Sadiqa Ilyas Mnist Assignment

```
#Loading fashion_mnist dataset with Keras
from tensorflow.keras.datasets import fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/tra
32768/29515 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/tra
26427392/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t1
8192/5148 [===============================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t1
4423680/4422102 [==============================] - 0s 0us/step
```

```
print(len(train_images))
print(len(train_labels))
```

```
60000
60000
```

```
print(train_images.shape)
print(train_labels.shape)
print(test_images.shape)
print(test_labels.shape)
```

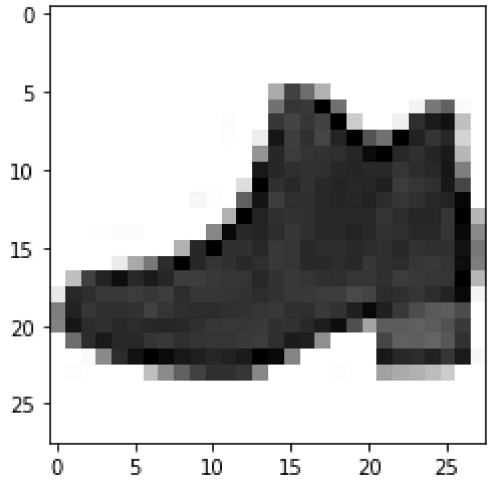
```
(60000, 28, 28)
(60000,)
(10000, 28, 28)
(10000,)
```

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

```
['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker',
```

```
# Image of Fashion_MNIST
import matplotlib.pyplot as plt
num = 15
item=train_labels[num]
print("Class Label:", class_names[item] )
plt.imshow(train_images[num],cmap=plt.cm.binary)
plt.show()
```

```
Class Label: Ankle boot
```



```
from keras import models
from keras import layers
import tensorflow as tf


model = tf.keras.Sequential()
model.add(layers.Dense(510,activation='relu',input_shape=(28*28,)))
model.add(layers.Dense(10,activation='softmax'))
model.add(tf.keras.layers.Flatten(input_shape=(28,28)))


# compilation
model.compile(optimizer="rmsprop", loss="categorical_crossentropy", metrics=["accuracy"])


#Preparing the image data
train_images=train_images.reshape((60000, 28*28))
train_images=train_images.astype("float32")/255


test_images=test_images.reshape((10000, 28*28))
test_images=test_images.astype("float32")/255


#One Hot Coding
from tensorflow.keras.utils import to_categorical
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)


#Train Model
model.fit(train_images, train_labels, epochs=10, batch_size=128)
```

```
Epoch 1/10
469/469 [==============================] - 4s 8ms/step - loss: 0.7466 - accuracy: 0.7
Epoch 2/10
469/469 [==============================] - 4s 8ms/step - loss: 0.4000 - accuracy: 0.8
Epoch 3/10
469/469 [==============================] - 4s 8ms/step - loss: 0.3378 - accuracy: 0.8
Epoch 4/10
469/469 [==============================] - 4s 8ms/step - loss: 0.3104 - accuracy: 0.8
Epoch 5/10
```

```
469/469 [==============================] - 4s 8ms/step - loss: 0.2903 - accuracy: 0.8
Epoch 6/10
469/469 [==============================] - 4s 8ms/step - loss: 0.2771 - accuracy: 0.8
Epoch 7/10
469/469 [==============================] - 4s 8ms/step - loss: 0.2648 - accuracy: 0.9
Epoch 8/10
469/469 [==============================] - 4s 8ms/step - loss: 0.2489 - accuracy: 0.9
Epoch 9/10
469/469 [==============================] - 4s 8ms/step - loss: 0.2432 - accuracy: 0.9
Epoch 10/10
469/469 [==============================] - 4s 8ms/step - loss: 0.2378 - accuracy: 0.9
<tensorflow.python.keras.callbacks.History at 0x7fdd0a594490>
```

```
#Check Model performance of Test Data
test_loss, test_acc = model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.3951 - accuracy: 0.8
```

```
print("Accuracy: ",test_acc,"\nTest Loss Value:",test_loss)
test_labels[7]
```

```
Accuracy:  0.8755999803543091
Test Loss Value: 0.39510831236839294
array([0., 0., 0., 0., 0., 0., 1., 0., 0., 0.], dtype=float32)
```