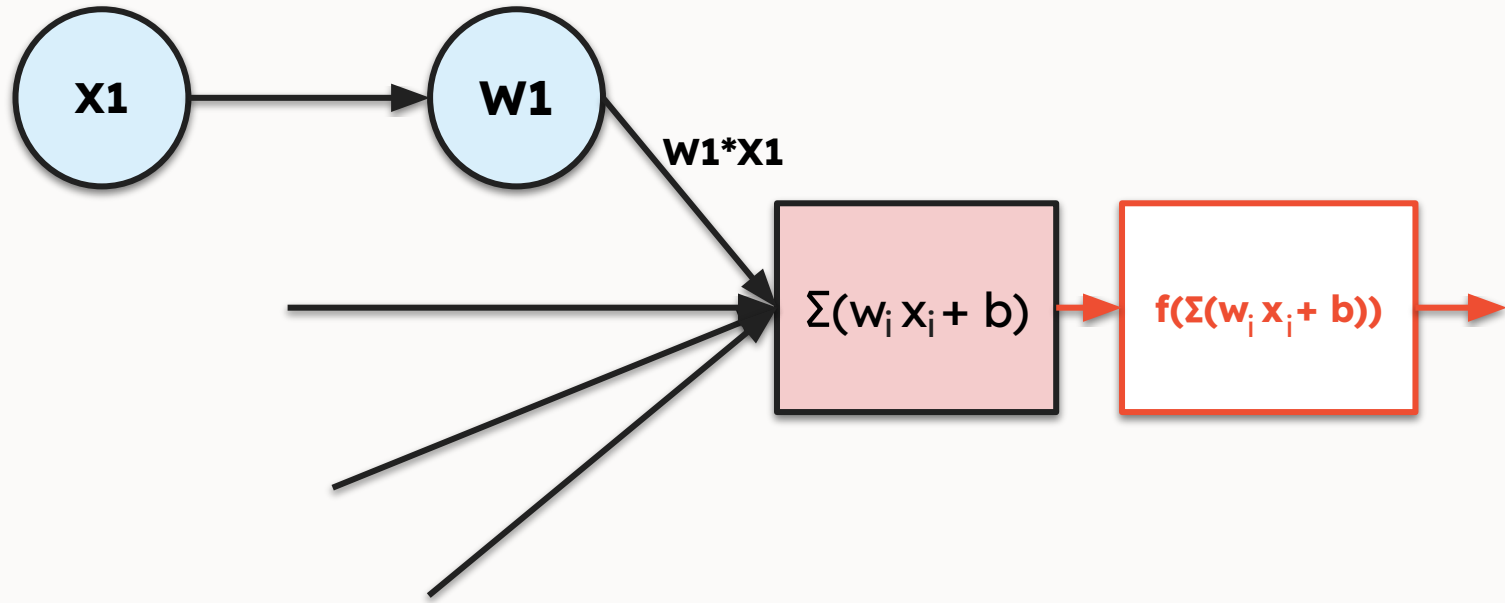


X : vector de características

W : vector de pesos



Perceptrón

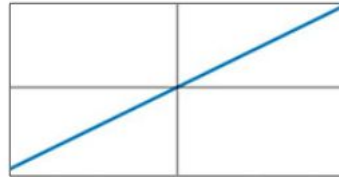




Función de Activación

Transmite la información generada por la combinación lineal de los pesos y las entradas

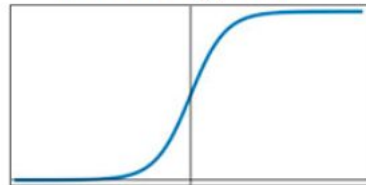
Función lineal



Función escalón



Función sigmoide



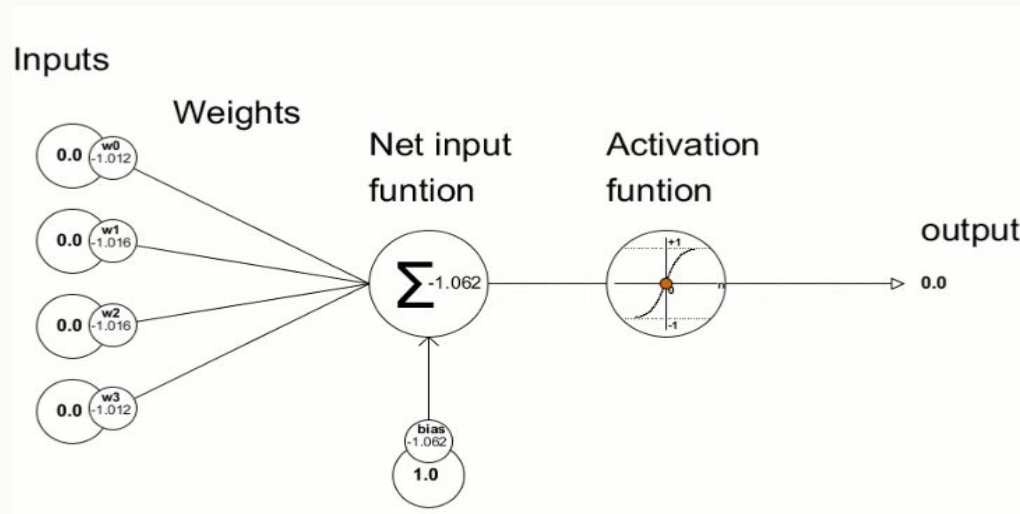
Función ReLU





Función de Costo (Loss)

- Diferencia lo real con lo predecido
- Partimos con parámetros al azar
- Se ajustan los pesos para minimizar la función costo

