

# Day 62

## DIY

### Q1. Problem Statement: Image Classification using Tensorflow

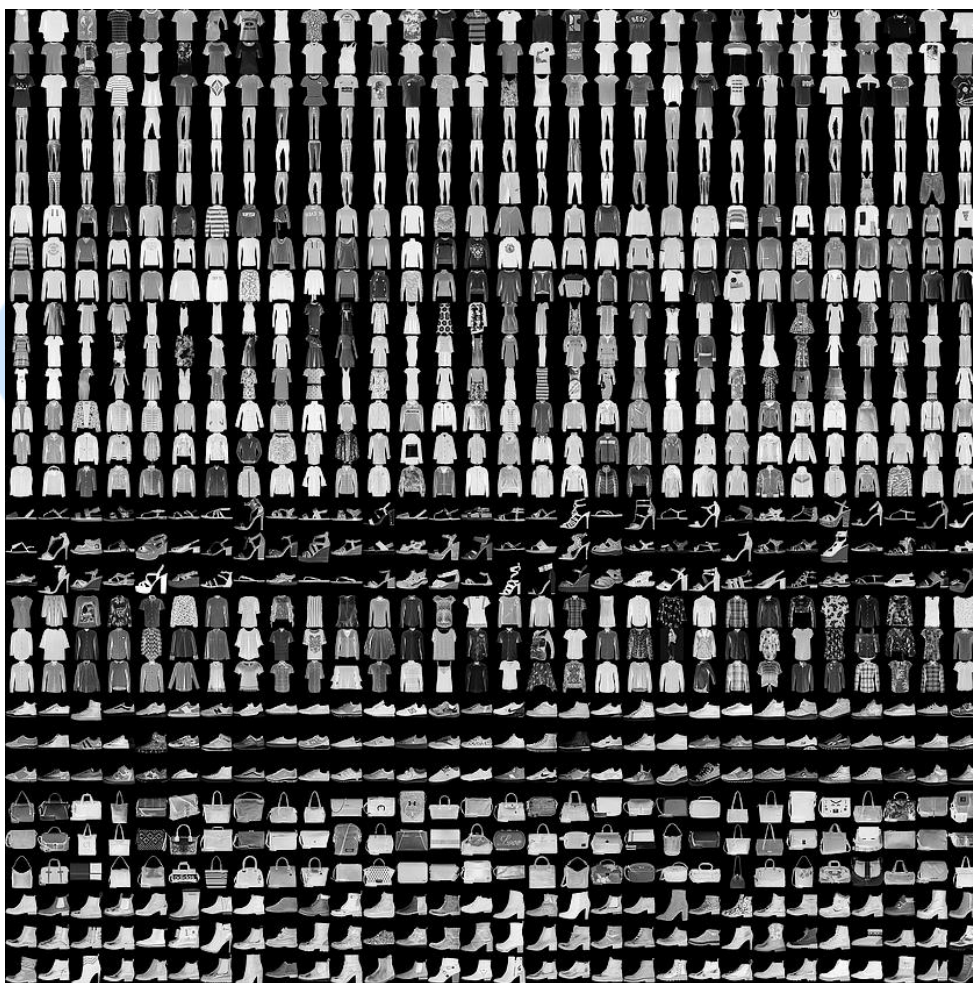
This problem uses the '*Fashion MNIST*' dataset, which contains 70,000 images of 10 categories of apparel. The images show individual articles of clothing at low resolution (28 by 28 pixels), as shown in the Dataset section below.

