

Fundamentos rápidos: del perceptrón a la red profunda

Una neurona, un cliente, una predicción de churn con números.

CONCEPTOS CLAVE

- Regresión predice un valor continuo; clasificación predice una etiqueta discreta. Métrica y KPI deben elegirse **antes** del modelo.
- Una neurona: $y = \sigma(\sum_i w_i x_i + b)$. La no linealidad es lo que da poder al MLP.
- Backprop = regla de la cadena sobre el grafo de operaciones. PyTorch lo hace por ti con `loss.backward()`.
- Optimizador por defecto en 2026: **AdamW**, $lr = 10^{-3}$, $weight_decay = 10^{-4}$.

EJEMPLO A MANO

Forward de una neurona, cliente con $x = (24, 80, 5)$, $w = (-0,04, 0,02, 0,55)$, $b = -0,50$:

$$z = (-0,04)(24) + (0,02)(80) + (0,55)(5) + (-0,50) = \mathbf{2.49},$$

$$\hat{y} = \text{sigmoide}(2,49) = \frac{1}{1+e^{-2,49}} = \mathbf{0.924}.$$

El modelo asigna 92,4 % de churn a ese cliente; con umbral 0,5 entra en campaña de retención.

CONCEPTO → APLICACIÓN TABULAR CLÁSICA

Modelo	Aplicación	KPI medible
Reg. lineal	Pricing alquiler vacacional (Airbnb Madrid)	MAE €/noche
Reg. logística	Scoring crediticio retail (BBVA)	AUC, Gini, € mora evitada
Random Forest	Detección de fraude en transacciones	Recall fraude, FP/día
Gradient Boosting	Forecast demanda multiproducto (Mercadona)	MAPE, días sin stock
MLP profundo	Cuando hay no-linealidades fuertes	Mismo KPI, +5–8 % típico

Idea clave del Día 02. Antes de saltar a redes neuronales, comprueba si una regresión logística entrenada bien ya entrega el 90 % del valor de negocio. **No es si usas IA: es cuánto contexto le das al modelo.**

Para llevarte: (i) lr mal puesto es responsable del 80 % de los entrenamientos fallidos; (ii) el accuracy es engañoso con clases desbalanceadas; (iii) lectura: Prince, *Understanding Deep Learning*, cap. 3.