

HOMEWORK 5 REPORT

Ques 1.

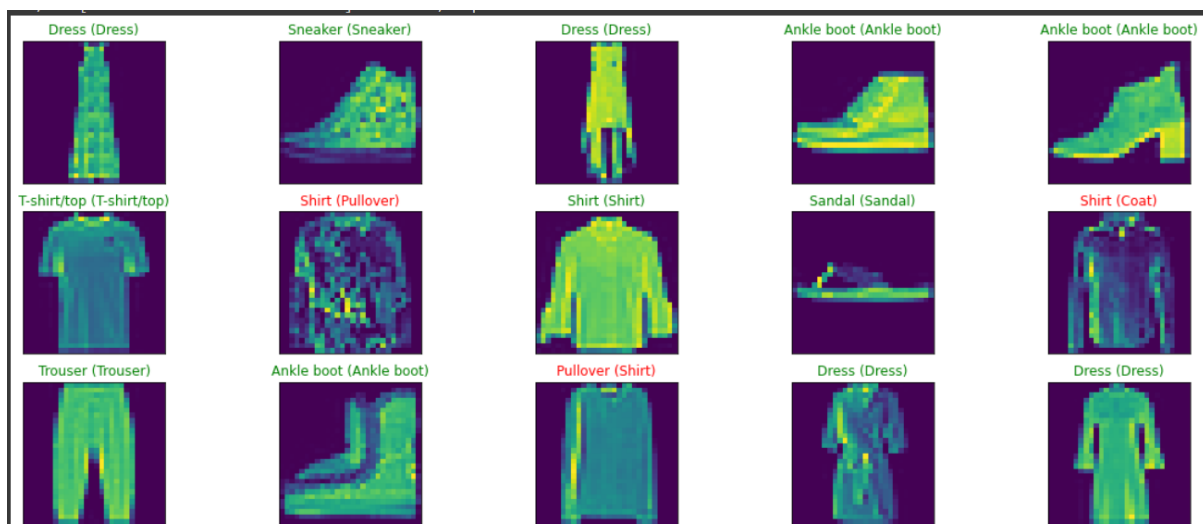
In this question, we can observe that adding a convolutional layer along with relu and a dropout gives better test accuracy. Although when we see the classification output in terms of images you can see how the misclassified and the correctly classified outputs.

```
Epoch 15: val_loss did not improve from 0.23182
860/860 [=====] - 91s 106ms/step - loss: 0.3126 - accuracy: 0.8847 - val_loss: 0.2342 - val_accuracy: 0.9154
Epoch 16/20
859/860 [=====>.] - ETA: 0s - loss: 0.3085 - accuracy: 0.8869
Epoch 16: val_loss improved from 0.23182 to 0.22962, saving model to model.weights.best.hdf5
860/860 [=====] - 95s 111ms/step - loss: 0.3085 - accuracy: 0.8869 - val_loss: 0.2296 - val_accuracy: 0.9124
Epoch 17/20
860/860 [=====] - ETA: 0s - loss: 0.3075 - accuracy: 0.8893
Epoch 17: val_loss improved from 0.22962 to 0.22878, saving model to model.weights.best.hdf5
860/860 [=====] - 91s 106ms/step - loss: 0.3075 - accuracy: 0.8893 - val_loss: 0.2288 - val_accuracy: 0.9148
Epoch 18/20
859/860 [=====>.] - ETA: 0s - loss: 0.3048 - accuracy: 0.8877
Epoch 18: val_loss did not improve from 0.22878
860/860 [=====] - 90s 105ms/step - loss: 0.3050 - accuracy: 0.8876 - val_loss: 0.2296 - val_accuracy: 0.9150
Epoch 19/20
860/860 [=====] - ETA: 0s - loss: 0.3038 - accuracy: 0.8881
Epoch 19: val_loss improved from 0.22878 to 0.22877, saving model to model.weights.best.hdf5
860/860 [=====] - 92s 108ms/step - loss: 0.3038 - accuracy: 0.8881 - val_loss: 0.2288 - val_accuracy: 0.9130
Epoch 20/20
859/860 [=====>.] - ETA: 0s - loss: 0.3065 - accuracy: 0.8862
Epoch 20: val_loss did not improve from 0.22877
860/860 [=====] - 90s 105ms/step - loss: 0.3064 - accuracy: 0.8862 - val_loss: 0.2370 - val_accuracy: 0.9104
<keras.callbacks.History at 0x7fcd376c670>
```

```
✓ [17] 1 model.load_weights ('model.weights.best.hdf5')
6s 2 score = model.evaluate(x_test,y_test,verbose = 0)
3 print("\n",'test losses',score[0])
4 print("\n",'test accuracy',score[1])
```

test losses 0.24865756928920746

test accuracy 0.9083999991416931



Ques 2

Here we have used PyTorch with the help pf which we pre-trained and added Convolutional and Relu to train our model and achieve better accuracy. We have used **ADAM optimizer** with following parameters:

- Learning rate = 0.0005
- Weight decay = 4e-5 (L2 penalty)

The **hyperparameters** are:

- Learning rate = 0.0005
- Batch size = 8
- Number of epochs = 20

Furthermore, due to computational limitations, for partition of the data into training, validation and testing, we have created a python file called “Wrapper” which would create .npy files for X_train, y_train, X_Val, y_val, X_test and y_test. This helped us train our final code in python file named “ResNet50”.

RMSE losses –

Training: [17.83030536 14.53167456 13.39173478 12.90225339 11.80618891 11.40607023 10.17670815 9.41862352 9.01958858 8.1717321 7.99084143 7.72224851 7.75391551 7.40045151 7.2985783 7.35081178 7.1979003 6.99527553 6.64132012 6.70200313]

Validation: [16.64563913 17.02206461 13.65341548 14.41499651 13.32973921 14.81964601 13.66807969 13.31877572 14.79746835 13.80046012 13.76470854 13.8190339 12.92743897 12.90926331 12.73436824 12.90561434 12.80094331 12.94288752 13.30139966 12.76232665]

Testing Accuracy: 12.767920696597056