Import the '*fashion\_mnist*' dataset using the `tf.keras.datasets.fashion_mnist()` command and perform the following tasks:

1. Download and import Tensorflow and other required libraries
2. Import the MNIST fashion test and train data using keras along with the train and test labels
3. Define the class names as - ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot'] for each category of articles
4. Analyze the train and test data by printing their respective sizes
5. Scale the test and train image datasets by dividing with the same number
6. Display the scaled data using matplotlib
7. Define a Keras model with a '*relu*' (Rectified Linear Unit) activation function

8. Train the model using the train dataset
9. Test the model and print the test accuracy
10. Predict to test the model
11. Define a function that uses the model to predict whether it's a shoe, shirt, pant, ....etc. and test the model by using a sample image from the test dataset

### Dataset:



### Sample Output:

1. Download and import Tensorflow and other required libraries

```

Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.8.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: tensorboard<2.9,>=2.8 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.8.0)
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.0.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.17.3)
Requirement already satisfied: gast>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.5.3)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.15.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.44.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (4.2.0)
Requirement already satisfied: keras<2.9,>=2.8.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.8.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (13.0.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tensorflow) (57.4.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.24.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Collecting tf-estimator-nightly==2.8.0.dev2021122109
  Downloading tf_estimator_nightly-2.8.0.dev2021122109-py2.py3-none-any.whl (462 kB)
    462 kB 4.2 MB/s
Requirement already satisfied: flatbuffers>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.21.6)

```

## 2. Import the 'MNIST fashion' test and train data using keras along with the train and test labels

```

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
32768/29515 [=====] - 0s 0us/step
40960/29515 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26427392/26421880 [=====] - 0s 0us/step
26435584/26421880 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
16384/5148 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4423680/4422102 [=====] - 0s 0us/step
4431872/4422102 [=====] - 0s 0us/step

```

## 3. Define the class names as - ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot'] for each category of articles

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

4. Analyze the train and test data by printing their respective sizes

```
Size of train data is:  
(60000, 28, 28)
```

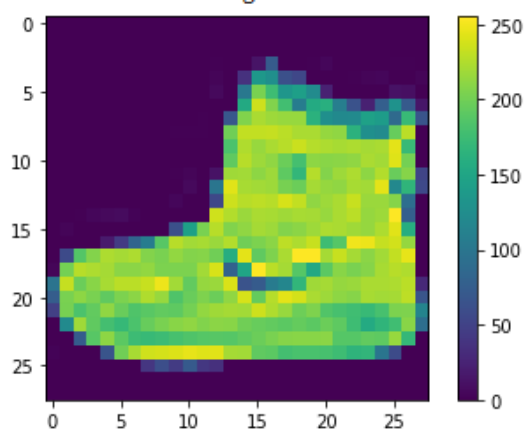
```
Size of train labels is:  
60000
```

```
Size of test data is:  
(10000, 28, 28)
```

```
Size of test labels is:  
10000
```

5. Scale the test and train image datasets by dividing with the same number

Data before scaling:



6. Display the scaled data using matplotlib



7. Define a Keras model with a 'relu' (Rectified Linear Unit) activation function

```
<keras.engine.sequential.Sequential at 0x7fe46382ce50>
```

8. Train the model using the train dataset

```
Epoch 1/10
1875/1875 [=====] - 4s 2ms/step - loss: 2.9329 - accuracy: 0.6939
Epoch 2/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.6828 - accuracy: 0.7589
Epoch 3/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.5999 - accuracy: 0.7890
Epoch 4/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.5222 - accuracy: 0.8231
Epoch 5/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.5022 - accuracy: 0.8313
Epoch 6/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.4818 - accuracy: 0.8365
Epoch 7/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.4803 - accuracy: 0.8383
Epoch 8/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.4736 - accuracy: 0.8410
Epoch 9/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.4655 - accuracy: 0.8455
Epoch 10/10
1875/1875 [=====] - 3s 2ms/step - loss: 0.4616 - accuracy: 0.8444
<keras.callbacks.History at 0x7fe0e944d2d0>
```

9. Test the model and print the test accuracy

```
Test accuracy: 0.805899977684021
```

## 10. Predict to test the model

```
array([7.20947685e-27, 4.61645405e-18, 0.00000000e+00, 5.82805191e-24,  
       0.00000000e+00, 5.47640584e-03, 2.83602250e-25, 1.04442565e-02,  
       9.09305089e-17, 9.84079301e-01], dtype=float32)
```

## 11. Define a function that uses the model to predict whether it's a shoe, shirt, pant,...etc. and test the model by using a sample image from the test dataset