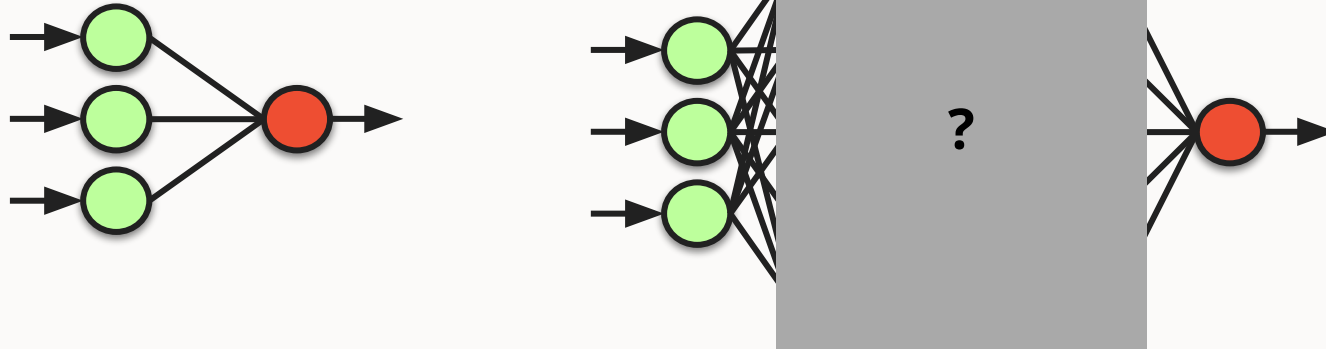


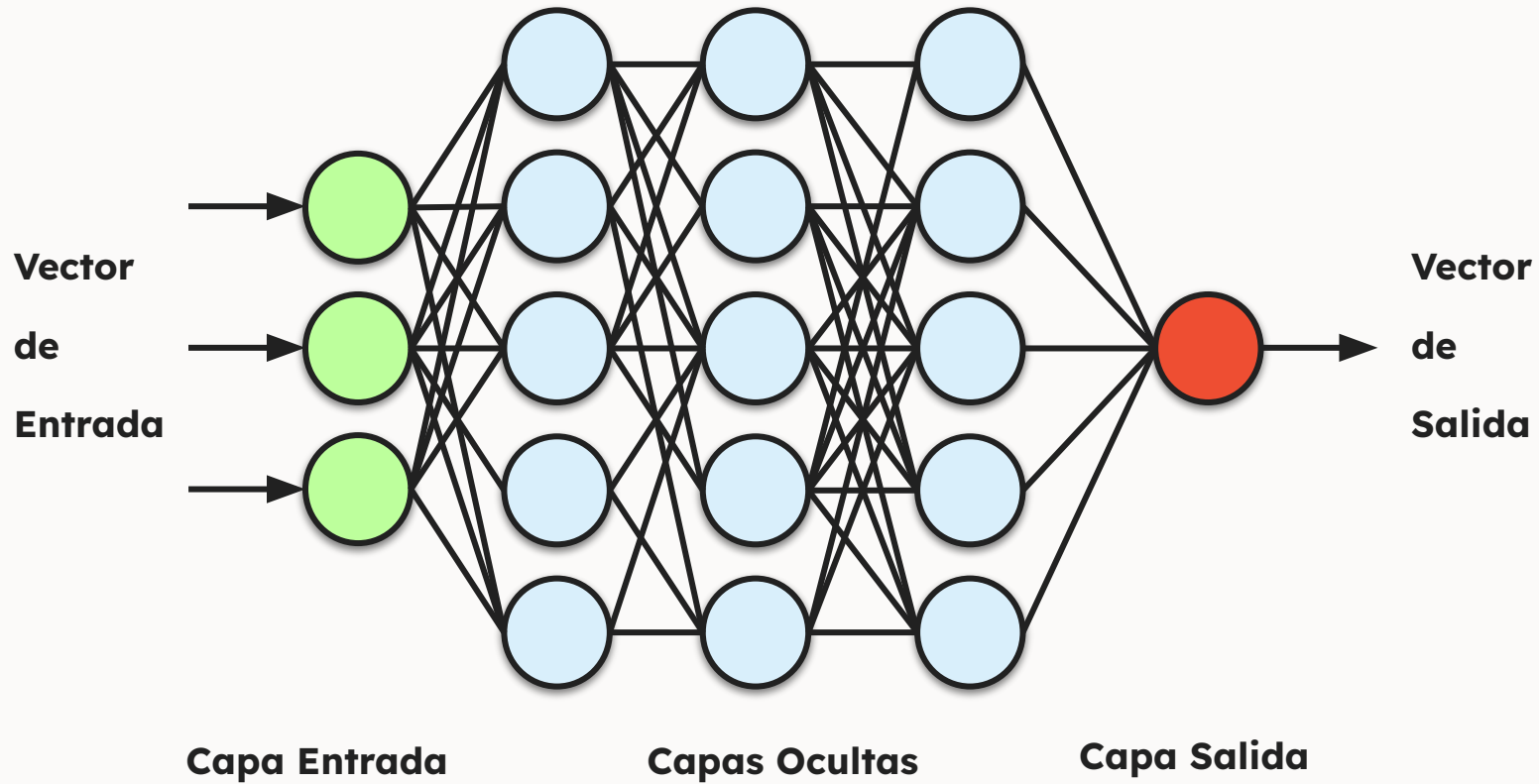


Red neuronal: Combinación de perceptrones

Perceptron puede resolver problemas lineales (clasificación y regresión)

Tiene dificultades para problemas no linealmente separables
-> Al tener capas ocultas podemos representar lo no lineal







Introducción a Keras

- Librería de alto nivel para Redes Neuronales
- Instalación:
 - Python
 - TensorFlow
 - Keras





Construcción de una red neuronal

- Definición del modelo
- Añadir capas al modelo
- Compilación del modelo
- Entrenamiento
- Evaluación
- Predicción





Ejemplo : MNIST

- Colección de base de datos que se utiliza ampliamente para el entrenamiento de diversos sistemas de procesamiento de imágenes
- 60.000 imágenes de entrenamiento y 10.000 imágenes de prueba
- Dígitos del 0 al 9



Ejemplo : MNIST





Ayudantía 13

Redes Neuronales

Por Sofía Hosiasson y

22 de Noviembre