Bucles de Entrenamiento y Evaluación de Modelos

Cómo entrenar correctamente una red neuronal

Cesar Garcia

2025

Introducción

- Entender **qué hace realmente** un bucle de entrenamiento
- Diferenciar claramente **entrenamiento** y **evaluación**
- Comprender épocas, batches e iteraciones
- Introducir validación como herramienta de diagnóstico
- Detectar sobreajuste y subajuste en la práctica

¿Qué parte del entrenamiento te parece más confusa: el código o el proceso?

El entrenamiento como proceso

Idea central

Entrenar una red neuronal **no es una sola operación**.
Es un proceso repetitivo que:

- calcula predicciones
- mide errores
- ajusta parámetros
- evalúa comportamiento

Todo esto ocurre siguiendo **reglas muy estrictas**.

> *¿Qué pasaría si ajustáramos los pesos usando los datos de validación?*

Épocas, batches e iteraciones

Tres escalas distintas

- **Batch**: subconjunto de datos
- **Iteración**: un forward + backward sobre un batch
- **Época**: pasar por todo el dataset una vez

Relación:

1 época = muchos batches = muchas iteraciones

¿Por qué no usamos todo el dataset en una sola iteración?

¿Por qué usar batches?

Motivación práctica

Usar batches permite:

- entrenar con datasets grandes
- introducir ruido útil en el gradiente
- mejorar eficiencia computacional

El gradiente por batch es:

- una **aproximación** del gradiente real
- suficiente para aprender

¿Qué efecto tendría usar un batch demasiado pequeño o demasiado grande?

Batch pequeño	Batch grande
Gradiente ruidoso	Gradiente estable
Más exploración	Menos exploración

Entrenamiento vs inferencia

Dos modos distintos

Durante **entrenamiento**:

- se calculan gradientes
- se actualizan parámetros
- algunas capas se comportan distinto

Durante **inferencia**:

- no se ajustan pesos
- solo se hacen predicciones

En PyTorch esto se refleja en:

- `model.train()`
- `model.eval()`

¿Por qué una red debería comportarse distinto al entrenar que al predecir?

El bucle de entrenamiento

Estructura mínima

Un bucle típico incluye:

1. Obtener un batch de datos
2. Forward pass
3. Calcular la pérdida
4. Backward pass
5. Actualizar parámetros
6. Reiniciar gradientes

Nada más. Nada menos.

¿Qué paso rompería el entrenamiento si lo olvidamos?

El bucle de validación

Propósito distinto

La validación sirve para:

- medir generalización
- detectar sobreajuste
- comparar modelos

Características clave:

- **no** se actualizan pesos
- **no** se calculan gradientes
- se usan datos no vistos

¿Por qué la validación no debe influir en el entrenamiento directamente?

Pérdida vs métricas

No son lo mismo

- **Pérdida**:
  - señal de entrenamiento
  - optimizable
  - diferenciable
- **Métricas (accuracy, etc.)**:
  - interpretación humana
  - no optimizables directamente

Entrenamos con pérdida. Evaluamos con métricas.

¿Por qué no entrenamos directamente usando accuracy?

Seguimiento del entrenamiento

Mirar curvas, no números

Observar:

- pérdida de entrenamiento
- pérdida de validación
- su relación en el tiempo

Patrones importan más que valores absolutos.

¿Qué información aporta la forma de la curva que no da un solo número?

Sobreajuste en la práctica

Señales claras

Sobreajuste ocurre cuando:

- pérdida de entrenamiento $\downarrow$
- pérdida de validación $\uparrow$

El modelo memoriza, no generaliza.

¿Entrenar más tiempo siempre empeora el sobreajuste?

Subajuste en la práctica

Capacidad insuficiente

Subajuste ocurre cuando:

- pérdida de entrenamiento alta
- pérdida de validación alta

El modelo no aprende el patrón.

¿Qué es más efectivo aquí: más datos o un modelo más complejo?

Errores comunes

Qué evitar

- Mezclar entrenamiento y validación
- Mirar solo accuracy
- No reiniciar gradientes
- Cambiar demasiadas cosas a la vez
- Sacar conclusiones con pocas épocas

¿Cuál de estos errores crees que es más fácil de cometer?

Idea clave de la sesión

Disciplina antes que complejidad

Un buen entrenamiento:

- es repetible
- es limpio
- es diagnosticable

Antes de arquitecturas modernas:

aprende a entrenar bien modelos simples.

¿Qué parte del bucle de entrenamiento te parece más frágil ahora?