Lab 10 - TensorFlow Hello World

Given the following code:

https://github.com/Univ-Wyo-Education/F21-1010/blob/main/class/lect/Lect-21/lab-09_start.py

```python
from __future__ import absolute_import, division, print_function, unicode_literals

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras

# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

# print(tf.__version__)

fashion_mnist = keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()

class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
               'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']

plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.grid(False)
plt.show()

# Scale these values to a range of 0 to 1 before feeding them to the neural
# network model. To do so, divide the values by 255. It's important that the
# training set and the testing set be preprocessed in the same way:

train_images = train_images / 255.0

test_images = test_images / 255.0

# To verify that the data is in the correct format and that you're ready to
# build and train the network, let's display the first 25 images from the
# training set and display the class name below each image.

plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
```

```
        plt.yticks([])
        plt.grid(False)
        plt.imshow(train_images[i], cmap=plt.cm.binary)
        plt.xlabel(class_names[train_labels[i]])
    plt.show()


model = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(128, activation='relu'),
        keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
                loss='sparse_categorical_crossentropy',
                metrics=['accuracy'])

model.fit(train_images, train_labels, epochs=10)


test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)

print('\nTest accuracy:', test_acc)
```

Get this to run.

Run it multiple times and check that you get slightly different results each time.

Turn in a text file with at least 2 runs with the accuracy from the runs.