Neural networks in Keras 3 and TensorFlow

The same ideas the TensorFlow way: Keras 3 layers, `compile` / `fit`, callbacks, and `tf.GradientTape` when you want your own training loop. TensorFlow 2.21 is stable and mostly receives fixes now; Google recommends Keras 3, JAX or PyTorch for new generative-AI work, and Keras 3 code like this also runs on JAX or PyTorch by changing one line.

Open in Google Colab · part of the inPace course library. Slides © DeepLearning.AI (CC BY-SA 2.0); this notebook: inPace (CC BY-SA 4.0).

Run it top to bottom (Runtime → Run all in Colab). Each part says what to look for.

```
# Works in Google Colab as it is (TensorFlow and Keras are already installed
there).
# On your own computer:  pip install tensorflow keras scikit-learn scipy
matplotlib
# One framework per notebook: PyTorch and TensorFlow can clash when loaded into
the same Python process.
import os
os.environ["KERAS_BACKEND"] = "tensorflow"   # Keras 3 also runs on "jax" or
"torch": change it here
os.environ.setdefault("TF_CPP_MIN_LOG_LEVEL", "3")
import numpy as np, matplotlib.pyplot as plt
import tensorflow as tf, keras
keras.utils.set_random_seed(0)
print("TensorFlow", tf.__version__, "| Keras", keras.__version__, "| backend:",
keras.backend.backend(),
      "| GPU:", bool(tf.config.list_physical_devices("GPU")))
```

1. Data: two moons

```
from sklearn.datasets import make_moons
X, y = make_moons(1000, noise=0.2, random_state=0)
X_tr, y_tr, X_dv, y_dv = X[:800].astype("float32"), y[:800],
X[800:].astype("float32"), y[800:]
```

2. One hidden layer with `fit`

```
model = keras.Sequential([keras.Input((2,)), keras.layers.Dense(8,
activation="tanh"), keras.layers.Dense(1)])
model.compile(optimizer=keras.optimizers.SGD(1.0),
loss=keras.losses.BinaryCrossentropy(from_logits=True),
              metrics=[keras.metrics.BinaryAccuracy(threshold=0.0)])
hist = model.fit(X_tr, y_tr, epochs=80, batch_size=800, verbose=0,
validation_data=(X_dv, y_dv))
print("dev accuracy:", round(hist.history["val_binary_accuracy"][-1], 3))
plt.plot(hist.history["loss"], label="train"); plt.plot(hist.history["val_loss"],
label="dev"); plt.legend(); plt.title("loss"); plt.show()
model.summary()
```

3. The same training with your own loop: `tf.GradientTape`

The tape records the forward pass; `tape.gradient` gives every gradient (TensorFlow's autograd).

```
W1 = tf.Variable(tf.random.normal((2, 8), stddev=0.5)); b1 =
tf.Variable(tf.zeros(8))
W2 = tf.Variable(tf.random.normal((8, 1), stddev=0.5)); b2 =
tf.Variable(tf.zeros(1))
params = [W1, b1, W2, b2]
opt = keras.optimizers.SGD(1.0)
Xt, yt = tf.constant(X_tr), tf.constant(y_tr.reshape(-1, 1), tf.float32)
@tf.function                        # compiles the step into a fast graph
def step():
    with tf.GradientTape() as tape:
        logits = tf.tanh(Xt @ W1 + b1) @ W2 + b2
        loss = tf.reduce_mean(tf.nn.sigmoid_cross_entropy_with_logits(yt, logits))
    opt.apply_gradients(zip(tape.gradient(loss, params), params))
    return loss
for i in range(3000): loss = step()
pred = (tf.tanh(X_dv @ W1 + b1) @ W2 + b2).numpy().ravel() > 0
print(f"final loss {loss.numpy():.3f} | dev accuracy {(pred == y_dv).mean():.3f}")
```

4. A deeper network on digits

```
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
d = load_digits()
Xa, Xb, ya, yb = train_test_split((d.data / 16.0).astype("float32"), d.target,
test_size=0.25, random_state=0)
deep = keras.Sequential([keras.Input((64,)), keras.layers.Dense(128,
activation="relu"),
                         keras.layers.Dense(64, activation="relu"),
keras.layers.Dense(10)])
deep.compile("adam", keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=["accuracy"])
deep.fit(Xa, ya, epochs=40, batch_size=64, verbose=0)
print(f"parameters: {deep.count_params():,} | test accuracy {deep.evaluate(Xb, yb,
verbose=0)[1]:.3f}")
```

Try this

1. Change `KERAS_BACKEND` in the first cell to `"jax"` (Colab has JAX) and run sections 1, 2 and 4 again: the same code, another engine.
2. In section 3, remove `@tf.function`. How much slower is the loop?