

MÓDULO 3

REDES NEURONALES



Índice

1. Introducción y motivación
2. El perceptrón
3. La neurona logística
4. Redes neuronales

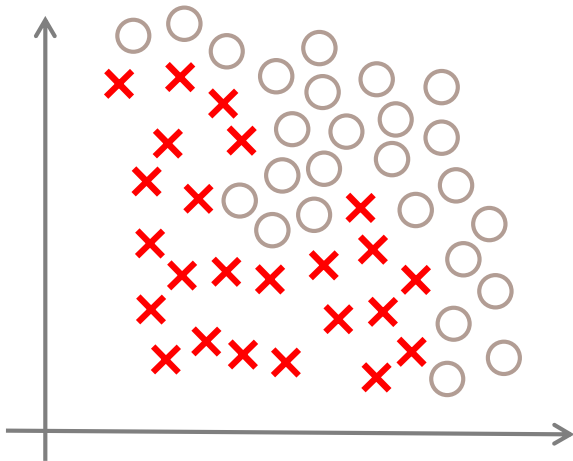


Índice

1. **Introducción y motivación**
2. El perceptrón
3. La neurona logística
4. Redes neuronales



Frontera de decisión no lineal

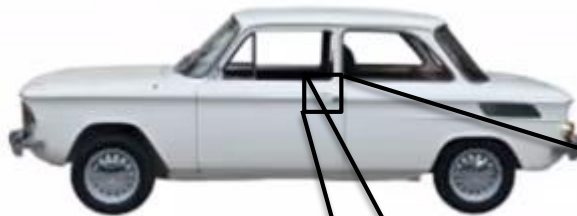


$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

- Añadimos nuevas características
- Si $n = 100$, cuando añadimos características polinomiales de orden 3, entonces $n = 170000$
- Poco eficiente/manejable
- Muy fácil sobre-aprender

Problemas más complejos: clasificación de imágenes

Nosotros vemos...



La cámara ve...

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50



Visión por computado: Detección de coches



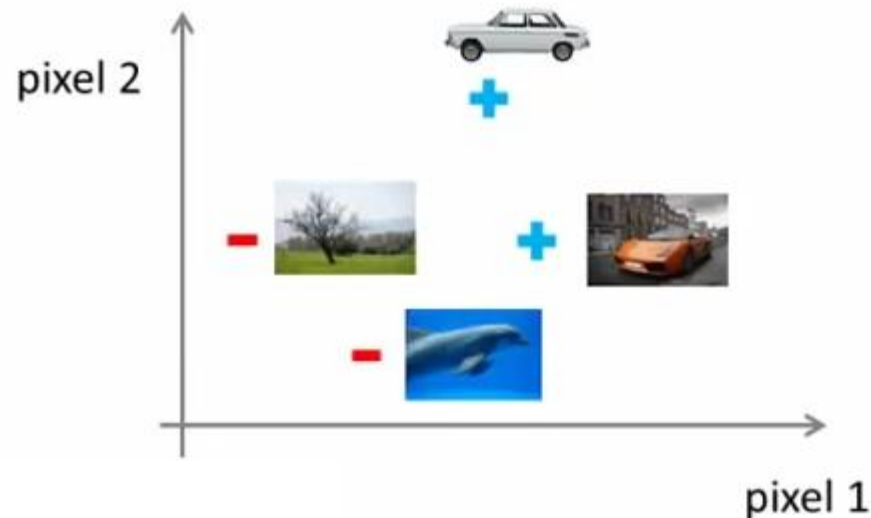
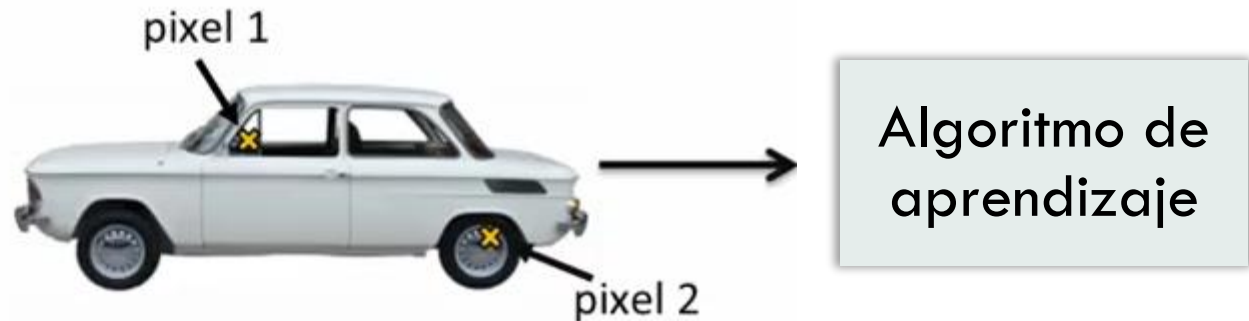
Nuevo ejemplo:



¿Qué es?



Fronteras de decisión complejas



Con imágenes pequeñas 50x50 (RGB), ¿cuántas características tendremos si añadimos características cuadráticas?

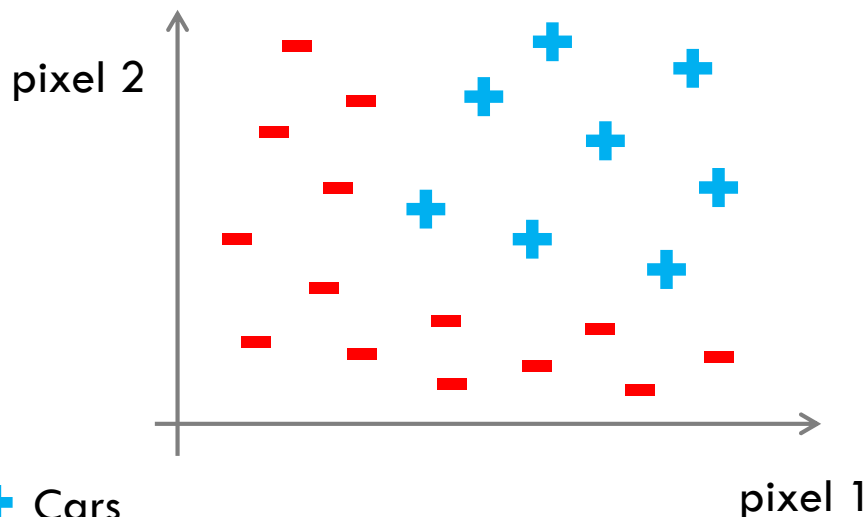


Fronteras de decisión complejas



Algoritmo de aprendizaje

Imágenes de $50 \times 50 \rightarrow 2500$ pixels
(7500 en caso de RGB)



+ Cars
- "Non"-Cars

$$x = \begin{bmatrix} \text{Intensidad pixel 1} \\ \text{Intensidad pixel 2} \\ \vdots \\ \text{Intensidad pixel 2500} \end{bmatrix}$$

Características cuadráticas ($x_i \times x_j$):
 ≈ 3 millones de características



Redes Neuronales

□ Origen

- Algoritmos que tratan de **imitar al cerebro humano**
- Muy utilizadas en los 80 e inicios de los 90
 - Pérdida de popularidad después

□ Actualmente

- **Una de las mejores técnicas** para muchas aplicaciones
 - **Imagen**
 - **Procesamiento del lenguaje natural**



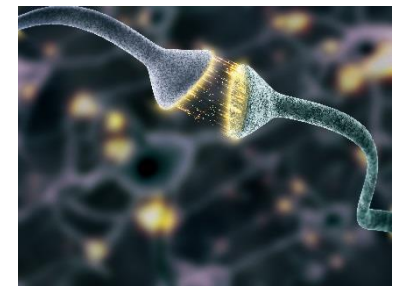
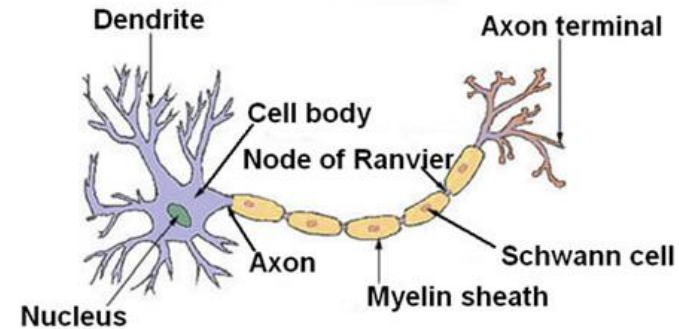
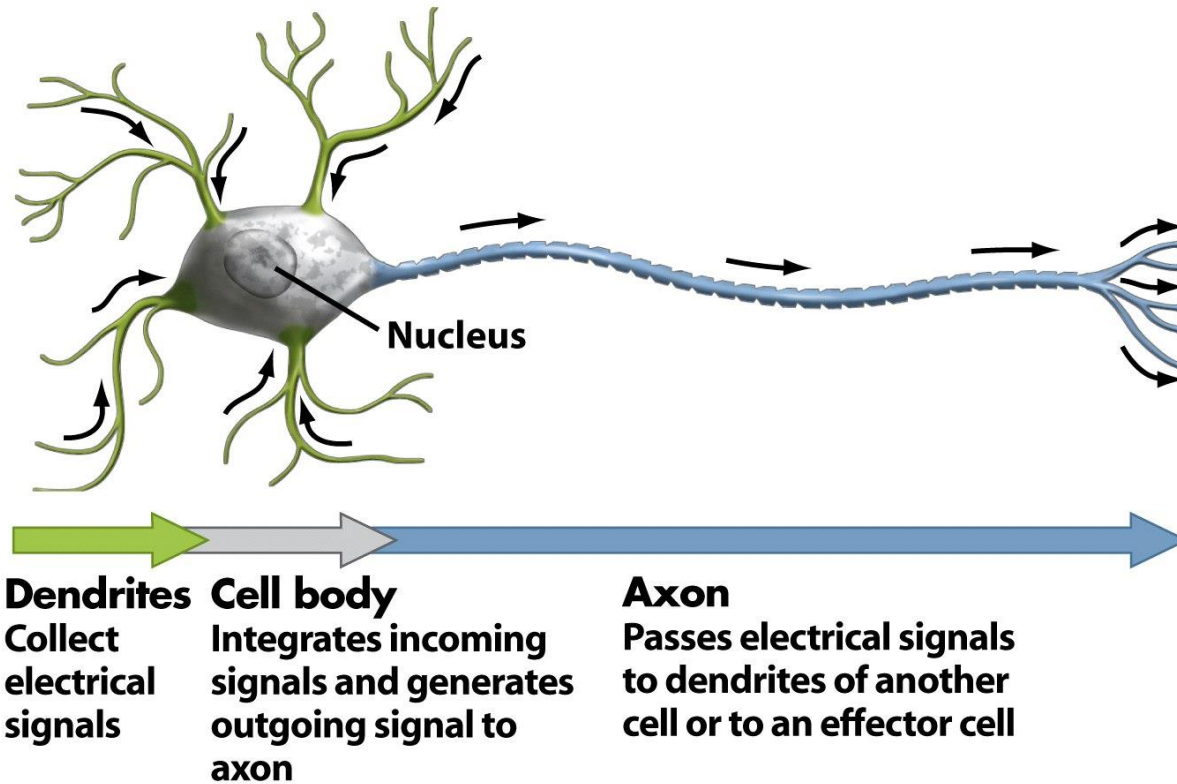
Índice

1. Introducción y motivación
2. **El perceptrón**
3. La neurona logística
4. Redes neuronales



Neuronas

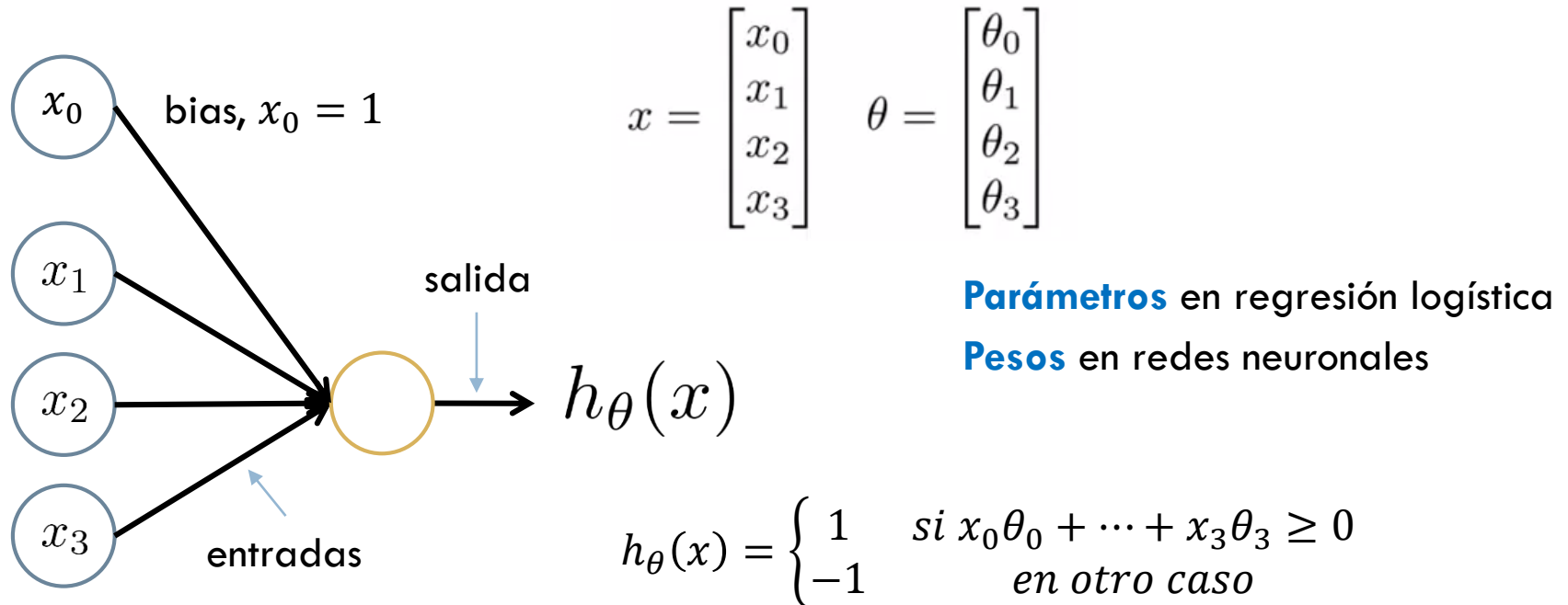
Information flow through neurons



Sinapse



Perceptrón



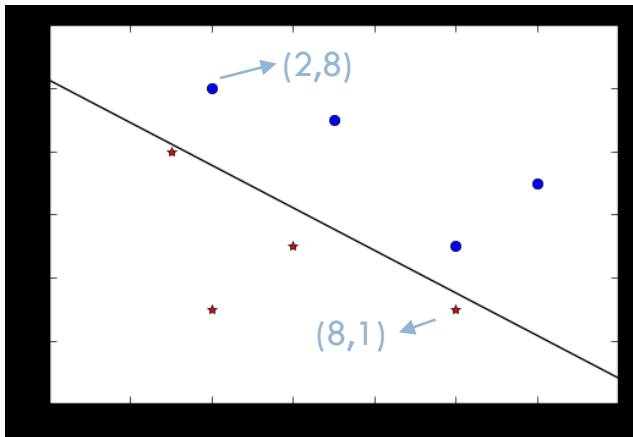
Función de activación

- Es una función escalón (poco suave)
- El hecho de que la salida sea -1 (en vez de 0) es solo por conveniencia



Perceptrón

□ Ejemplo



▶ Pesos

▶ $\theta_1 = 6.92$

▶ $\theta_2 = 10.28$

▶ $\theta_0 = -71.01$

▶ $x^{(1)} = (2,8), \hat{y}^{(1)} = 1$

▶ $x^{(2)} = (8,1), \hat{y}^{(2)} = 0$

- Ejemplo $x^{(1)} = (2,8)$. Salida del perceptrón
 - $6.92 \times 2 + 10.28 \times 8 - 71.01 = 25.07$. $\hat{y} = 1$
- Ejemplo $x^{(2)} = (8,1)$. Salida del perceptrón
 - $6.92 \times 8 + 10.28 \times 1 - 71.01 = -5.37$. $\hat{y} = -1$



Perceptrón

- Los pesos determinan el **hiperplano** que separa a las dos clases. La ecuación de este hiperplano es

$$x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b = 0$$

- En el caso de datos en dos dimensiones, la recta de separación tiene los siguientes parámetros:

- Pendiente: $-\frac{w_1}{w_2}$

- Ordenada en el origen: $-\frac{b}{w_2}$



Perceptrón

- Formulación del objetivo del perceptrón
- Dado
 - ▣ Conjunto de datos de entrenamiento
 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$
 - ▣ Parámetros iniciales del perceptrón $\theta_0, \dots, \theta_n$
- Encontrar
 - ▣ El valor de los parámetros del perceptrón tales que

$$y^{(i)} = \hat{y}^{(i)} \text{ para todo } i = 1, \dots, m$$



Perceptrón

- El perceptrón es uno de los ejemplos más claros de **online learning**

- El modelo recibe una secuencia de ejemplos en orden

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$$

- El modelo recibe $x^{(1)}$ y predice $\hat{y}^{(1)}$
 - Al modelo se le muestra $y^{(1)}$
 - El modelo puede utilizar esta información (o no) para mejorar el aprendizaje
 - Repetimos con el resto de muestras $x^{(2)}, \dots, x^{(m)}$



Perceptrón

- El algoritmo de aprendizaje consta de los siguientes pasos
 - ▣ Paso 1: inicialización **aleatoria** de los pesos (o a cero, es perfectamente válido)
 - ▣ Paso 2: se toma un elemento del vector de entrada y se calcula la salida del perceptrón
 - ▣ Paso 3:
 - si la salida es igual a la clase real, no se modifican los pesos
 - Si la salida es distinta, cada peso se modifica como
 - $\theta_j = \theta_j + (clase\ real - salida)x_j$
 - ▣ Paso 4: parada
 - Si todos los datos de entrada están perfectamente clasificados



Perceptrón

- Recordando...el algoritmo de aprendizaje del perceptrón en **online**
 - ▣ Recibimos los datos **de uno en uno** y por cada uno de ellos actualizamos los pesos
 - Recibimos las coordenadas del dato 1
 - Calculamos su clase según el Perceptrón
 - Recibimos su clase real
 - Actualizamos los pesos del Perceptrón
 - Recibimos las coordenadas del dato 2
 - Calculamos su clase según el Perceptrón
 - ...



Perceptrón

□ Problemas

- ¿Y si nunca es capaz de alcanzar un estado en el que $y^{(i)} = \hat{y}^{(i)}$ para todo i ?
 - Problemas linealmente no separables
- El algoritmo de aprendizaje solo converge
 - Si el problema es linealmente separable
- Un problema de clasificación tipo XOR no puede solucionarse mediante un perceptrón

