Clase 108 — Inicialización (Glorot, He)

Parte: 2 — Deep Learning · Fuente: Géron, cap. 11 § Glorot and He Initialization. Duración estimada: 55 min.

Clase 108 — Inicialización (Glorot, He)

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 11 § Glorot and He Initialization. Duración estimada: 55 min.*

Objetivo

Saber inicializar los pesos de cada capa para que la varianza de las activaciones y de los gradientes se mantenga estable a lo largo del forward y backward pass. Diferenciar Glorot (Xavier) —para sigmoid/tanh— de He (Kaiming) —para ReLU y variantes—. Saber cuál usa Keras por default y cuándo cambiarlo.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar la idea: Var(W) ≈ 1 / fan_in (o promedio fan_in/fan_out) para preservar varianza.
- Aplicar kernel_initializer='glorot_uniform' (default Keras), 'he_normal', 'he_uniform'.
- Calcular a mano los límites de la distribución para Glorot uniform: ±√(6/(fan_in+fan_out)).
- Reconocer que la combinación correcta es He init + ReLU, Glorot + tanh/sigmoid.
- Inspeccionar el efecto visualmente: histogramas de activaciones por capa.

Temas

- ¿Por qué importa la varianza? Productos de N capas amplifican o atenúan exponencialmente.
- Glorot (2010): Var(W) = 2/(fan_in + fan_out). Asume activación lineal/simétrica.
- He (2015): Var(W) = 2/fan_in. Compensa que ReLU "mata" la mitad de las salidas.
- Distribuciones: uniform o normal. Equivalentes prácticamente.
- LeCun init: Var(W) = 1/fan_in. Para SELU.

Definiciones y características

- fan_in: número de entradas a la capa (= units de la capa anterior).
- fan_out: número de salidas (= units de la capa).
- Glorot uniform: U(-√(6/(fan_in+fan_out)), +√(6/(fan_in+fan_out))). Default de Keras Dense.
- He normal: N(0, √(2/fan_in)). Recomendado para ReLU, Leaky ReLU, ELU.
- LeCun normal: N(0, √(1/fan_in)). Para SELU (self-normalizing networks).
- Inicialización por capa: Dense(64, kernel_initializer='he_normal', bias_initializer='zeros').

Dataset / recursos

- Fashion-MNIST.
- Librerías: tensorflow, keras, matplotlib.

Ejercicios

1. Inspección de defaults: para Dense(128, input_shape=(784,)), imprimir

model.layers[0].kernel.numpy(). Calcular la varianza empírica y compararla con la teórica de Glorot.

2. Comparación: entrenar MLP [512, 256, 128, 64, 10] con ReLU. Probar 3 inits: Glorot, He, RandomNormal(stddev=0.01). Graficar val_loss en las 3.
3. Histogramas de activaciones: para cada capa del modelo bien inicializado, plot del histograma de salidas para un batch. Verificar que la varianza se mantiene similar entre capas.
4. He init + Tanh: probar la combinación incorrecta (He con tanh). Comparar contra Glorot + tanh. Verificar que importa.
5. Reset y reproducibilidad: con tf.random.set_seed(42) + np.random.seed(42), entrenar 2 veces y verificar que da idéntico. Sin seed, varía.

Homework verificable

Sobre MLP [300, 200, 100, 50, 10] con ReLU sobre Fashion-MNIST:

1. Entrenar con 3 inits: default Glorot, He uniform, He normal.
2. Reportar val_accuracy tras 20 épocas para cada uno.
3. Para el mejor (He init), inspeccionar la norma de cada kernel antes y después del entrenamiento.

Criterio de aceptación: He init debe igualar o superar Glorot por ≥ 0.5 pp; la norma de los kernels post-entrenamiento debe ser similar entre capas (no orders of magnitude apart).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Modelo arranca lento con ReLU	Estás usando Glorot por default. Fix: kern
RandomNormal(stddev=0.01) por costumbre de	Vanishing inmediato. Fix: usar Glorot/He s
Reset de pesos no funciona con model.set_w	Hay que guardar initial = [w.numpy() for w
Bias init glorot_uniform por error	Bias debe arrancar en zeros (default). Ini
Inicializar igual una capa convolucional q	fan_in para Conv es kernel_h × kernel_w ×

Preguntas frecuentes

¿Glorot uniform o normal?

Para Glorot, casi indistinguible en práctica. Keras default es glorot_uniform. Para He, he_normal es ligeramente preferido pero las diferencias son ínfimas.

¿LeCun init cuándo?

Solo con SELU (activación que se auto-normaliza). Esto fue un experimento de 2017 (Klambauer et al.) que perdió tracción frente a BatchNorm + ReLU/GELU.

¿BatchNorm hace innecesaria la inicialización?

Casi. BN normaliza el output dentro del forward → init importa menos. Pero un mal init aún hace que las primeras épocas sean inestables.

¿Init para Transformers?

Combinación cuidadosa: linears con truncated_normal(stddev=0.02) (GPT/BERT-style), embeddings con normales escaladas. Lo verás en clase 126.

¿Por qué fan_in + fan_out en Glorot?

Es el promedio armónico de "preservar varianza en forward (1/fan_in)" y "preservar varianza en backward (1/fan_out)". Compromiso óptimo bajo activación lineal.

Referencias

- Géron, cap. 11 — Glorot and He Initialization.
- Glorot & Bengio (2010), Understanding the difficulty of training deep feedforward neural networks, AISTATS.
- He, Zhang, Ren & Sun (2015), Delving Deep into Rectifiers, ICCV — paper original de He init.
- Keras initializers.

Siguiente clase

Clase 109 — Activaciones: ReLU, ELU, GELU, Swish, Mish

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb