Project 2

Task-

To train Multi Layer Network and Convolutional Neural Network on the FASHION MNIST dataset and predicting classes out of the 10 clothing items.

Approach for training-

The training dataset of 60000 images was divided into 50000 as training set and 10000 as validation set for testing the model. Mini Batch based Optimization algorithm was used and the data was shuffled before each epoch. The loss was calculated for the mini batch and propagated and the accuracy was test on the validation set. After testing on the validation set , if similar error loss was generated then the same model was tested on the test set.

Hyperparameters used-

1)MLP-

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                  [-1, 150]         117,750
            Linear-2                  [-1, 100]          15,100
       BatchNorm1d-3                  [-1, 100]             200
            Linear-4                   [-1, 50]           5,050
            Linear-5                   [-1, 10]             510
================================================================
Total params: 138,610
Trainable params: 138,610
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.53
Estimated Total Size (MB): 0.53
----------------------------------------------------------------
```

Architecture of the Network- 4 layers were created and then the loss and the accuracy on the dataset was checked keeping other things same and no significant improvement was seen on the accuracy and loss , so to keep the computational complexity low due to learning of larger number of weights 4 layers were fixed.

Batchsize- MiniBatch was used to accommodate for the higher variance in the single sized gradient descent. The size was varied from 8 to 64 and difference in the training loss and testing loss was decreased on increasing the minibatch size as the weight update was performed after seeing the impact of larger number of samples. Batch sizes were kept small so as to add some regularizing noise and decrease the generalisation error.

Optimization Algorithm- Adam was used as an optimization algorithm because of its generalized faster convergence rates. The learning rate was varied from 0.01 to 0.001 to 0.0001. It was noticed that the lesser learning rates meant that the loss was showing a large variability in the behaviour even for a large number of epochs and was cycling between a larger and a smaller value periodically.

The rate was thus decreased to 0.0001 at which the variability was still there but it was decreasing with increasing number of epochs.

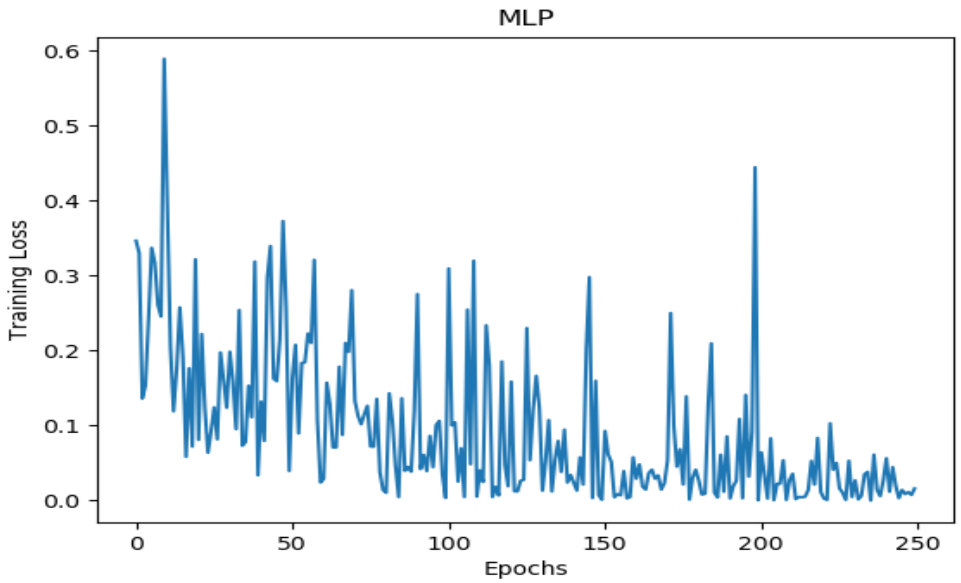
SGD with momentum was also tried but not much significant increase was observed on keeping the other parameters as same.

Other parameters- Batch Normalization was used in 1 layer to introduce regularizing effect and decrease the generalization error.

Findings-

Accuracy of 88.6% was achieved on the test set and the training loss was of the order of 0.01. The test loss was of the order of 0.1. The accuracy on the validation set was 89.5%. Thus the model was not overfitting the data and performing equally well on the unseen validation and the test sets.

Observations- As the epochs were increased the variability in the training loss was also decreased

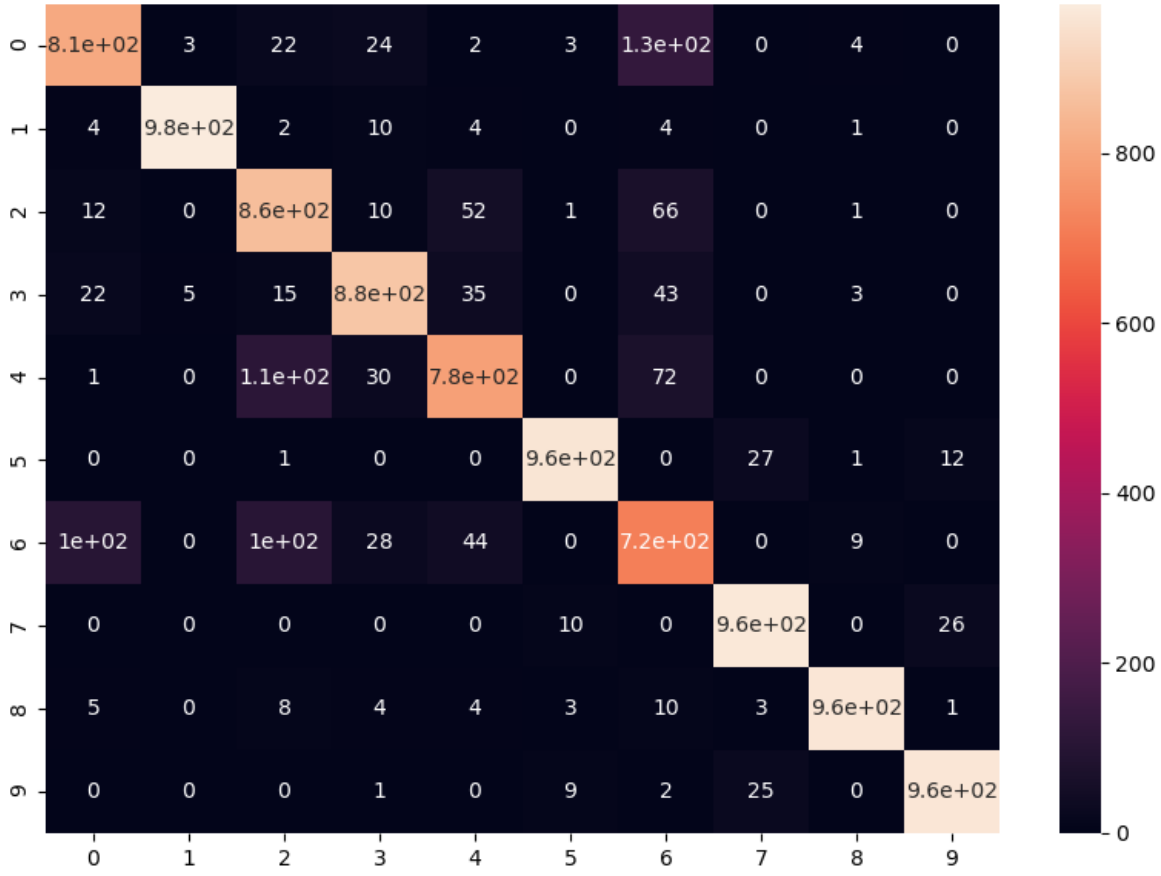


We infer from the confusion matrix that class 6 has the least true positive rate and often class 0 and class 2 are misclassified as class 6. Both the entries for the (6,0) and (0,6) are higher suggesting that the network has high confusion classifying either one of them and often misclassifies one as the other (namely shirt and t-shirt.).

Class 2 has least positive predictive value and often class 2is misclassified as class 4 and class 6.

Class 1 has highest positive predictive value and is least misclassified as anything else.

Class 7,8,9 has high true positive rate ,that is they are least misclassified as something other.


0 8.1e+02 3 22 24 2 3 1.3e+02 0 4 0
1 4 9.8e+02 2 10 4 0 4 0 1 0
2 12 0 8.6e+02 10 52 1 66 0 1 0
3 22 5 15 8.8e+02 35 0 43 0 3 0
4 1 0 1.1e+02 30 7.8e+02 0 72 0 0 0
5 0 0 1 0 0 9.6e+02 0 27 1 12
6 1e+02 0 1e+02 28 44 0 7.2e+02 0 9 0
7 0 0 0 0 0 10 0 9.6e+02 0 26
8 5 0 8 4 4 3 10 3 9.6e+02 1
9 0 0 0 1 0 9 2 25 0 9.6e+02
0 1 2 3 4 5 6 7 8 9
800
600
400
200
0

2)CNN-

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 4, 28, 28]              40
            Conv2d-2           [-1, 12, 14, 14]             444
       BatchNorm2d-3           [-1, 12, 14, 14]              24
            Linear-4                 [-1, 588]         346,332
       BatchNorm1d-5                 [-1, 588]           1,176
            Linear-6                 [-1, 100]          58,900
            Linear-7                  [-1, 10]           1,010
================================================================
Total params: 407,926
Trainable params: 407,926
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.07
Params size (MB): 1.56
Estimated Total Size (MB): 1.63
----------------------------------------------------------------
```

Architecture of the Network- 2 convolution layers were created with 2 downsampling layers using Maxpool. Window size and padding was varied so that no significant information is lost as a result of convolution. Loss was decreased when a padding of 1 was used with both of the convolution filter layers. The features were then flattened and then the loss and the accuracy on the dataset was checked keeping other things same and no significant improvement was seen on the accuracy and loss , so to keep the computational complexity low due to learning of larger number of weights layers were not increased.

Batchsize- MiniBatch was used to accommodate for the higher variance in the single sized gradient descent. The size was varied from 8 to 64 and difference in the training loss and testing loss was decreased on increasing the minibatch size as the weight update was performed after seeing the impact of larger number of samples. Batch sizes were kept small so as to add some regularizing noise and decrease the generalisation error.

Optimization Algorithm- Adam was used as an optimization algorithm because of its generalized faster convergence rates. The learning rate was varied from 0.01 to 0.001 to 0.0001. It was noticed that the lesser learning rates meant that the loss was showing a large variability in the behaviour even for a large number of epochs and was cycling between a larger and a smaller value periodically. The rate was thus decreased to 0.0001 at which the variability was still there but it was decreasing with increasing number of epochs.

SGD with momentum was also tried but not much significant increase was observed on keeping the other parameters as same.

Other parameters- Batch Normalization was used in 2 layers to introduce regularizing effect and decrease the generalization error.

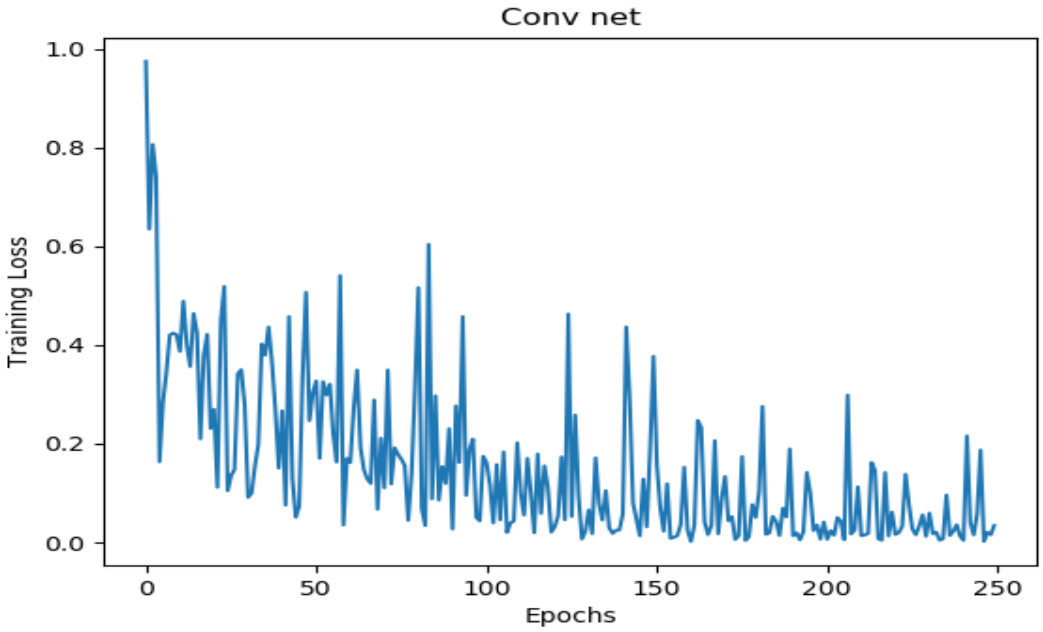
Findings- Accuracy of 90.31 was observed on the test set and 91.2 on the validation set , thus not showing much of the variability. Training loss was in the order of 0.01 and Test loss was in the order of 0.1. Thus the model was not overfitting the data and performing equally well on the unseen validation and the test sets.

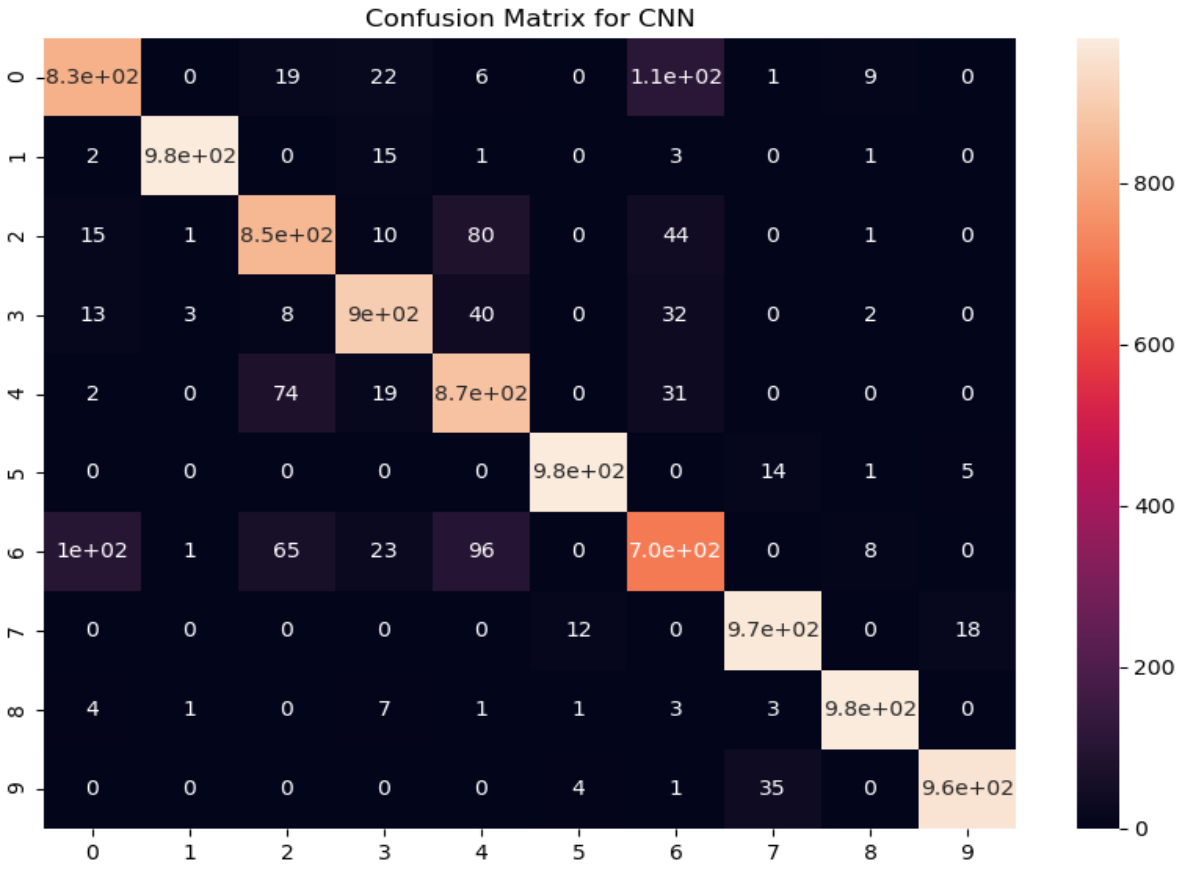
Observations-As the epochs are increased the variability in the loss is decreased.

From the confusion matrix, we infer that again classifier faces issues in distinguishing between class 0 and class 6 namely shirt and t-shirt.

Class 0 and class 6 have the least true positive rate and more often other classes are classifies as the two whereas class 5 and class 8 have the highest true positive rate and less often other classes are classified as them.

Class 1 has highest positive predictive value that is it is least classified as something else.




Confusion Matrix for CNN
0 -8.3e+02 0 19 22 6 0 1.1e+02 1 9 0
1 2 9.8e+02 0 15 1 0 3 0 1 0
2 15 1 8.5e+02 10 80 0 44 0 1 0
3 13 3 8 9e+02 40 0 32 0 2 0
4 2 0 74 19 8.7e+02 0 31 0 0 0
5 0 0 0 0 0 9.8e+02 0 14 1 5
6 1e+02 1 65 23 96 0 7.0e+02 0 8 0
7 0 0 0 0 0 12 0 9.7e+02 0 18
8 4 1 0 7 1 1 3 3 9.8e+02 0
9 0 0 0 0 0 4 1 35 0 9.6e+02
0 1 2 3 4 5 6 7 8 9
800
600
400
200
0