

---

## **Clase 119 — Losses, métricas, capas, modelos custom**

Parte: 2 — Deep Learning · Fuente: Géron, cap. 12 § Customizing Models and Training Algorithms. Duración estimada: 70 min.

## Clase 119 — Losses, métricas, capas, modelos custom

Parte: 2 — Deep Learning · Fuente: Géron, cap. 12 § Customizing Models and Training Algorithms.  
Duración estimada: 70 min.

### Objetivo

Crear losses, métricas y capas custom cuando los builtins de Keras no alcanzan: focal loss, métrica F1 macro, una capa con normalización custom, modelo subclassed con `train_step` propio. Diferenciar `stateless` (función) de `stateful` (clase con estado acumulado por época).

### Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Definir loss custom como función: `def my_loss(y_true, y_pred): return ...`
- Definir métrica `stateful` heredando `keras.metrics.Metric` con `update_state`, `result`, `reset_state`.
- Crear capa custom heredando `keras.layers.Layer` con `build` (declara pesos) y `call` (forward).
- Overridar `train_step` de un modelo subclassed (Model) para custom training logic.
- Saber cuándo usar custom vs cuándo los builtins de Keras alcanzan (casi siempre).

### Temas

- Loss como función simple: 2 args (`y_true`, `y_pred`), devuelve un tensor.
- Métrica `stateless` (función) vs `stateful` (Metric class).
- Capa custom: `__init__` (config), `build` (pesos), `call` (forward).
- Model subclass con `train_step(self, data)` custom.

### Definiciones y características

- Loss: función que devuelve un escalar (o vector por sample) que minimizamos.
- Métrica `stateful`: acumula a lo largo del epoch (necesario para F1, precision, recall, AUC).
- `build(input_shape)`: hook donde Keras te dice las dims; ahí declararás pesos con `self.add_weight`.
- `get_config()`: serialización — necesario si querés `model.save()`.
- `train_step`: lo que Keras llama por cada batch durante fit. Default es: forward, compute loss, backward, optimizer step.

### Dataset / recursos

- Fashion-MNIST o cualquier dataset previo.
- Librerías: tensorflow, keras.

### Ejercicios

1. Loss custom: implementar Focal Loss  $FL(p_t) = -\alpha(1-p_t)^\gamma \log(p_t)$  (Lin et al. 2017, útil para imbalance). Aplicar a Fashion-MNIST y comparar contra cross-entropy.

- Métrica F1 macro: heredar `keras.metrics.Metric`, mantener confusion matrix acumulada por época, calcular F1 macro en `result()`.
- Capa custom: `class L2Normalize(Layer)` que normaliza cada vector a norma 1. Probarla en un modelo.
- Modelo con `train_step` custom: subclass que en cada batch aplica gradient clipping manual + logging extra.
- `get_config`: agregar a una capa custom; verificar que `model.save()` y `load_model(custom_objects=...)` funciona.

## Homework verificable

Sobre un dataset desbalanceado (simular o usar uno con clase minoritaria al 5 %):

- Entrenar con cross-entropy estándar; reportar F1 macro.
- Entrenar con Focal Loss custom ( $\gamma=2.0$ ,  $\alpha=0.25$ ); reportar F1.
- Implementar una métrica F1 macro custom y usarla durante el training.

Criterio de aceptación: Focal Loss debe mejorar F1 macro en  $\geq 2$  pp respecto a cross-entropy en el caso desbalanceado.

## Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Loss custom devuelve shape incorrecto	Debe ser (batch,) o escalar. Fix: revisar
Métrica custom devuelve siempre el mismo v	Olvidaste <code>reset_state</code> entre épocas. Keras
<code>model.save()</code> falla con capa custom	Falta <code>get_config</code> . Fix: implementarlo y <code>cls</code>
Pesos creados en <code>__init__</code> en vez de <code>build</code>	Funciona pero pierde la flexibilidad de sh
Override <code>train_step</code> y se pierde el logging	Hay que actualizar <code>self.compiled_metrics.u</code>

## Preguntas frecuentes

¿Custom loss o `tf.keras.losses.Loss` subclass?

Función para casos simples; subclass cuando tenés estado o configs complejas.

¿Métrica función vs subclass?

Función para batch-wise (accuracy). Subclass para cualquier cosa que necesite acumular (precision, recall, F1, AUC).

¿call o `__call__` en capa custom?

Definí `call`; Keras envuelve para llamar a través de `__call__` agregando training/masking.

¿Custom training loop o subclass con `train_step`?

Subclass `train_step` cuando podés (preserva fit, callbacks, etc.). Custom loop completo solo si necesitás control de flujo realmente complejo (clase 108).

¿Por qué `add_weight` y no `tf.Variable`?

`add_weight` registra automáticamente la variable como peso entrenable del modelo, gestiona dtype, initializer,

regularizer y nombre.

## Referencias

- Géron, cap. 12 — Custom Loss Functions, Metrics, Layers and Models.
- Lin et al. (2017), Focal Loss for Dense Object Detection, ICCV.
- Keras — Making new layers and models.
- Keras — Customize what happens in fit().

## Siguiente clase

Clase 120 — Funciones y grafos (autograph)

## Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

## Archivos complementarios

- notebook.ipynb