Técnicas de Regularización

Cómo reducir overfitting y mejorar generalización

Cesar Garcia

2025

Introducción

- Identificar **overfitting** en curvas de entrenamiento/validación
- Entender regularización **L2** (weight decay) y **L1** (sparsity)
- Explicar **dropout** como regularización estocástica
- Usar **early stopping** como control de capacidad efectiva
- Conectar regularización con **generalización** (no solo métricas)

¿Cómo sabes que tu modelo "memoriza" en vez de aprender patrones?

El problema: overfitting

Señal típica en curvas

- **Train loss** baja de forma consistente
- **Val loss** deja de bajar y luego sube
- La brecha train–val crece

Consecuencia:

- buen desempeño en datos vistos
- mal desempeño en datos nuevos

Si tu accuracy en train sube pero en validación baja, ¿qué está pasando?

Intuición: capacidad vs datos

Por qué ocurre

Overfitting aparece cuando:

- el modelo tiene **mucha capacidad**
- hay **pocos datos** (o poca diversidad)
- hay **ruido** o etiquetas imperfectas

Una red grande puede aprender:

- reglas generales
- y también "atajos" del conjunto de entrenamiento

¿Qué cambiarías primero: el modelo, los datos o el criterio de parada?

Regularización

Qué significa "regularizar"

Regularizar es **forzar preferencias** sobre soluciones:

- más simples
- más estables
- menos sensibles al ruido

Se puede lograr:

- modificando la **función de pérdida**
- modificando la **arquitectura**
- modificando el **proceso de entrenamiento**

¿Qué tipo de "preferencia" crees que es útil imponer?

L2 (Weight Decay)

Penalizar pesos grandes

Idea:

- agregar un costo por pesos grandes
- empuja a soluciones con menor norma

Forma típica:

$$L_{total} = L_{data} + \lambda \|W\|_2^2$$

Efecto práctico:

- decisiones más suaves
- menor sensibilidad a outliers

¿Qué comportamiento esperas si λ es demasiado grande?

L1

Promover sparsity

L1 penaliza la suma absoluta:

$$L_{total} = L_{data} + \lambda \|W\|_1$$

Propiedad clave:

- empuja muchos pesos a **cero**
- induce modelos más "escasos" (sparse)

Útil cuando:

- quieres selección implícita de features
- hay muchas variables irrelevantes

¿Por qué L1 tiende a "apagar" pesos en lugar de solo reducirlos?

Dropout

Regularización estocástica

Durante entrenamiento:

- se “apagan” neuronas aleatoriamente con probabilidad $p)
- la red aprende a no depender de una sola ruta

Intuición:

- entrenas un “ensamble” de subredes
- reduces co-adaptación

Importante:

- en evaluación, dropout se desactiva

¿Qué tipo de dependencia del modelo rompe dropout?

Dropout: detalle operativo

Qué cambia en forward

Durante entrenamiento:

- máscara aleatoria $\rightarrow$ activaciones parciales

Durante evaluación:

- activaciones completas (sin aleatoriedad)

Por eso:

- `model.train()` $\neq$ `model.eval()`

¿Qué error común ocurre si olvidas cambiar a `eval()` al evaluar?

Early stopping

Parar antes de memorizar

Early stopping:

- monitorea validación
- detiene entrenamiento cuando no mejora

Efecto:

- limita “capacidad efectiva”
- evita que el modelo empiece a ajustar ruido

Regla práctica:

- paciencia (patience) + mejor checkpoint

¿Por qué early stopping es una forma de regularización aunque no cambie la arquitectura?

Data augmentation

Más datos (sin recolectar más)

Data augmentation crea variaciones:

- rotaciones, recortes, flips (imágenes)
- jitter, ruido, mezclas (numérico/series)

Meta:

- aumentar diversidad
- enseñar invariancias

En 2D (toy):

- agregar ruido controlado o pequeñas transformaciones

¿Qué invariancia "enseñas" si agregas ruido gaussiano al input?

Comparación mental rápida

Cuándo usar qué

- L2: baseline robusto, casi siempre
- L1: sparsity / selección implícita
- Dropout: redes densas y modelos con co-adaptación
- Early stopping: cuando val mejora y luego se degrada
- Augmentation: cuando faltan datos o diversidad

¿Cuál de estas técnicas aplicarías primero y por qué?

Puente al notebook

Qué vamos a medir

En el notebook:

1. provocaremos overfitting con **pocos datos** + **modelo grande**
2. aplicaremos:
   - L2 (weight decay)
   - L1 (penalización manual)
   - Dropout
   - Early stopping
   - Augmentación simple (ruido)
3. compararemos curvas y generalización

Si solo pudieras elegir una técnica, ¿cuál elegirías para este caso?

Idea clave de la sesión

Generalizar es el objetivo

Regularización no es “truco”:

- es un mecanismo para controlar capacidad
- y alinear el aprendizaje con patrones estables

Meta final:

mejor desempeño en datos no vistos

¿Qué evidencia te convence de que un modelo generaliza?