# Import necessary libraries

import tensorflow as tf

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

import datetime

# Load and preprocess the Iris dataset

data = load\_iris()

X, y = data.data, data.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Convert labels to categorical format

y\_train = tf.keras.utils.to\_categorical(y\_train, num\_classes=3)

y\_test = tf.keras.utils.to\_categorical(y\_test, num\_classes=3)

# Build the model

model = tf.keras.Sequential([

tf.keras.layers.InputLayer(input\_shape=(4,)),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dense(32, activation='relu'),

tf.keras.layers.Dense(16, activation='relu'),

tf.keras.layers.Dense(3, activation='softmax')

])

# Compile the model

model.compile(optimizer=tf.keras.optimizers.Adam(learning\_rate=0.001),

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Set up TensorBoard callback

log\_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")

tensorboard\_callback = tf.keras.callbacks.TensorBoard(log\_dir=log\_dir, histogram\_freq=1)

# Train the model with TensorBoard callback

model.fit(X\_train, y\_train,

validation\_data=(X\_test, y\_test),

epochs=100,

batch\_size=8,

callbacks=[tensorboard\_callback])

# Evaluate the model

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f"Test Accuracy: {accuracy:.4f}")

# Instructions to run TensorBoard in Google Colab

# Load the TensorBoard extension

%load\_ext tensorboard

# Launch TensorBoard

%tensorboard --logdir logs/fit

