

Generalización y el Panorama del Deep Learning

Qué problemas resolvemos y qué herramientas existen

Cesar Garcia

2025

Introducción

- Entender qué significa que un modelo *aprenda*
- Diferenciar: generalización, sobreajuste y subajuste
- Reconocer los principales componentes de un proyecto de ML
- Ubicar arquitecturas y optimizadores en el panorama general

¿Qué significa aprender?

Idea central

Un modelo aprende cuando:

- captura **patrones reales**
- funciona bien en **datos no vistos**

Aprender **no es memorizar.**

¿Un modelo que tiene 100 % de accuracy en entrenamiento necesariamente aprendió?

Generalización

Concepto clave

Generalizar significa que el modelo:

- funciona bien en datos nuevos
- no depende de ejemplos específicos

La generalización es el objetivo real del entrenamiento.

¿Por qué nos importa más el rendimiento en validación que en entrenamiento?

Overfitting

Ocurre cuando el modelo:

- aprende detalles irrelevantes (ruido)
- pierde capacidad de generalizar

Señales típicas:

- pérdida muy baja en entrenamiento
- pérdida alta en validación

¿Qué tipo de modelo es más propenso al sobreajuste: uno simple o uno muy complejo?

Underfitting

Ocurre cuando el modelo:

- no tiene suficiente capacidad
- no captura el patrón real

Señales típicas:

- pérdida alta en entrenamiento
- pérdida alta en validación

¿Puede un modelo subajustado mejorar solo entrenando más tiempo?

Técnicas para mejorar la generalización

Visión general

Algunas estrategias comunes:

- Regularización
- Dropout
- Early stopping
- Más y mejores datos

Todas buscan el mismo objetivo:

controlar la complejidad efectiva del modelo.

¿Estas técnicas cambian el modelo o cambian cómo se entrena?

Importancia de los datos

Antes del modelo

- Calidad > cantidad
- Datos mal preparados → modelos pobres
- El modelo solo puede aprender lo que los datos contienen

Procesos comunes:

- normalización
- estandarización
- balanceo (95% Clase A y 5% Clase B)

¿Un modelo muy avanzado puede compensar datos de mala calidad?

Flujo general de un proyecto de ML

Panorama completo

- ① Recolección de datos
- ② Limpieza y procesamiento
- ③ División entrenamiento / validación
- ④ Selección del modelo
- ⑤ Entrenamiento
- ⑥ Evaluación
- ⑦ Ajuste y regularización
- ⑧ Implementación

Este flujo se repite iterativamente.

¿En qué etapa crees que se cometan más errores en la práctica?

Qué tipos de modelos existen

- **Redes densas**: datos tabulares
- **Autoencoders**: compresión y generación
- **CNNs**: imágenes
- **RNNs / LSTM**: secuencias
- **Transformers**: texto, visión, multimodal

¿Por qué no usamos la misma arquitectura para todos los problemas?

Optimizadores modernos

Visión general (sin entrar en detalles)

Durante el entrenamiento ajustamos parámetros usando optimizadores:

- **SGD**: descenso básico
- **Momentum**: acumula dirección
- **RMSProp**: adapta el paso por parámetro
- **Adam**: combina velocidad y estabilidad

Los estudiaremos **en profundidad más adelante**.

¿Por qué tendría sentido usar distintos optimizadores para el mismo modelo?

Idea clave de la sesión

Todo se conecta

- Datos influyen en generalización
- Arquitectura define capacidad
- Optimizadores controlan el aprendizaje
- El objetivo final es **generalizar**

Hoy construimos el mapa; luego veremos el mecanismo.

¿Qué parte del proceso te genera más curiosidad o incertidumbre ahora?