1 Assignment 1: Introduction to TensorFlow and Keras

Objective:

Install TensorFlow and Keras

Verify the installation

Load a dataset

Build and train a simple model

[1]:
```
# Step 1: Install TensorFlow (uncomment if needed)
# !pip install tensorflow
```

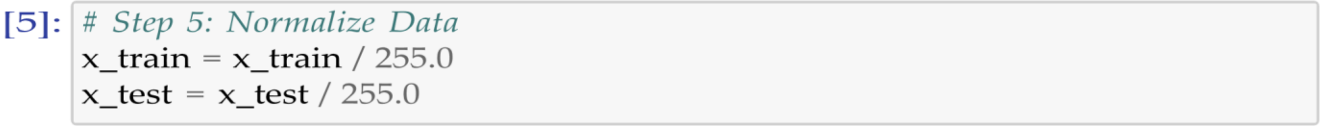
[2]:
```
# Step 2: Import Libraries
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
# Step 3: Print TensorFlow and Keras Versions
print("TensorFlow version:", tf._version_)
print("Keras version:", keras._version_)
```

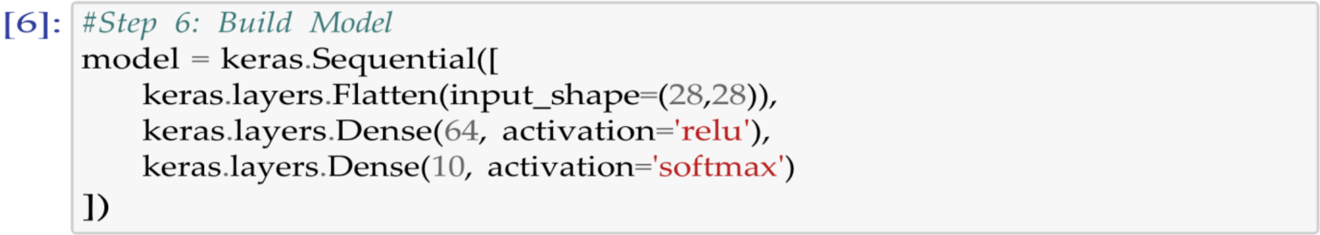
```
TensorFlow version: 2.18.0
Keras version: 3.8.0
```

[4]:
```
# Step 4: Load MNIST dataset
mnist = keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
print("Train shape:", x_train.shape)
print("Test shape:", x_test.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras- datasets/mnist.npz
 11490434/11490434                    0s
0us/step
Train shape: (60000, 28, 28)
Test shape: (10000, 28, 28)
```

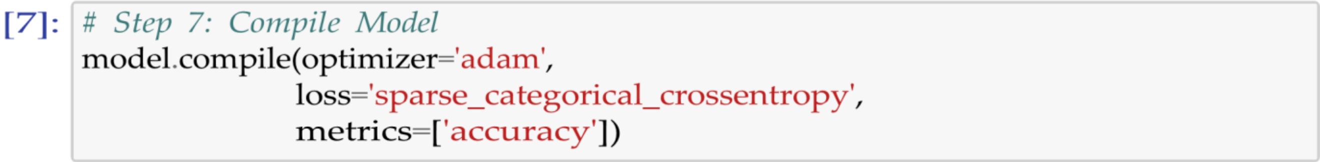


```
[5]: # Step 5: Normalize Data
     x_train = x_train / 255.0
     x_test = x_test / 255.0
```

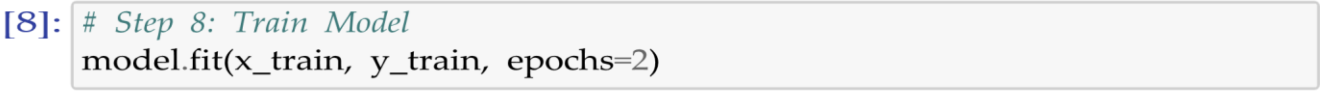


```
[6]: #Step 6: Build Model
     model = keras.Sequential([
         keras.layers.Flatten(input_shape=(28,28)),
         keras.layers.Dense(64, activation='relu'),
         keras.layers.Dense(10, activation='softmax')
     ])
```

```
/usr/local/lib/python3.11/dist-    packages/keras/src/layers/reshaping/flatten.py:37: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models,
prefer using an `Input(shape)` object as the first layer in the model instead. super()._init_(**kwargs)
```



```
[7]: # Step 7: Compile Model
     model.compile(optimizer='adam',
                   loss='sparse_categorical_crossentropy',
                   metrics=['accuracy'])
```

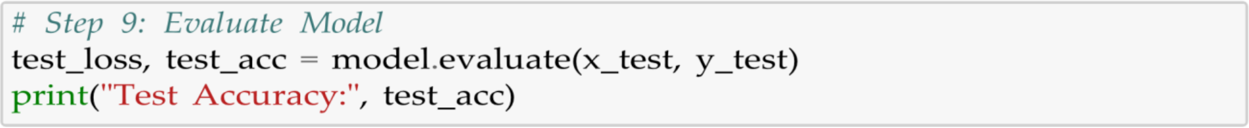


```
[8]: # Step 8: Train Model
     model.fit(x_train, y_train, epochs=2)
```

```
Epoch 1/2
1875/1875                    7s 3ms/step -
accuracy: 0.8612 - loss: 0.4987
Epoch 2/2
1875/1875                    5s 3ms/step -
accuracy: 0.9544 - loss: 0.1571
```

```
[8]: <keras.src.callbacks.history.History at 0x7e434222ea90>
```

[9]:



```
# Step 9: Evaluate Model
test_loss, test_acc = model.evaluate(x_test, y_test)
print("Test Accuracy:", test_acc)
```

```
313/313                    1s 4ms/step -
accuracy:   0.9594   -   loss:   0.1346   Test
Accuracy: 0.9635000228881836
```

This notebook introduces TensorFlow and Keras by building a simple neural network to classify handwritten digits from the MNIST dataset. It includes steps for loading and normalizing data, defining and training a model, and evaluating its accuracy. The model achieves predictions using a basic feedforward architecture. Visualizations help verify both data and model performance.

1.1 Instruction to Student Instructions:

- Run each cell step by step.
- Write 2-3 lines **explaining what happens in each step.**
- Take a screenshot of the final accuracy output.
- Download your notebook as .ipynb and as PDF.
- Submit both files.