Clase 116 — Regularización: L1/L2, dropout, max-norm, MC dropout (+ Stochastic Depth, DropPath)

Parte: 2 — Deep Learning · Fuente: Géron, cap. 11 § Regularization + Huang et al. (2016) Deep Networks with Stochastic Depth. Duración estimada: 80 min.

Clase 116 — Regularización: L1/L2, dropout, max-norm, MC dropout (+ Stochastic Depth, DropPath)

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 11 § Regularization + Huang et al. (2016) Deep Networks with Stochastic Depth. Duración estimada: 80 min.*

Objetivo

Conocer las técnicas de regularización en DL —L1/L2, dropout (Srivastava et al. 2014), max-norm, MC dropout para incertidumbre— y las técnicas modernas que se usan en arquitecturas profundas (ResNets, ViT, Transformers): Stochastic Depth, DropPath y LayerDrop.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Aplicar keras.regularizers.l1(...), l2(...), l1_l2(...) en una capa.
- Aplicar Dropout(rate=0.5) y entender qué hace en train vs en inference (default desactivado).
- Implementar Monte Carlo dropout (Dropout(0.5) activo en inference → predicciones diferentes → incertidumbre).
- Aplicar Stochastic Depth en una ResNet: dropear bloques residuales completos al azar durante training.
- Aplicar DropPath (estándar en ViT, Swin Transformer, ConvNeXt).

Temas

- L1/L2 como penalización en la loss. λ típicamente 1e-4 a 1e-2.
- Dropout: enmascarar fracción r de las activaciones por batch.
- Inverted dropout: en inference no se hace nada porque train ya escala por 1/(1-r).
- Max-norm constraint: ||w|| ≤ c por neurona después de cada update.
- MC Dropout (Gal & Ghahramani 2016): incertidumbre bayesiana aproximada.
- Complemento moderno: Stochastic Depth, DropPath (= Stochastic Depth aplicado a paths de attention/FFN), LayerDrop (Fan et al. 2020).

Versión profundizada — 2026

El tema moderno que vivía como complemento dentro de esta clase ahora tiene clase propia dedicada con patrón completo, ejercicios y homework:

- Clase 104b — Regularización moderna: Stochastic Depth, DropPath, LayerDrop

Definiciones y características

- L1 regularization: agrega λ·Σ|w| a la loss. Promueve sparsity.
- L2 regularization (weight decay): agrega λ·Σw². Mantiene pesos chicos.
- Dropout: enmascara fracción r de neuronas por batch. Forzar redundancia.
- MC Dropout: hacer N predicciones con dropout activo → distribución de predicciones → incertidumbre.

- Max-norm: constraint sobre la norma de los pesos por unidad.
- Stochastic Depth: dropear bloques residuales enteros durante training.
- DropPath: como Stochastic Depth pero para paths en transformer (attention o FFN).

Dataset / recursos

- Fashion-MNIST + un MLP propenso a overfit.
- Librerías: tensorflow, keras, matplotlib.

Ejercicios

1. Sin regularización: entrenar un MLP grande ([512, 256, 128]) en Fashion-MNIST y observar overfitting (gap train/val ≥ 5 pp).
2. L2: agregar kernel_regularizer=keras.regularizers.l2(1e-3) a cada Dense. Comparar.
3. Dropout: agregar Dropout(0.3) entre Dense layers. Comparar.
4. MC Dropout: para 1 sample de test, hacer 100 predicciones con model(x, training=True). Calcular mean ± std de las probabilidades. Interpretar la incertidumbre.
5. Stochastic Depth simulado: en un mini ResNet con 8 bloques, dropear cada bloque con prob 0.1 lineal. Comparar contra sin stochastic depth.

Homework verificable

Sobre Fashion-MNIST con MLP [512, 256, 128, 64]:

1. Entrenar 4 versiones: sin regularización; L2(1e-3); Dropout(0.3); L2 + Dropout combinados.
2. Reportar train_acc y val_acc; calcular el gap.
3. Para el mejor modelo, hacer MC Dropout con 50 muestras sobre 5 imágenes ambiguas y reportar incertidumbre.

Criterio de aceptación: el modelo regularizado tiene gap train-val menor a 3 pp (vs ~6 pp del baseline) y val_acc igual o mejor. MC dropout debe asignar mayor std a las imágenes ambiguas.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Dropout en inferencia da resultados distin	Pasaste training=True por error. Fix: en i
L2 con λ=1.0 y modelo no aprende	Penalización demasiado fuerte. Fix: λ típi
Dropout(0.5) en la última capa antes de so	Distorsiona logits. Fix: dropout en capas
L2 + AdamW con weight_decay → doble penali	Usar uno: AdamW(wd=...) o kernel_regulariz
Stochastic Depth con p_i constante en luga	Funciona pero menos óptimo. Fix: p_i = i/N

Preguntas frecuentes

¿Dropout 0.5 siempre?

0.5 para capas Dense grandes. Para capas Conv: 0.1-0.2. Para embeddings y attention en Transformers: 0.1.

¿BN ya regulariza, necesito dropout también?

Depende. En CNNs/MLPs con BN, dropout a veces ya no aporta. En Transformers, sí (BN no se usa allí; LN + dropout + DropPath).

¿MC Dropout es bayesiano "de verdad"?

Aproxima un proceso gaussiano variacional. No es bayesiano riguroso pero es una excelente aproximación práctica para incertidumbre.

¿Stochastic Depth en CNN no residual?

No tiene sentido — Stochastic Depth necesita la skip connection para que dropear no rompa el forward.

¿Cuánta dropout/droppath en ViT base?

ViT-Base original: dropout=0.1 en attention, droppath=0.1 lineal en cada bloque. Para fine-tuning, suele bajarse a 0.0.

Referencias

- Géron, cap. 11 — Regularization Using Dropout.
- Srivastava et al. (2014), Dropout, JMLR.
- Gal & Ghahramani (2016), Dropout as a Bayesian Approximation, ICML — MC dropout.
- Huang et al. (2016), Deep Networks with Stochastic Depth, ECCV.
- Fan et al. (2020), Reducing Transformer Depth on Demand with Structured Dropout (LayerDrop).
- keras DropPath / StochasticDepth.

Siguiente clase

Clase 117 — Regularización moderna: Stochastic Depth, DropPath, LayerDrop

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb