Clase 118 — TensorFlow: tensores, variables, operaciones

Parte: 2 — Deep Learning · Fuente: Géron, cap. 12 § Using TensorFlow like NumPy.
Duración estimada: 60 min.

Clase 118 — TensorFlow: tensores, variables, operaciones

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 12 § Using TensorFlow like NumPy. Duración estimada: 60 min.*

Objetivo

Bajar un nivel por debajo de Keras: trabajar directamente con tensores (tf.Tensor) y variables (tf.Variable), entender la API NumPy-like de TF (tf.matmul, tf.reduce_*, tf.cast, tf.reshape), y diferenciar inmutable (Tensor) de mutable (Variable, base de los pesos).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Crear tensores con tf.constant, tf.zeros, tf.ones, tf.random.normal.
- Aplicar operaciones: aritméticas, matmul, broadcasting, indexing/slicing, reduce_mean/sum/max.
- Convertir entre TF y NumPy (.numpy(), tf.convert_to_tensor).
- Crear y modificar tf.Variable con .assign, .assign_add.
- Reconocer dtypes (float32, float64, int32, bool) y forzar cast cuando hace falta.

Temas

- tf.Tensor: inmutable, similar a np.ndarray.
- tf.Variable: mutable, base de los pesos.
- Broadcasting (idéntico a NumPy).
- Operaciones reduce con axis=.
- tf.function (anticipo clase 107) — convierte una función Python en grafo.
- TF en GPU: ops sobre tensores van a GPU automáticamente si está disponible.

Definiciones y características

- tf.constant: crea un Tensor inmutable.
- tf.Variable: tensor mutable, registrable como peso de un modelo.
- Broadcasting: (3, 1) + (1, 4) → (3, 4). Reglas idénticas a NumPy.
- dtype: tipo numérico. Mismatchs disparan errores; usar tf.cast(x, tf.float32).
- .assign(): actualizar el contenido de una Variable in-place. v.assign_add(g) es v += g.
- .shape y .numpy(): forma del tensor y conversión a array NumPy.

Dataset / recursos

- Operaciones a mano (no requiere dataset).
- Librerías: tensorflow, numpy.

Ejercicios

1. Tensores básicos: crear t = tf.constant([[1.0, 2.0], [3.0, 4.0]]). Imprimir shape, dtype, t.numpy().
2. Operaciones: tf.matmul(t, t), tf.transpose(t), tf.reduce_sum(t, axis=0). Verificar shapes.
3. Broadcasting: a = tf.constant([[1.], [2.], [3.]]) (shape 3,1), b = tf.constant([10., 20., 30.]) (3,). a + b → ¿qué shape?
4. Variable y assign: v = tf.Variable([1., 2., 3.]); v.assign([4., 5., 6.]); v.assign_add([1., 1., 1.]). Verificar.
5. dtype mismatch: tf.constant([1, 2, 3]) + tf.constant([1.0, 2.0, 3.0]) → error. Arreglar con tf.cast.

Homework verificable

Implementar regresión lineal manualmente con TF (sin Keras):

1. Generar X = tf.random.normal((100, 2)), y_true = X @ tf.constant([[1.], [2.]]) + 3 + ruido.
2. Variables w = tf.Variable(tf.random.normal((2, 1))), b = tf.Variable(0.0).
3. Loss = MSE.
4. Loop manual: calcular pred, loss, gradientes (con GradientTape — anticipo clase 108), update con assign_sub.
5. Reportar w, b finales, comparar con verdaderos.

Criterio de aceptación: tras 500 iteraciones con LR=0.01, w ≈ [1, 2] (±0.1) y b ≈ 3 (±0.1).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
InvalidArgumentError: cannot compute Add a	dtype mismatch. Fix: tf.cast(x, tf.float32
AttributeError: 'EagerTensor' object has n	Dentro de @tf.function los tensores son si
Asignar a tf.constant	Inmutable. Fix: usar tf.Variable.
Operaciones tipo arr[mask] = 0 en TF	TF tensors son inmutables. Fix: tf.where(m
t.numpy() lento dentro de un loop de entre	Forza sync GPU → CPU. Fix: mantener todo e

Preguntas frecuentes

¿TF tensors o NumPy arrays?

TF cuando trabajás con un modelo y querés que las operaciones corran en GPU + sean diferenciables. NumPy para análisis CPU de datos. Conversión es barata pero no gratis.

¿tf.Variable o tf.constant para datos de entrada?

Constant (los datos no cambian). Variables son para pesos que se entrenan.

¿Por qué float32 y no float64?

GPUs son MUCHO más rápidas en float32 (y aún más en bfloat16). Default ML.

¿Cómo seteo el device manualmente?

with tf.device('/GPU:0'): ... o tf.config.set_visible_devices(...). En la mayoría de los casos no hace falta — TF lo elige bien.

¿Equivalencia con PyTorch?

tf.constant ↔ torch.tensor(..., requires_grad=False). tf.Variable ↔ torch.Parameter o nn.Parameter.

Operaciones casi 1:1.

Referencias

- Géron, cap. 12 — Custom Models and Training with TensorFlow.
- TF Tensor guide.
- TF Variable guide.

Siguiente clase

Clase 119 — Losses, métricas, capas, modelos custom

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb