Clase 104 — Callbacks, TensorBoard, guardar/restaurar modelos

Parte: 2 — Deep Learning · Fuente: Géron, cap. 10 § Using Callbacks y § Using TensorBoard for Visualization. Duración estimada: 60 min.

Clase 104 — Callbacks, TensorBoard, guardar/restaurar modelos

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 10 § Using Callbacks y § Using TensorBoard for Visualization. Duración estimada: 60 min.*

Objetivo

Inyectar lógica al loop de entrenamiento sin modificarlo, mediante callbacks (EarlyStopping, ModelCheckpoint, ReduceLROnPlateau, custom). Visualizar el progreso del training en TensorBoard (loss, métricas, histogramas de pesos, embeddings). Saber guardar y restaurar correctamente — arquitectura + pesos + estado del optimizador.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Aplicar los 4 callbacks más usados: EarlyStopping, ModelCheckpoint, ReduceLROnPlateau, TensorBoard.
- Configurar TensorBoard: keras.callbacks.TensorBoard(log_dir='./logs') y abrirlo con tensorboard --logdir=./logs.
- Escribir un callback custom heredando de keras.callbacks.Callback con hooks como on_epoch_end.
- Distinguir guardado completo (model.save('m.keras')) vs solo pesos (model.save_weights('w.weights.h5')).
- Restaurar y continuar entrenamiento desde checkpoint sin pérdida.

Temas

- Callbacks: hooks (on_train_begin, on_epoch_end, on_batch_end, ...).
- EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True).
- ModelCheckpoint(filepath, save_best_only=True, monitor='val_accuracy', mode='max').
- ReduceLROnPlateau(factor=0.5, patience=5) — bajar LR cuando se estanca.
- TensorBoard: scalars, histograms, distributions, images, projector (embeddings).
- Custom callbacks: class MyCallback(keras.callbacks.Callback): def on_epoch_end(self, epoch, logs): ....

Definiciones y características

- Callback: objeto con métodos hook que Keras invoca en momentos específicos del training. Permite logging, early stopping, LR scheduling, etc.
- EarlyStopping: monitorea una métrica y corta cuando deja de mejorar. restore_best_weights=True revierte a los mejores pesos vistos.
- ModelCheckpoint: guarda el modelo (o solo pesos) cuando una métrica mejora.
- ReduceLROnPlateau: reduce el LR multiplicándolo por factor cuando la métrica se estanca por patience épocas.
- TensorBoard: la herramienta de visualización oficial de TF. Sirve via tensorboard --logdir=....
- logs dict: contiene loss, val_loss, métricas, etc. Disponible en hooks on_epoch_end.

Dataset / recursos

- Fashion-MNIST o cualquier modelo entrenable de clases anteriores.
- Librerías: tensorflow, keras, tensorboard (incluido).

Ejercicios

1. EarlyStopping + Checkpoint: entrenar Fashion-MNIST con ambos callbacks. Verificar que cortó cuando val_loss se estancó y que 'best.keras' contiene los mejores pesos.
2. TensorBoard: agregar TensorBoard(log_dir=f'./logs/run-{time}'), entrenar 10 épocas. Lanzar tensorboard --logdir=./logs y revisar scalars + histograms.
3. ReduceLROnPlateau: configurar factor=0.5, patience=3, min_lr=1e-6. Graficar el LR a lo largo de las épocas (usar el logs del callback).
4. Custom callback: escribir uno que loggee a un CSV el (epoch, loss, val_loss, lr_actual) para análisis offline.
5. Restaurar y continuar: entrenar 10 épocas, guardar, recargar y continuar 5 épocas más. Verificar que el optimizer state (momentum de Adam) se preservó.

Homework verificable

Entrenamiento "production-ready" sobre Fashion-MNIST:

1. MLP [300, 100].
2. Callbacks: EarlyStopping(patience=10), ModelCheckpoint('best.keras', save_best_only=True), ReduceLROnPlateau(patience=3), TensorBoard('./logs/').
3. Entrenar epochs=100 (sabiendo que EarlyStopping cortará antes).
4. Recargar best.keras y evaluar en test.
5. Capturar un screenshot del TensorBoard mostrando loss y val_loss a lo largo del training.

Criterio de aceptación: el modelo final restaurado debe ser el del epoch con menor val_loss; accuracy en test ≥ 0.87.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
EarlyStopping corta en la primera época co	Default es patience=0. Fix: patience=5-10
ModelCheckpoint guarda en cada época con s	Llena el disco. Fix: save_best_only=True.
model.save_weights('w.h5') y al cargar dic	Hay que reconstruir la arquitectura antes
TensorBoard no muestra nada en localhost:6	El log_dir está mal o aún no se escribió n
ReduceLROnPlateau y LearningRateScheduler	Conflicto, ambos cambian el LR. Fix: elegí

Preguntas frecuentes

¿save_best_only=True con qué métrica?

monitor='val_loss' por default (modo min). Si monitoreás accuracy, agregá mode='max'.

¿Puedo usar varios ModelCheckpoint?

Sí. Útil para guardar el mejor por val_loss y el mejor por val_accuracy en dos archivos.

¿TensorBoard funciona en Colab?

Sí: %load_ext tensorboard + %tensorboard --logdir=./logs. Funciona inline.

¿Custom callback necesita herencia explícita?

Sí: class MyCB(keras.callbacks.Callback). La base provee self.model y manejo de logs.

¿Format .keras vs .h5?

.keras (Keras 3+) es zip transparente, futuro-proof. .h5 aún funciona pero deprecado para nuevos proyectos.

Referencias

- Géron, cap. 10 — Using Callbacks y Using TensorBoard.
- Keras callbacks.
- TensorBoard quickstart.

Siguiente clase

Clase 105 — Keras Tuner (+ Optuna, Ray Tune)

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb