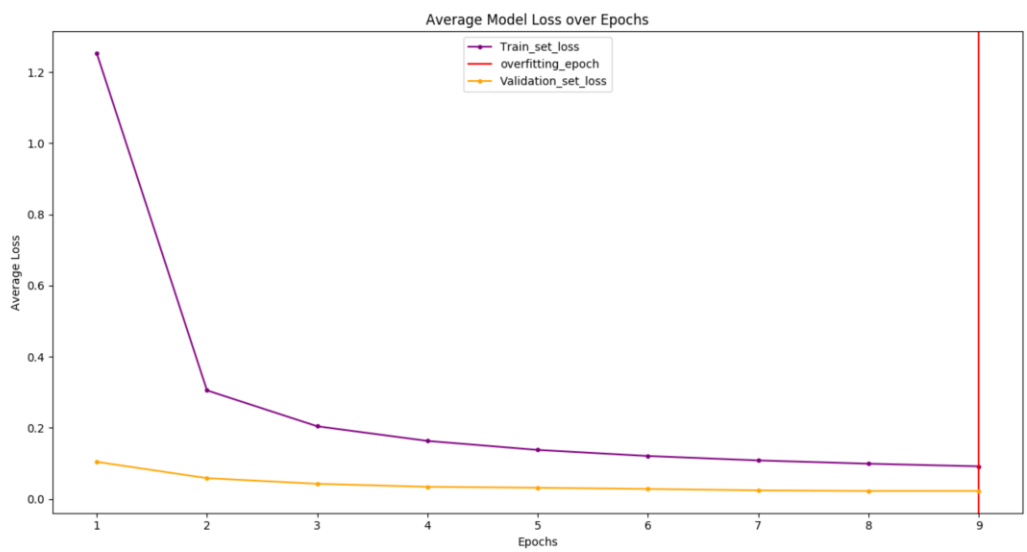
Assignment-3(Machine Learning)

Name: Rajkumar Lenin Pillai

Question 1:

Solution:-

1. Plot of cross-entropy vs. epoch



The red line shows the epoch after which overfitting begins. As we can see from the above diagram the overfitting occurs at epoch 9 . After epoch 8 the error on the validation set increases. The learning curve shows it has a very good learning rate. Initially it begins with high amount of loss but reduces significantly till the 3rd epoch from the 3rd epoch weights keep reducing and the training is converging. So it means after epoch 8 the overfitting begins and to prevent that the model weights are updated only till epoch 8 in program q1.py.

Given below is the output of the program q1.py during training:-

```
Epoch 1, Train_loss: 1.2534150946537654
Epoch 1, Val_loss: 0.10454591814676921
Epoch 2, Train_loss: 0.3054039208889008
Epoch 2, Val_loss: 0.05864247370759646
Epoch 3, Train_loss: 0.2044561408261458
Epoch 3, Val_loss: 0.04277738685409228
Epoch 4, Train_loss: 0.16329040914277235
Epoch 4, Val_loss: 0.03432376619676749
Epoch 5, Train_loss: 0.1379809760674834
Epoch 5, Val_loss: 0.03160546486079693
Epoch 6, Train_loss: 0.12090019050737222
Epoch 6, Val_loss: 0.02830372356623411
Epoch 7, Train_loss: 0.10847834086666505
Epoch 7, Val_loss: 0.02431035769234101
Epoch 8, Train_loss: 0.09924284329513709
Epoch 8, Val_loss: 0.02263870626936356
Epoch 9, Train_loss: 0.0918239643437167
Epoch 9, Val_loss: 0.022804594368984302
Prev loss_validation  0.02263870626936356
Runnning loss_validation  0.022804594368984302
Finished Training
Time to converge 58.116286277770996
```

As you can observe the validation loss at epoch 8 is 0.0226 and at epoch 9 it is 0.0228 which is larger than validation loss at epoch 8. So the model is not trained further and this model weights are stored in file q1_model_weights. The time required to converge the weights was 58.116 seconds.

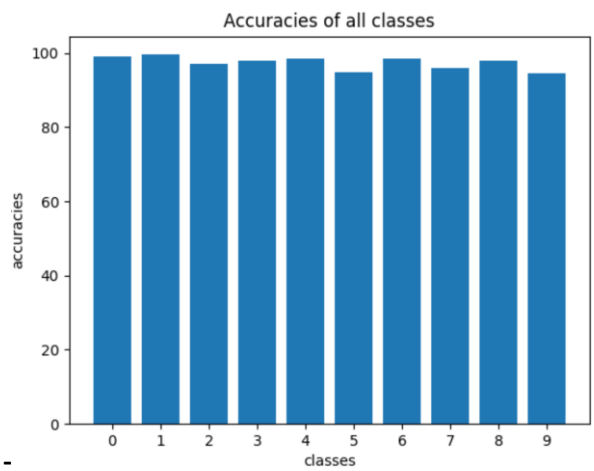
2.) Accuracies

```
Accuracy calculation

Accuracy of the network on the  test images: 97 %

Accuracy of     0 : 99 %  Incorrect samples  8  correct samples  972
Accuracy of     1 : 99 %  Incorrect samples  5  correct samples  1130
Accuracy of     2 : 97 %  Incorrect samples  29  correct samples  1003
Accuracy of     3 : 97 %  Incorrect samples  21  correct samples  989
Accuracy of     4 : 98 %  Incorrect samples  15  correct samples  967
Accuracy of     5 : 94 %  Incorrect samples  47  correct samples  845
Accuracy of     6 : 98 %  Incorrect samples  14  correct samples  944
Accuracy of     7 : 95 %  Incorrect samples  42  correct samples  986
Accuracy of     8 : 97 %  Incorrect samples  20  correct samples  954
Accuracy of     9 : 94 %  Incorrect samples  54  correct samples  955
```

The above results are derived from accuracy calculation in program q1.py

The accuracy of the model is 97% and the individual class accuracies are above 93%.

Plot of Accuracies of all class:

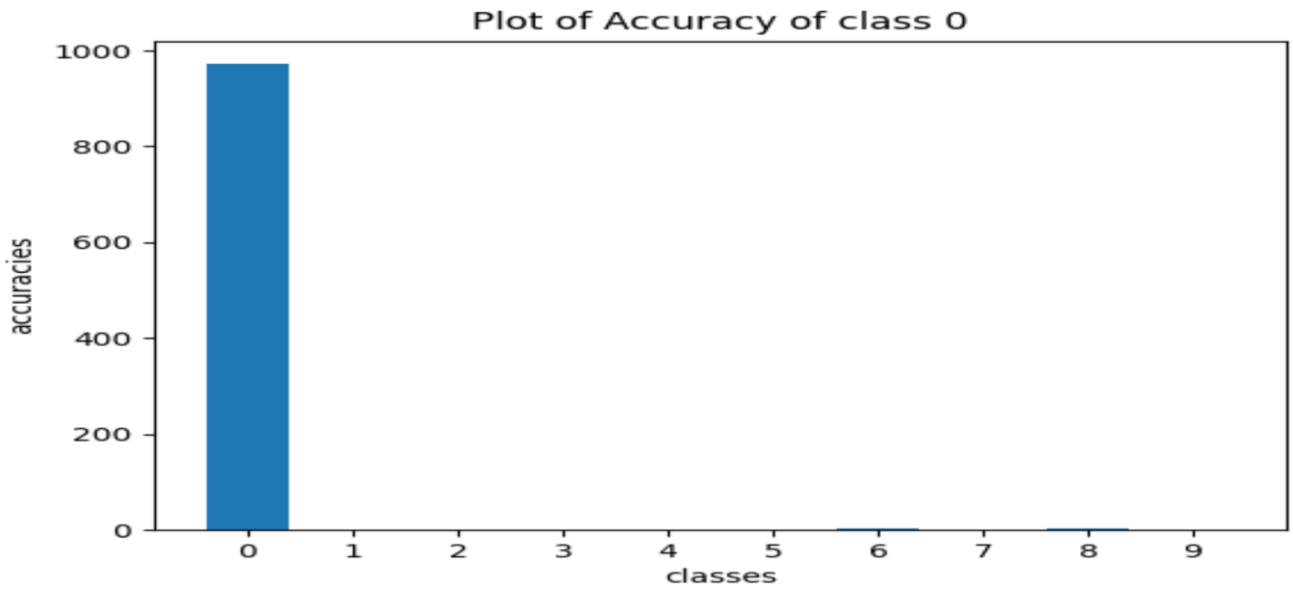


-

From the above plot we can observe that class-0 and class-1 have accuracy 99% and class 9 has lowest accuracy of 94%. So it means that model performs better in predicting the images of digit 1 and 0 and it is less efficient in predicting the image of digit 5.

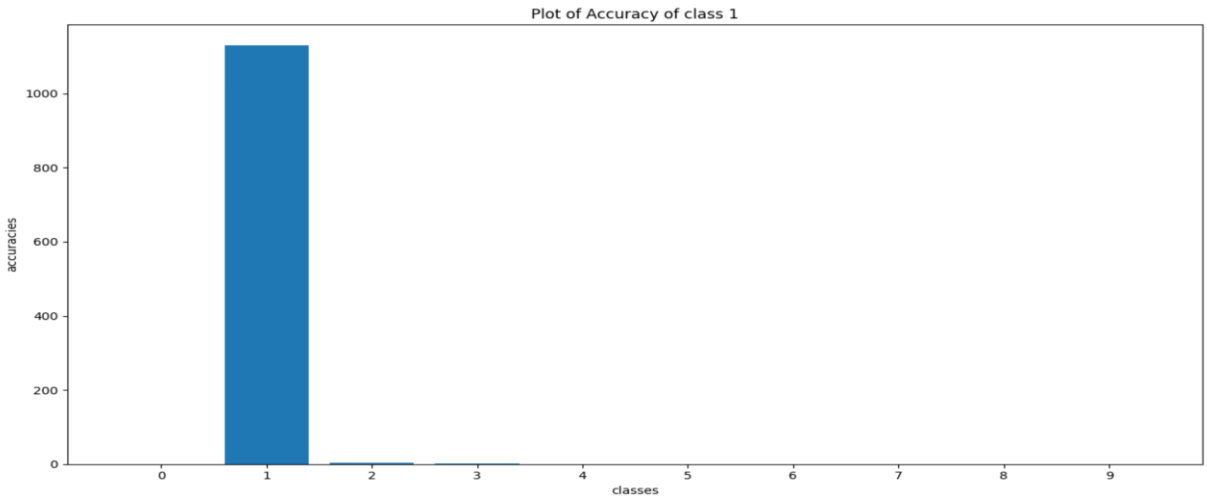
Plot of Accuracies of individual class:-

Plot of accuracy of class – 0



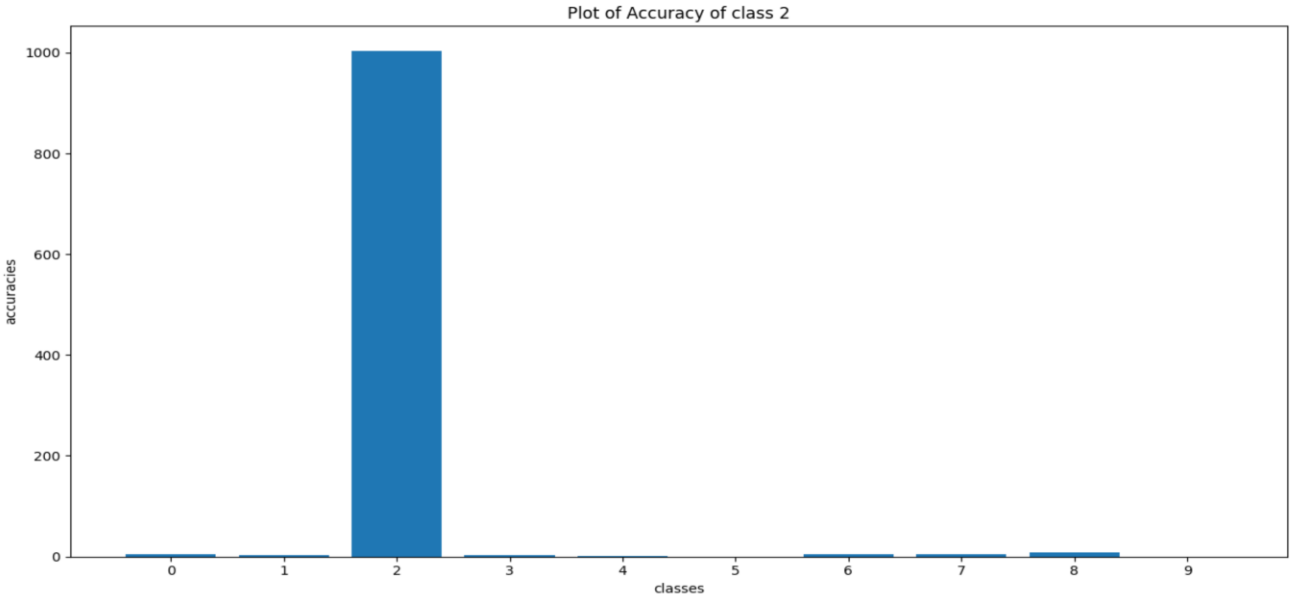
Class-0 has accuracy 99%. If we observe carefully the above plot there is a chance of the model interpreting digit 6 and 7 as zero.

Plot of accuracy of class – 1



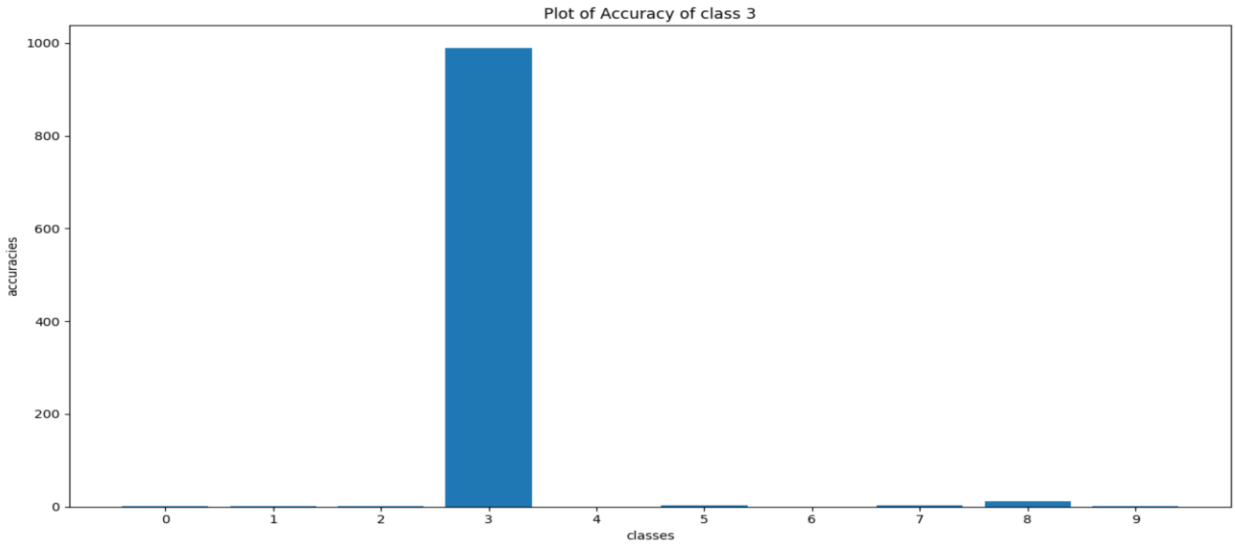
Class-1 has accuracy 99%. If we observe carefully the above plot there is a chance of the model interpreting digit 2 and 3 as one.

Plot of accuracy of class – 2



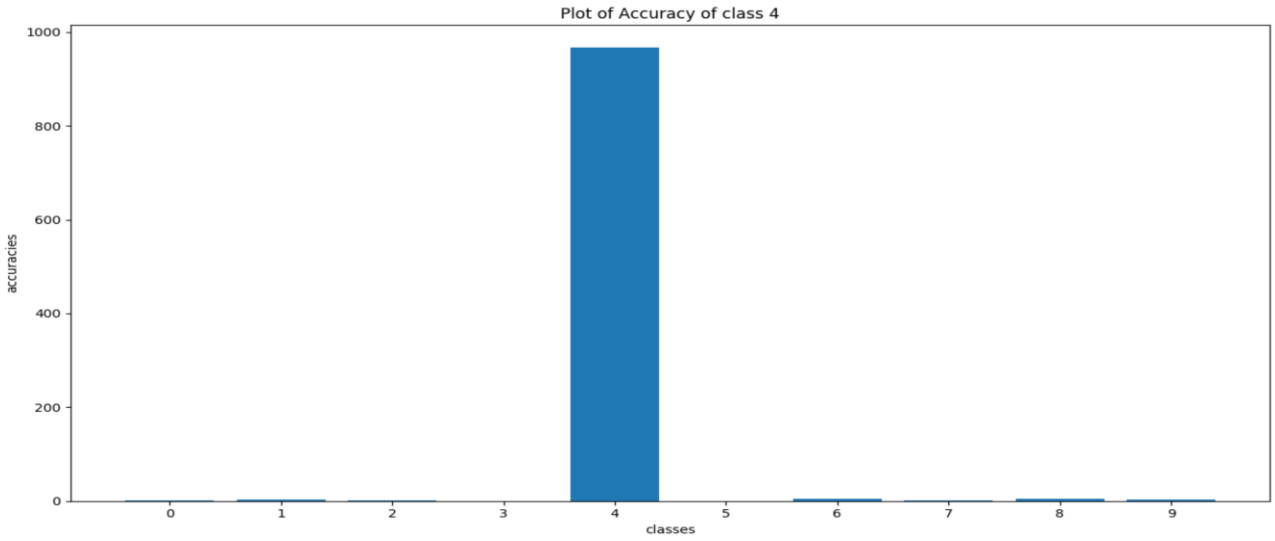
Class-2 has accuracy 97%. If we observe carefully the above plot there is a chance of the model interpreting digit 6 , 7,8,0,1 as two.

Plot of accuracy of class – 3



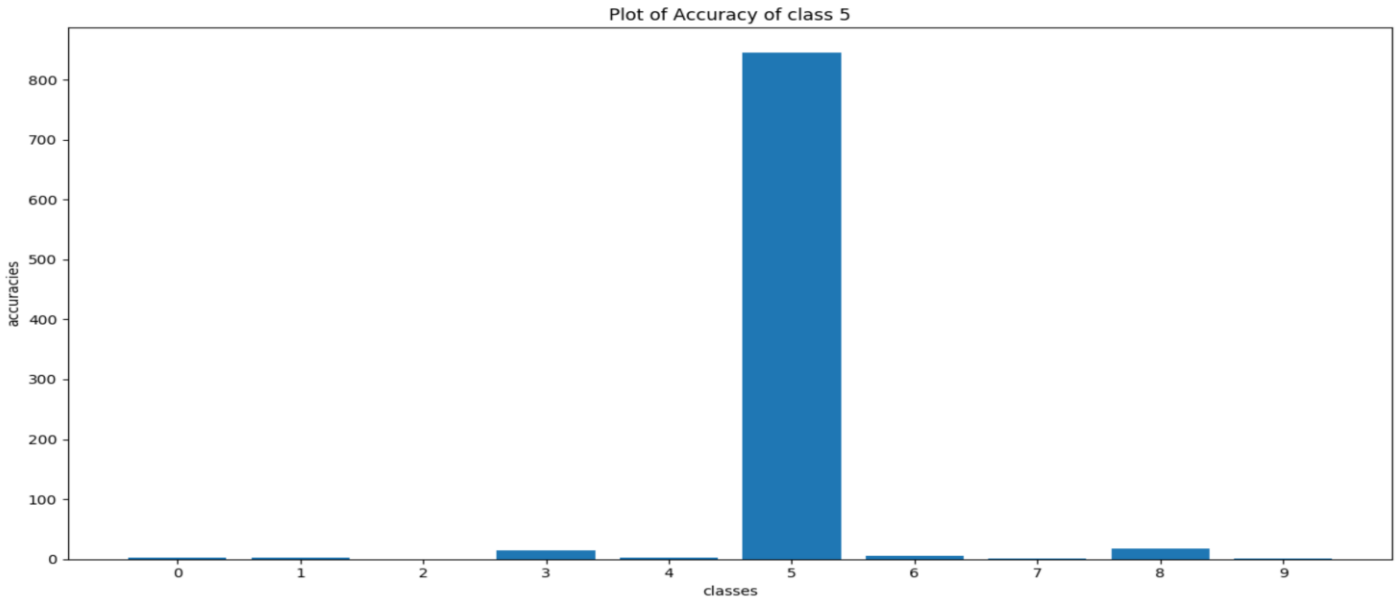
Class-3 has accuracy 97%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,2,5,7,8 as three.

Plot of accuracy of class – 4



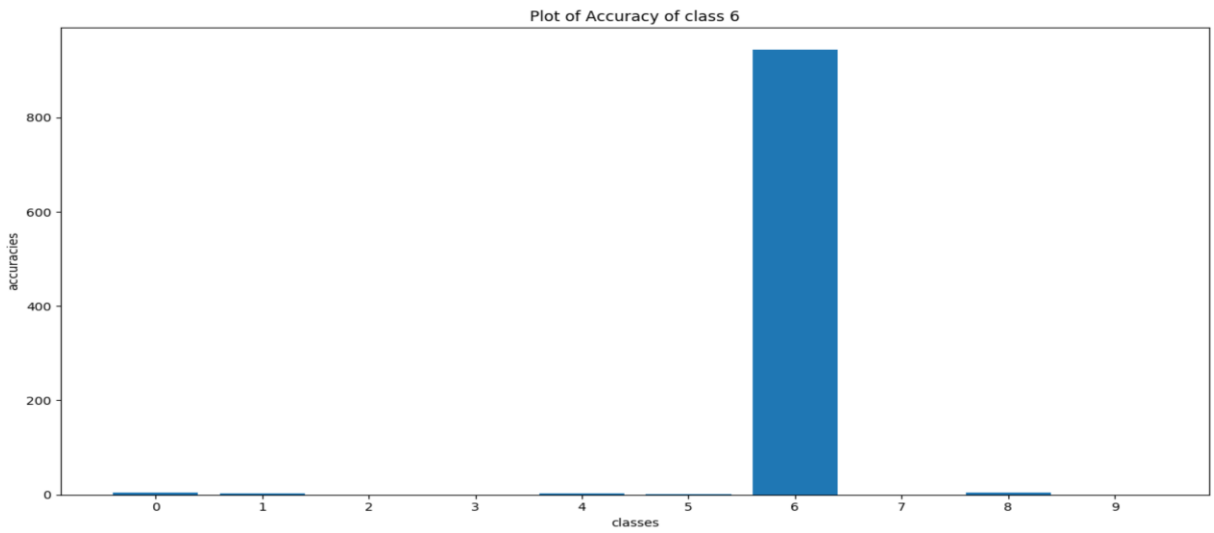
Class-4 has accuracy 98%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,6,7,8,9 as four.

Plot of accuracy of class – 5



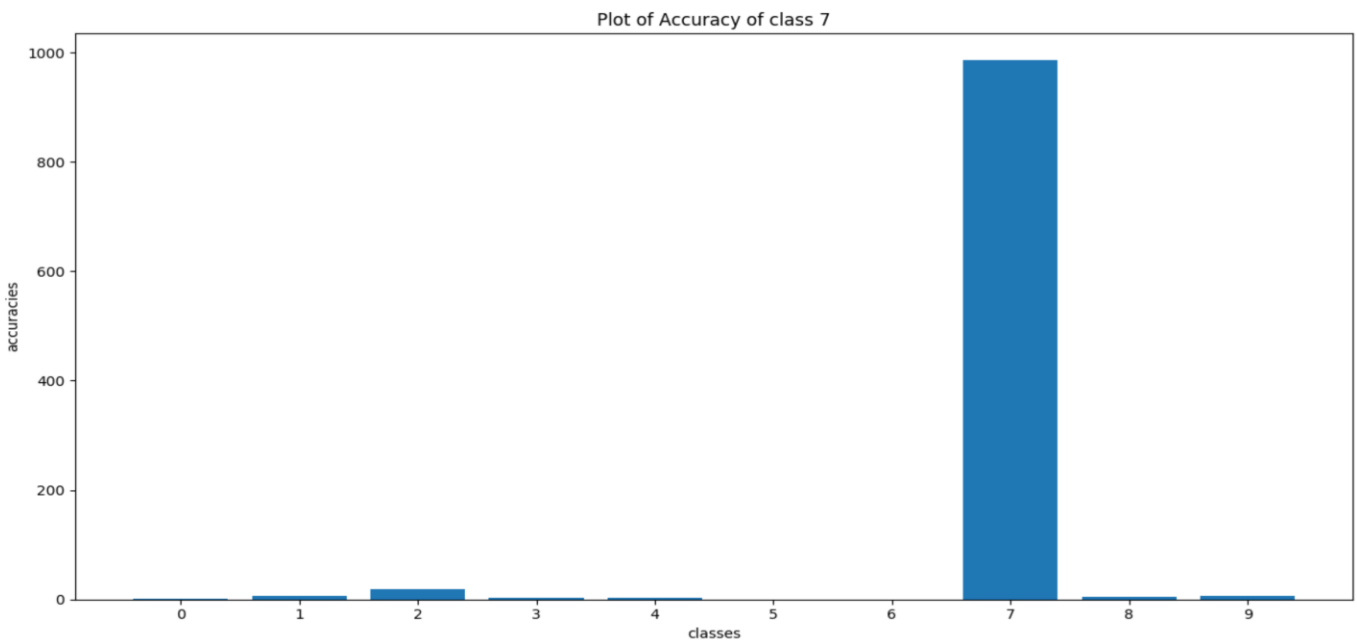
Class-5 has accuracy 94%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,3,4,6,8 as five.

Plot of accuracy of class – 6



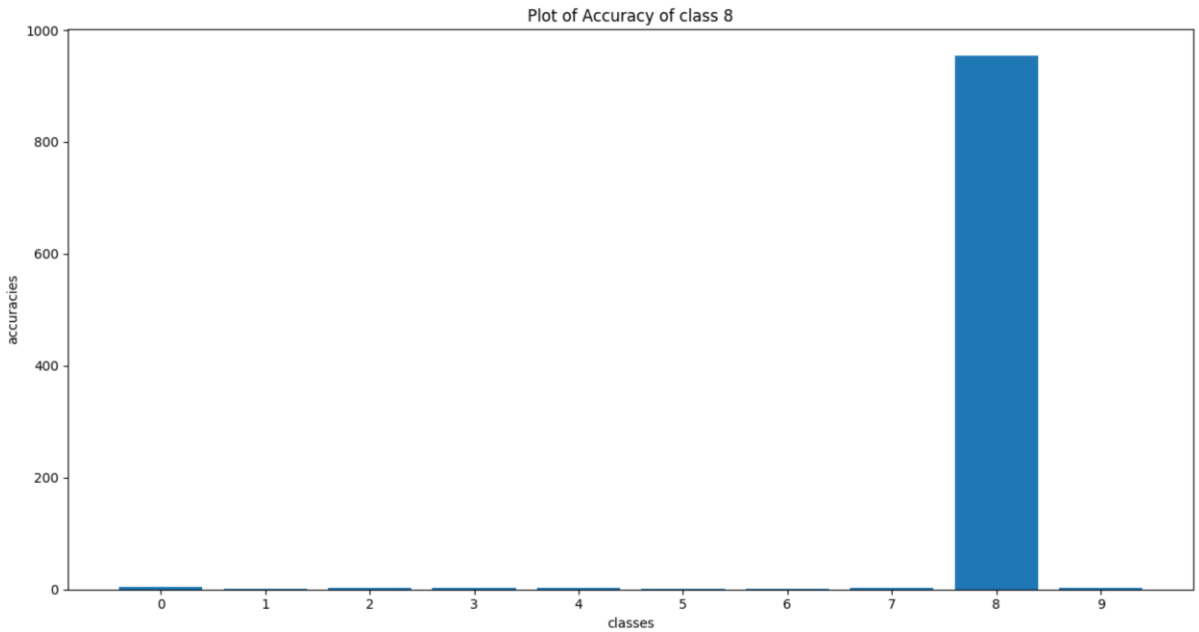
Class-6 has accuracy 98%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,4,8 as six.

Plot of accuracy of class – 7



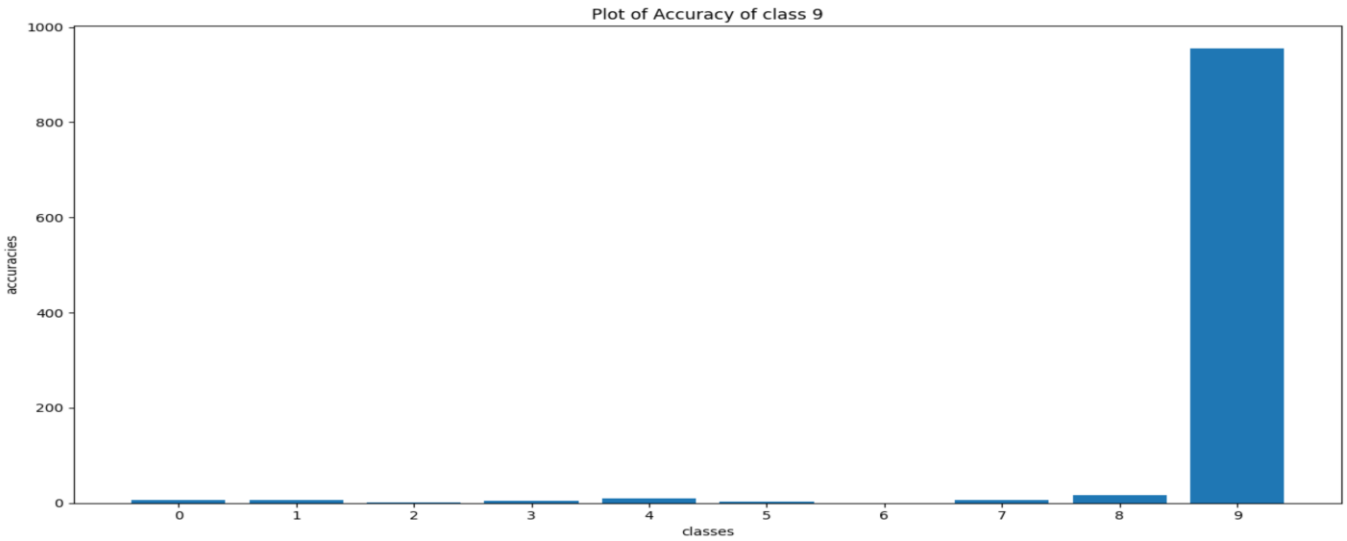
Class-7 has accuracy 95%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,2,3,4,8,9 as seven.

Plot of accuracy of class – 8



Class-8 has accuracy 97%. If we observe carefully the above plot there is a chance of the model interpreting the rest of the digits as eight.

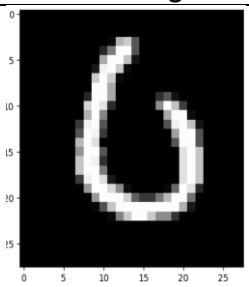
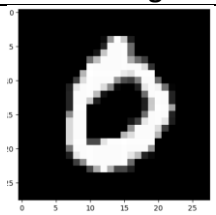
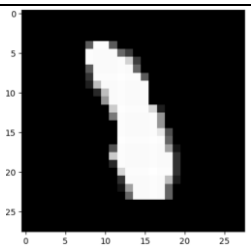
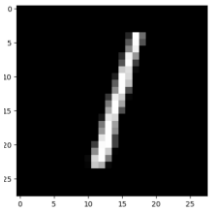
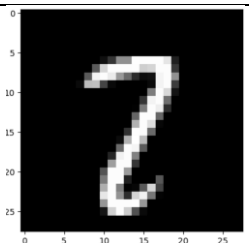
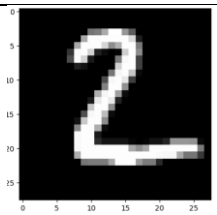
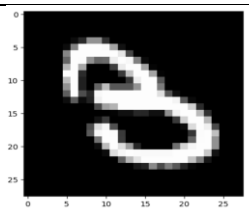
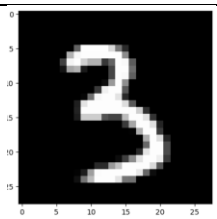
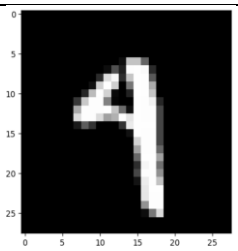
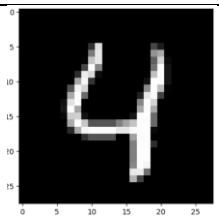
Plot of accuracy of class – 9

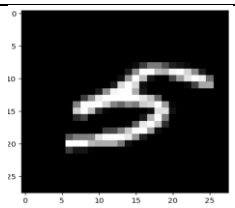
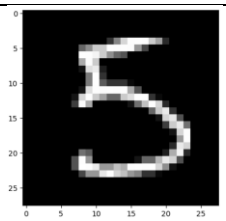
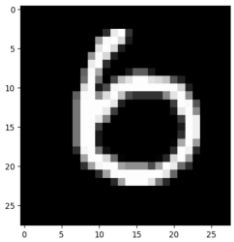
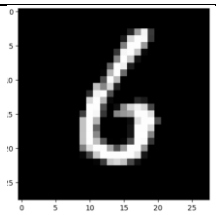
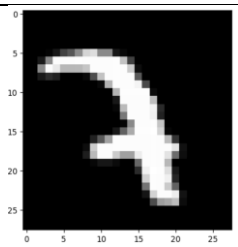
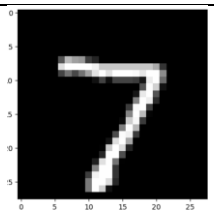
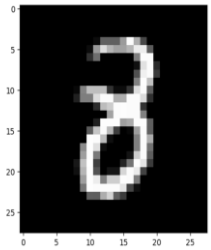
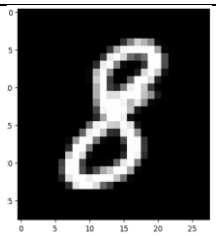
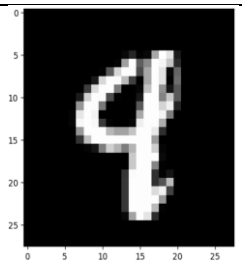
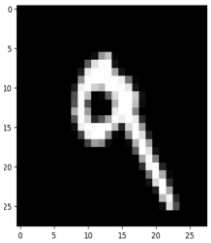


Class-9 has accuracy 94%. If we observe carefully the above plot there is a chance of the model interpreting digit 0,1,2,3,5,6,7,8 as nine.

From all the accuracies above we can observe that class 9 is most incorrectly predicted class among all classes.

3. Table of predictions of test samples:-

Class labels	Incorrect Image samples	Correct Image samples
0	 Actual label - 0 Predicted label- 6	 Actual label - 0 Predicted label- 0
1	 Actual label - 1 Predicted label- 3	 Actual label - 1 Predicted label- 1
2	 Actual label - 2 Predicted label- 7	 Actual label - 2 Predicted label- 2
3	 Actual label - 3 Predicted label- 8	 Actual label - 3 Predicted label- 3
4	 Actual label - 4 Predicted label- 9	 Actual label - 4 Predicted label- 4

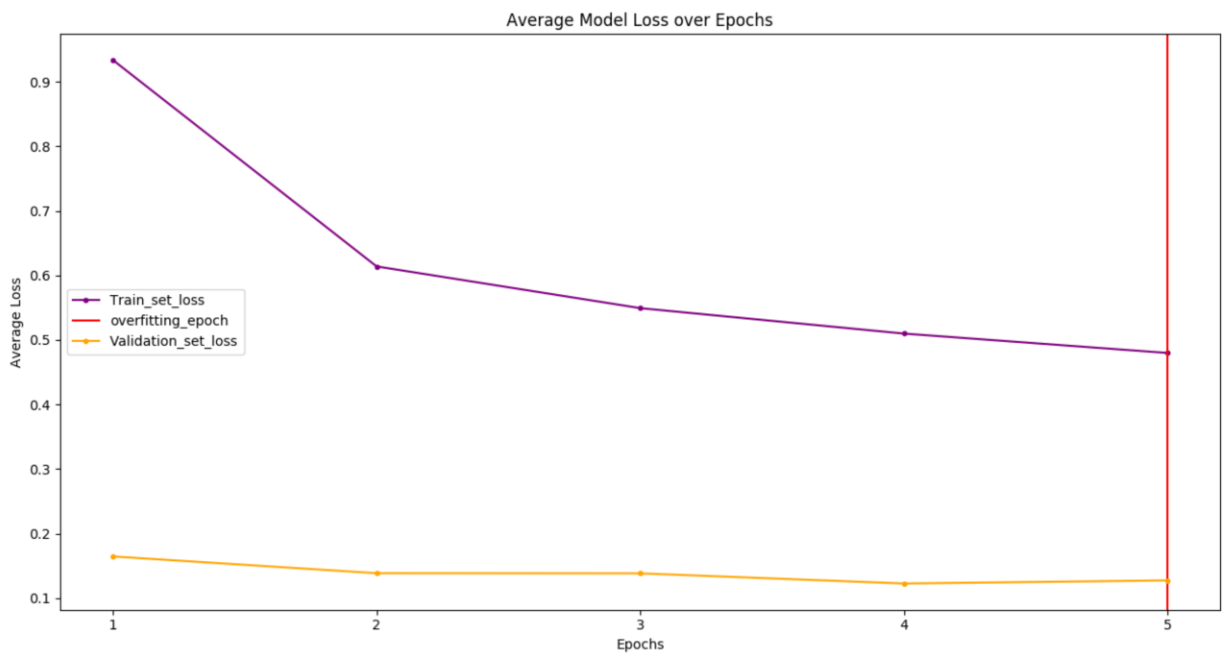
5	 Actual label - 5 Predicted label- 3	 Actual label - 5 Predicted label- 5
6	 Actual label - 6 Predicted label- 0	 Actual label - 6 Predicted label- 6
7	 Actual label - 7 Predicted label- 3	 Actual label - 7 Predicted label- 7
8	 Actual label - 8 Predicted label- 3	 Actual label - 8 Predicted label- 8
9	 Actual label - 9 Predicted label- 4	 Actual label - 9 Predicted label- 9

From the above table of correct and incorrect samples we can observe that the model is good in interpreting the digits which are visually readable with human eyes for eg. from above table we can

observe that in the in the incorrect samples column it is difficult for the model to interpret digit 2 as 2 , it appears to be like 7 and also digit 4 in the incorrect samples appears to be like 9 and the model has predicted it as 9. In the correct samples column we can see the model is very good in interpreting all the digits that can be visually interpreted easily. 0 in the incorrect samples column is interpreted as 6 which means it is difficult to predict images which are disjoint and if they resemble another digit, the model interprets as that incorrect digit.

Question 2:

1. Plot of cross-entropy vs. epoch



As we can see from the above diagram the overfitting occurs at epoch 5 . After epoch 4 the error on the validation set increases. The weights are reducing and training is converging. Compared to training on MNIST dataset where the training started to converge at epoch 9 using the nearly LeNet5 network adapting the model to fashion-MNIST dataset, we can observe that the training starts converging in less no of epochs which is 5. So using the pretrained we have better performance of training the model in less time.

Given below is the output of the program q2.py during training:-

```
Epoch 1, Train_loss: 0.9337569942077001
Epoch 1, Val_loss: 0.16450421226024628
Epoch 2, Train_loss: 0.613940746029218
Epoch 2, Val_loss: 0.13853246370951336
Epoch 3, Train_loss: 0.5492605069478352
Epoch 3, Val_loss: 0.13828227416674296
Epoch 4, Train_loss: 0.509861283659935
Epoch 4, Val_loss: 0.12260438525676727
Epoch 5, Train_loss: 0.4798030924797058
Epoch 5, Val_loss: 0.12735667260487873
Prev loss_validation  0.12260438525676727
Runnning loss_validation  0.12735667260487873
Finished Training
Time to converge 33.46678400039673
Plotting
```

As you can observe the validation loss at epoch 5 is 0.1273566 and at epoch 4 it is 0.1226 which is less than validation loss at epoch 5.

The time required to converge the weights was 33.466 seconds. Hence transfer learning helps in reducing the time required to train a model.

2.) Accuracies:-

The accuracy of Nearly-LeNet5 network before training tested on fashion-MNIST test set is as follows:-

```
Accuracy of the nearly LeNet5 network on the FashionMNIST test images: 5 %

Accuracy of     0 :  1 %  Incorrect samples  988  correct samples  12
Accuracy of     1 :  0 %  Incorrect samples  996  correct samples  4
Accuracy of     2 :  0 %  Incorrect samples  997  correct samples  3
Accuracy of     3 :  0 %  Incorrect samples  997  correct samples  3
Accuracy of     4 :  1 %  Incorrect samples  990  correct samples  10
Accuracy of     5 :  1 %  Incorrect samples  982  correct samples  18
Accuracy of     6 :  0 %  Incorrect samples  993  correct samples  7
Accuracy of     7 :  3 %  Incorrect samples  968  correct samples  32
Accuracy of     8 : 44 %  Incorrect samples  558  correct samples  442
Accuracy of     9 :  0 %  Incorrect samples  999  correct samples  1
```

fig(2.1)

The above results shows that without training on fashion-MNIST data, the model gives a very bad accuracy of 5% .This means that we must use a pretrained model to predict a different dataset only if features learned from the previous dataset are general features.

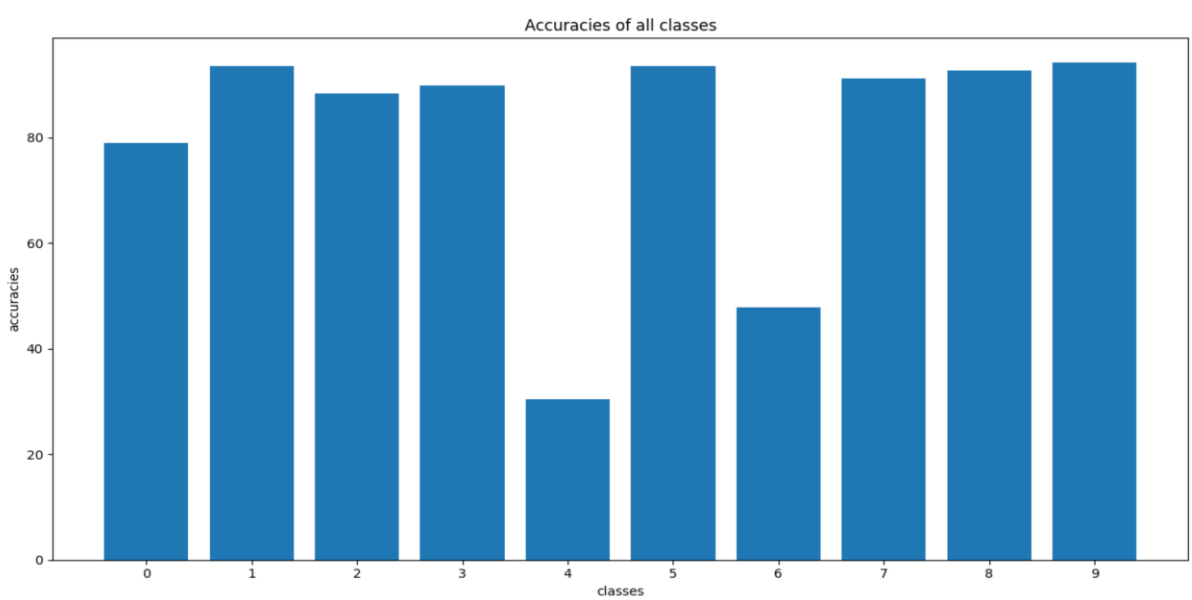
After transfer learning adapting the MNIST trained Nearly-LeNet5 weights to Fashion-MNIST. The following are the accuracies obtained:-

```
Accuracy of the network on the  test images: 80 %

Accuracy of     0 : 79 %  Incorrect samples  210  correct samples  790
Accuracy of     1 : 93 %  Incorrect samples  64  correct samples  936
Accuracy of     2 : 88 %  Incorrect samples  116  correct samples  884
Accuracy of     3 : 89 %  Incorrect samples  102  correct samples  898
Accuracy of     4 : 30 %  Incorrect samples  695  correct samples  305
Accuracy of     5 : 93 %  Incorrect samples  64  correct samples  936
Accuracy of     6 : 47 %  Incorrect samples  522  correct samples  478
Accuracy of     7 : 91 %  Incorrect samples  88  correct samples  912
Accuracy of     8 : 92 %  Incorrect samples  72  correct samples  928
Accuracy of     9 : 94 %  Incorrect samples  58  correct samples  942
```

fig(2.2)

As we observe fig 2.1 and fig 2.2 , the no of correct samples which are predicted for each class has increased significantly and the overall accuracy before training on Fashion-MNIST dataset is 5% and after transfer learning is 80% but after transfer learning we can say that we have much more correct samples being classified and the model was trained in less time 33.466 seconds compared to the MNIST trained Nearly-LeNet5 network which was implemented from scratch was trained in time 58.116 seconds. So transfer learning helps in reducing the time to converge the weights. But compared to nearly LeNet5 network which had 97% accuracy and this model after training on Fashion-MNIST dataset gives accuracy of 80%. The accuracy after training increased from 5% to 80% but still much accuracy is expected in image classification problems.

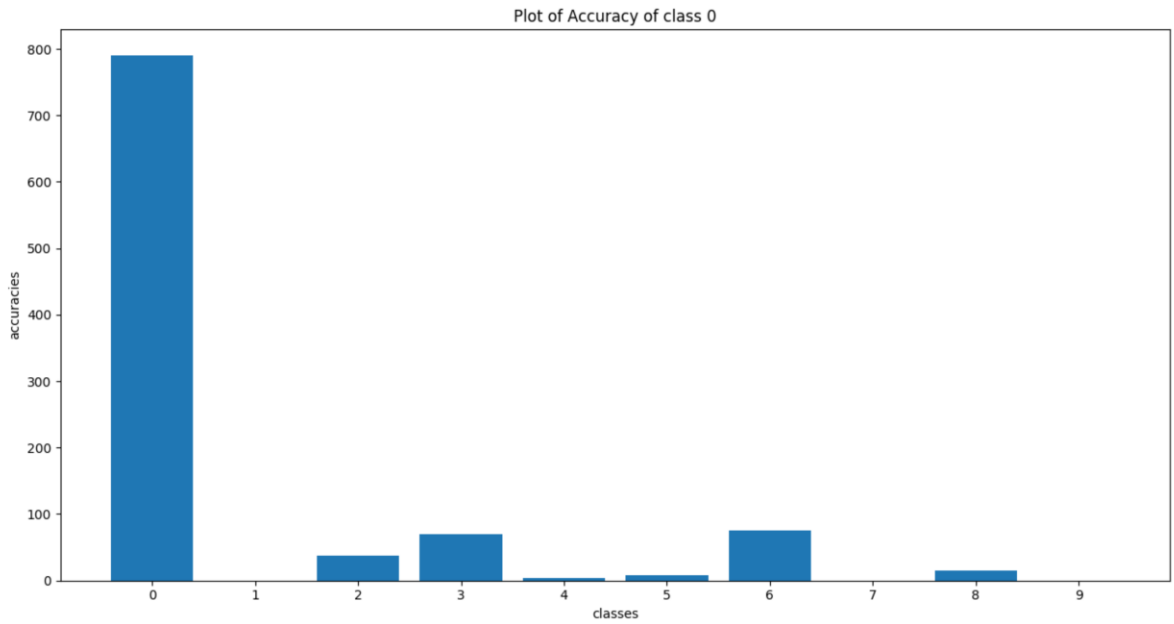
Plot of Accuracies of all class:-



From the above accuracies we can observe that accuracy of class 9 is maximum and class 4 is minimum which means the model is very poor in predicting the label 4 and very good in predicting the label 9.

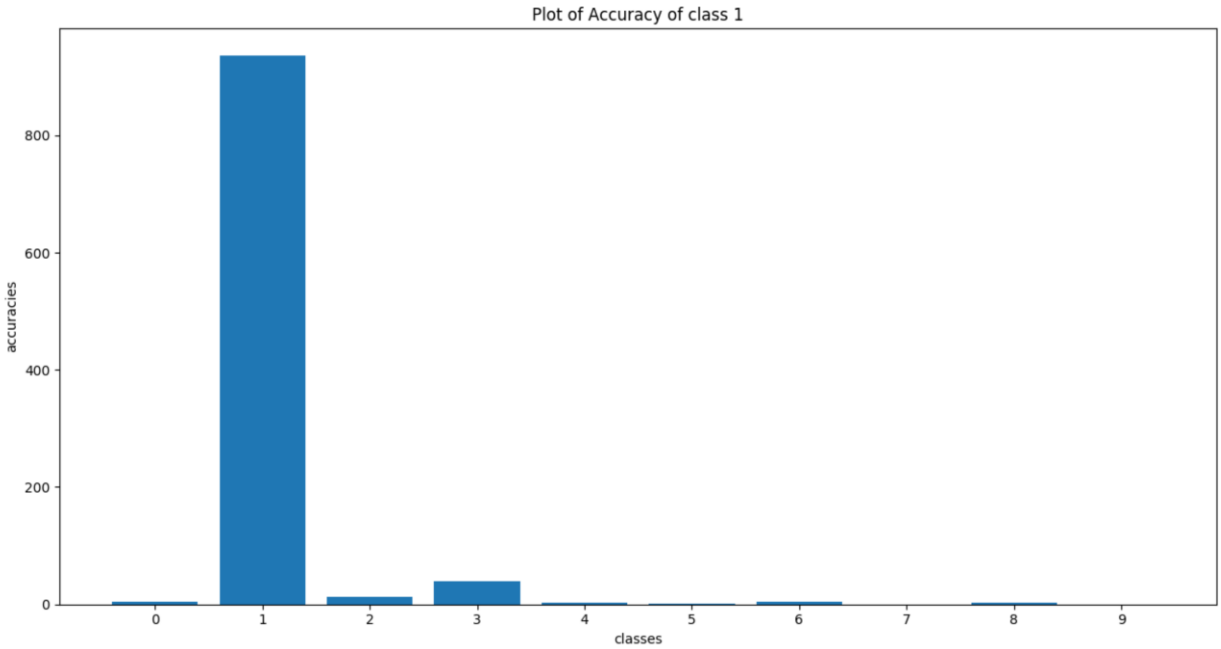
Plot of Accuracies of individual class:-

Plot of accuracy of class – 0



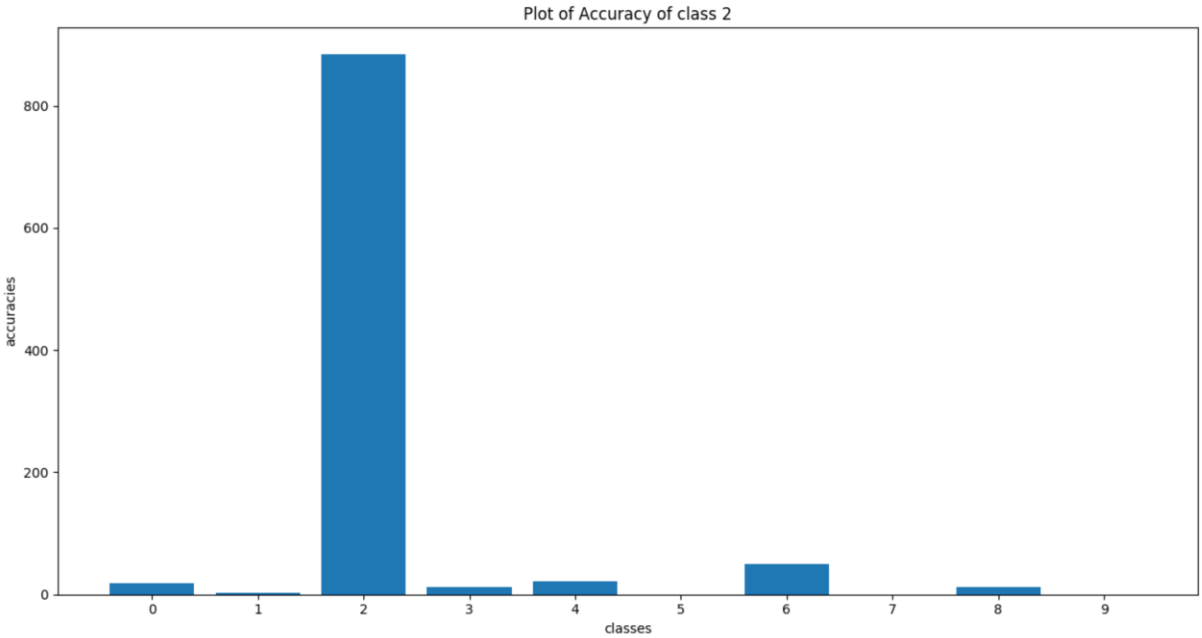
Class-0 : If we observe carefully the above plot there is more chance of the model interpreting label 6 as zero.

Plot of accuracy of class – 1



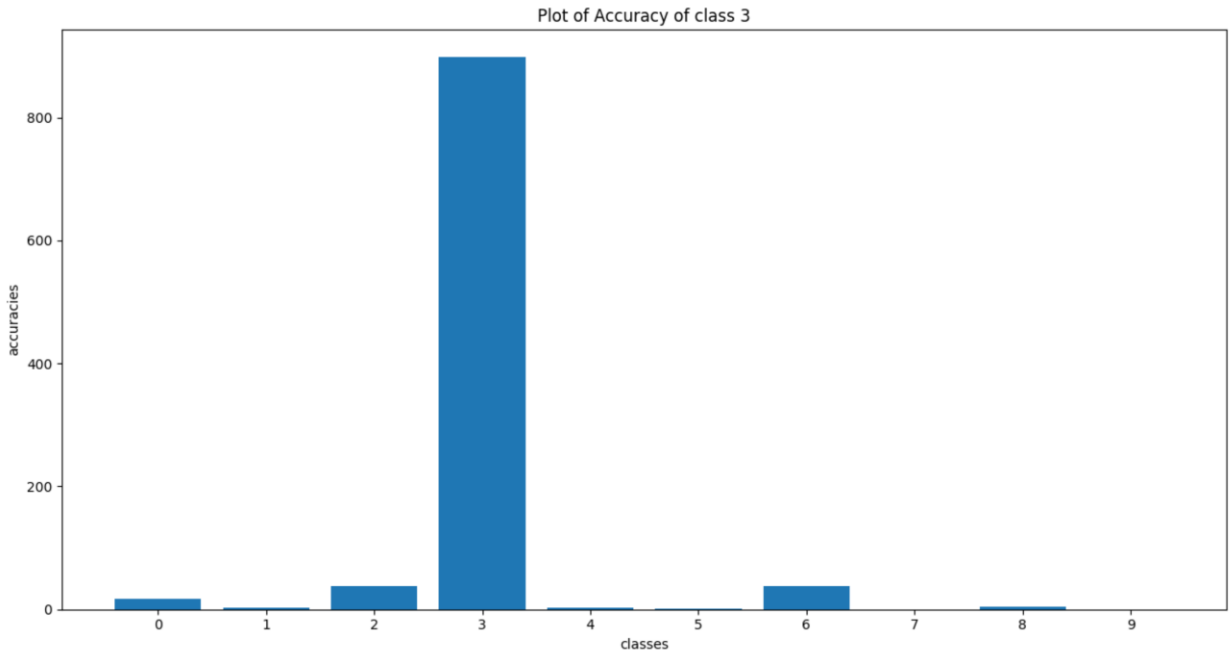
Class-1 : If we observe carefully the above plot there is more chance of the model interpreting labels 3 as one.

Plot of accuracy of class – 2



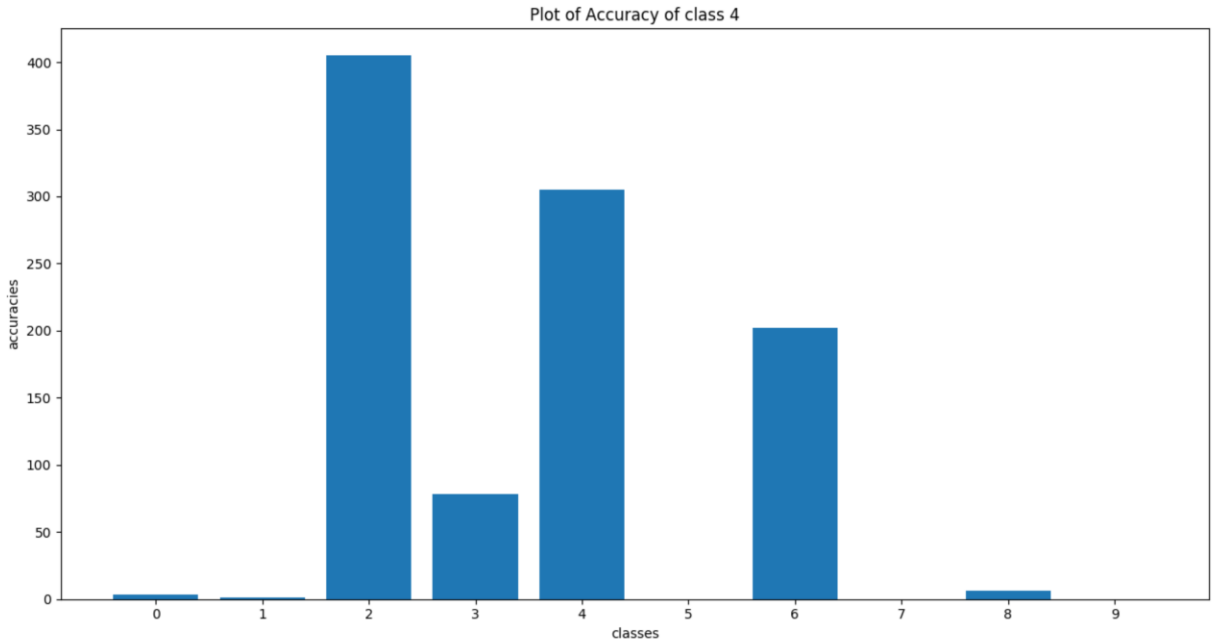
Class-2 : If we observe carefully the above plot there is more chance of the model interpreting labels 6 as 2.

Plot of accuracy of class – 3



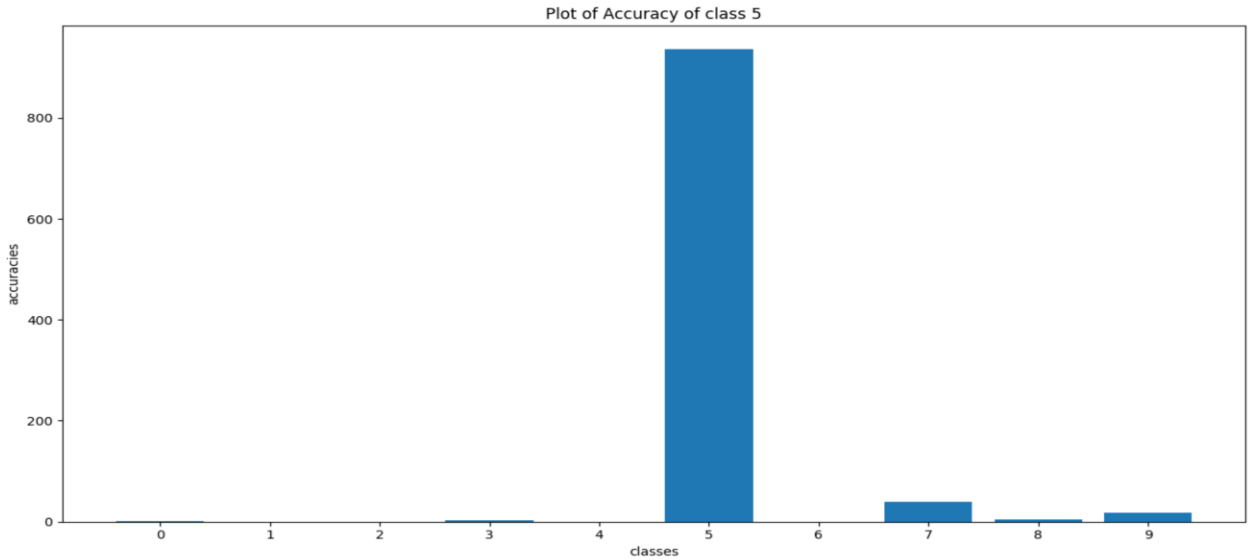
Class-3 : If we observe carefully the above plot there is more chance of the model interpreting labels 6 and 2 as three.

Plot of accuracy of class – 4



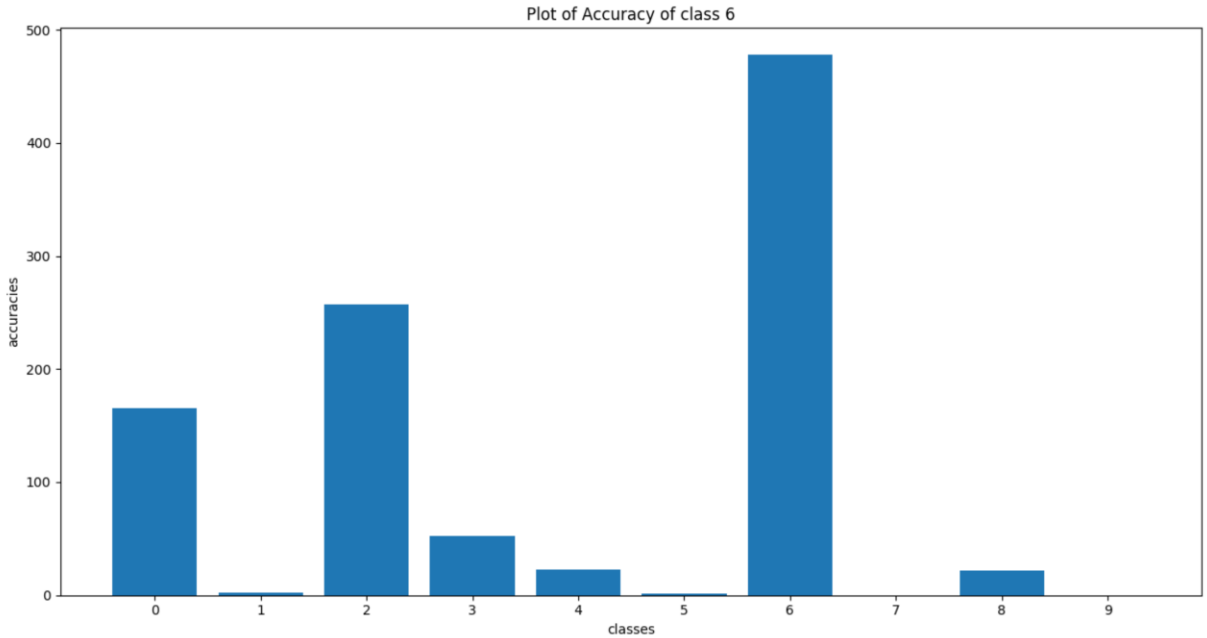
Class-4 : If we observe carefully the above plot there is more chance of the model interpreting label 4 as 2 and this class has the lowest accuracies among all classes because the model is poor predicting the label 4 and in many cases it predicts it as 2.

Plot of accuracy of class – 5



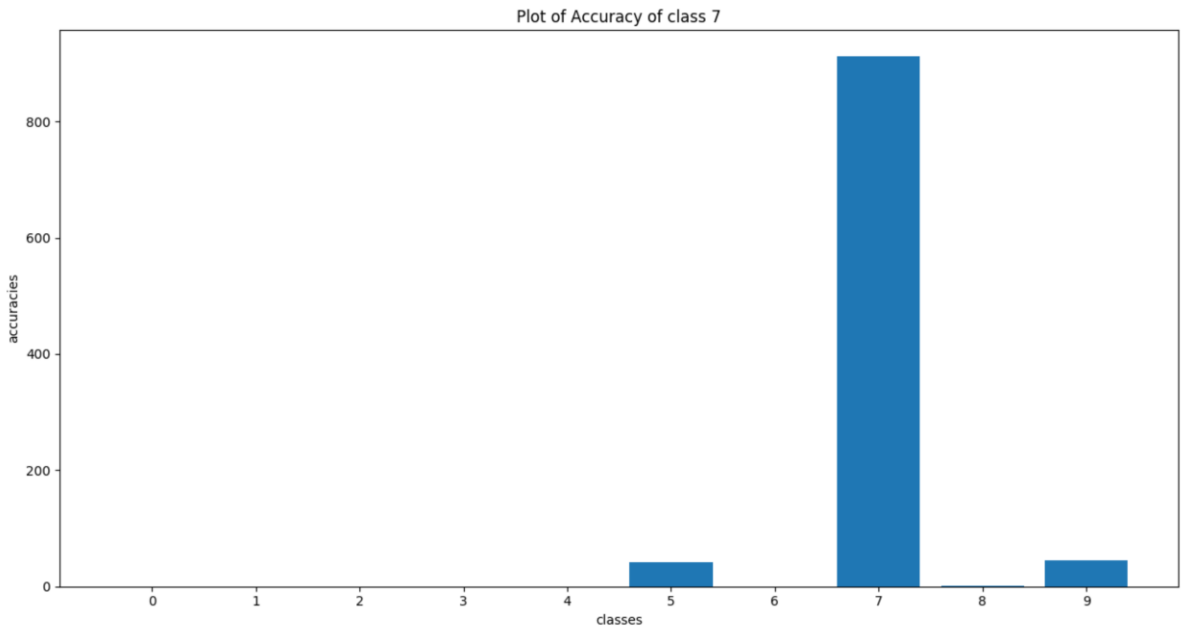
Class-5 : If we observe carefully the above plot there is more chance of the model interpreting label 7 as five.

Plot of accuracy of class – 6



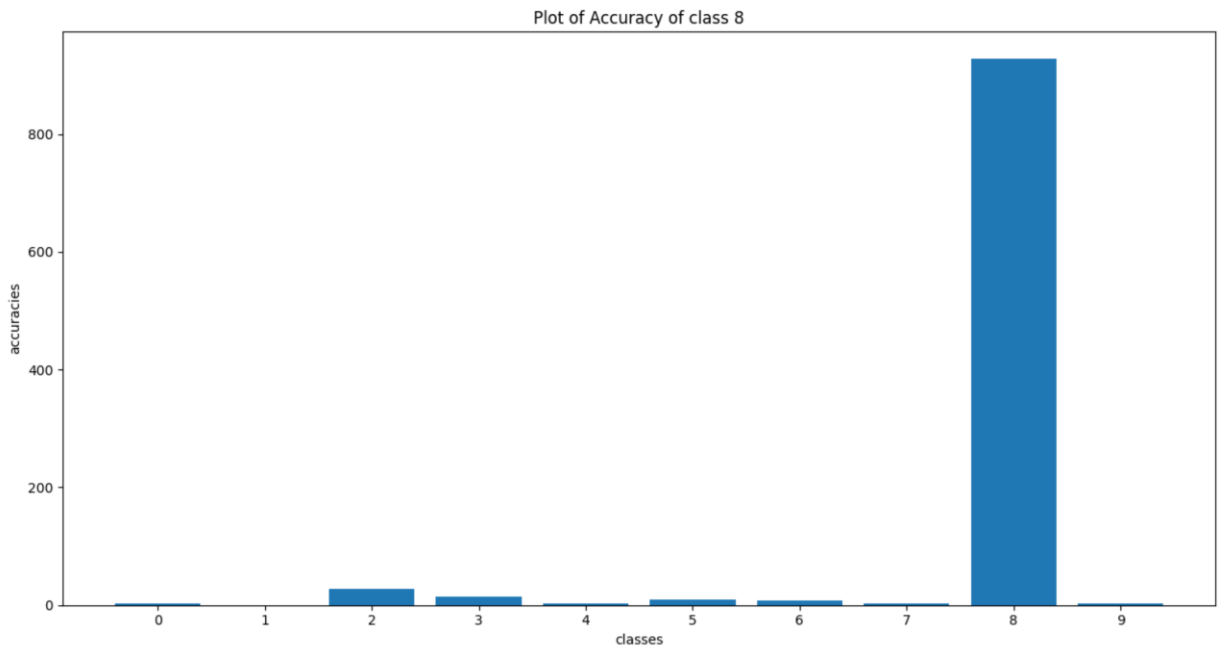
Class-6 : If we observe carefully the above plot there is more chance of the model interpreting 2 as six. This class has accuracy of 47% and so the model is poor in predicting this label and misinterprets as labels 0,2,3,4,8.

Plot of accuracy of class – 7



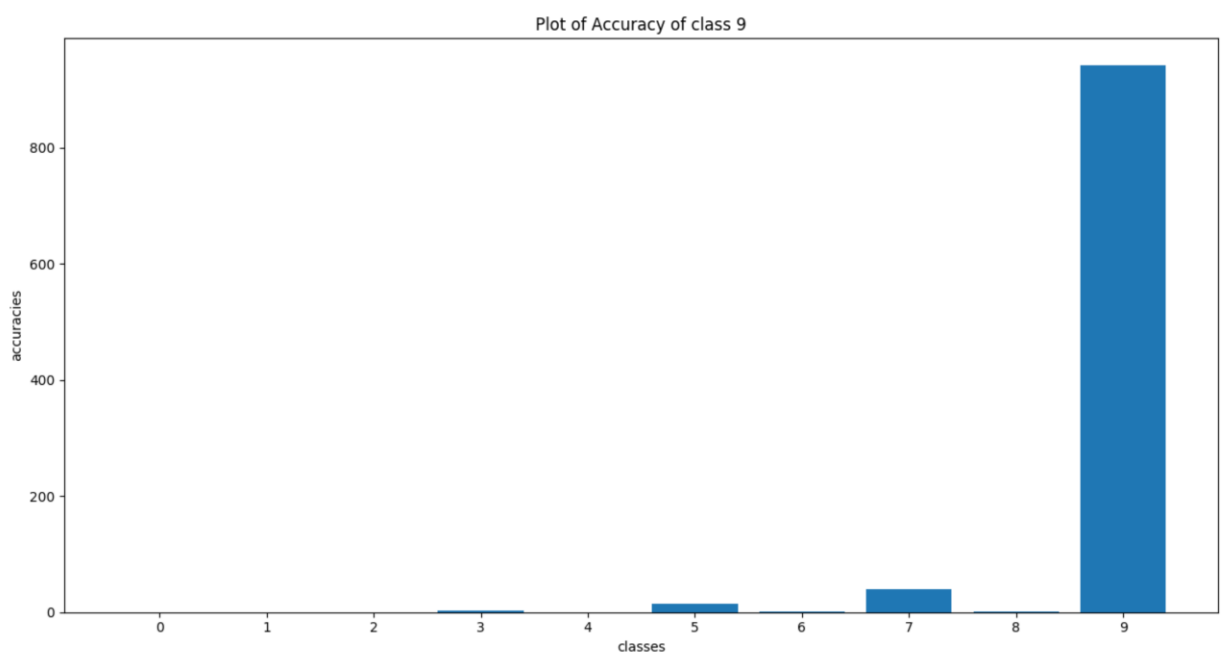
Class-7 : If we observe carefully the above plot there is more chance of the model interpreting label 9 as seven.

Plot of accuracy of class – 8



Class-8 : If we observe carefully the above plot there is more chance of the model interpreting label 2 as eight.

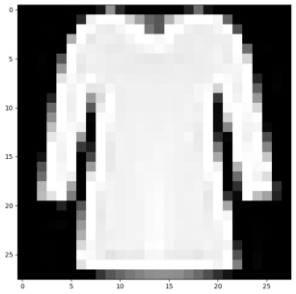
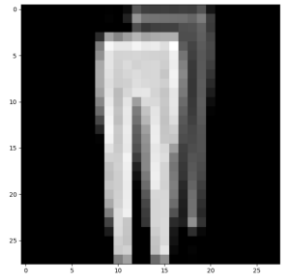
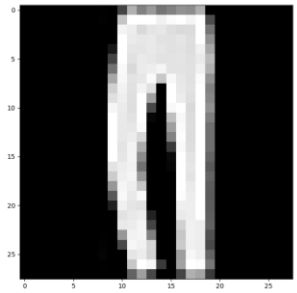
Plot of accuracy of class – 9

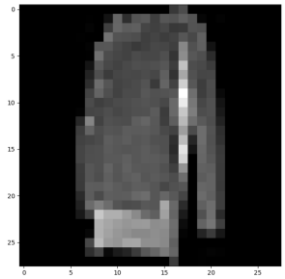
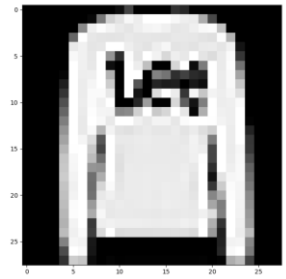
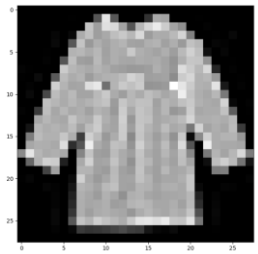
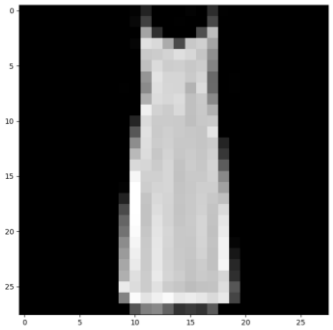
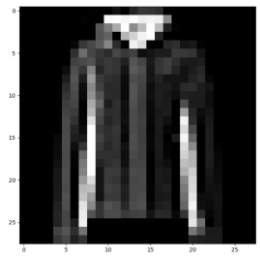
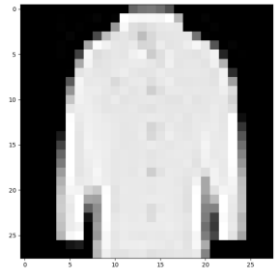
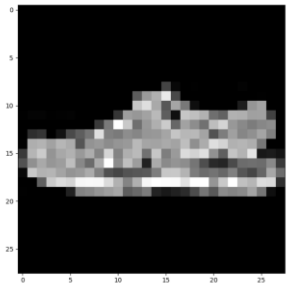
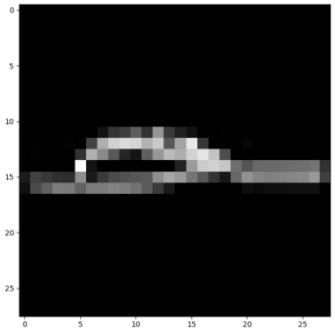


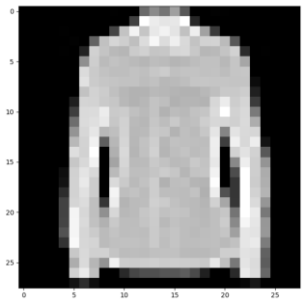
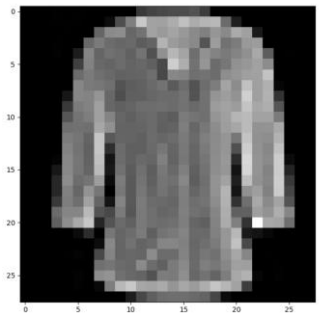
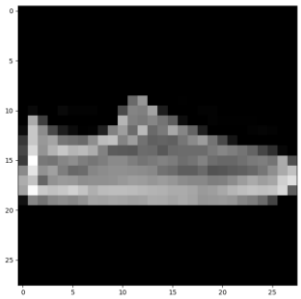
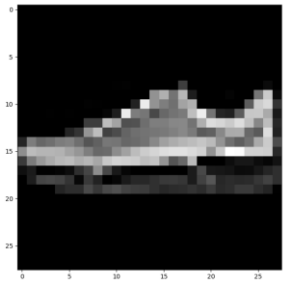
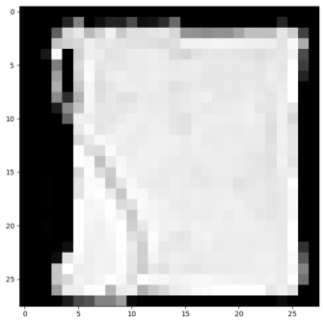
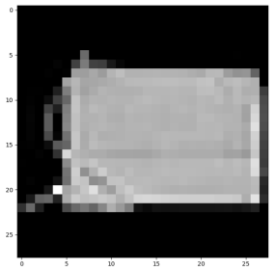
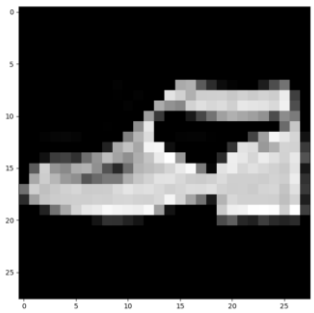
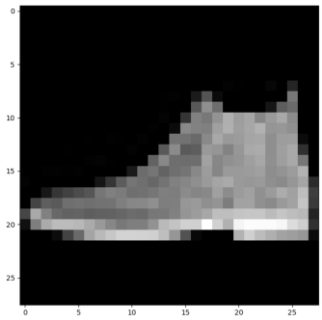
Class-9 : If we observe carefully the above plot there is more chance of the model interpreting label 7 as nine.

From all the accuracies above we can observe that the model is finding it difficult to predict the labels 4 and 6 which are coat and shirt respectively and misinterprets most of the time as label 2 which is pullover. So the model is not very effective in recognizing the difference between coat , shirt and pullover.

a.) Table of predictions of test samples:-

Class labels	**Incorrect Image samples**	**Correct Image samples**
0	 Actual label - 0 (T-shirt/top) Predicted label- 2	 Actual label - 0 Predicted label- 0
1	 Actual label - 1 (Trouser) Predicted label- 3	 Actual label - 1 Predicted label- 1

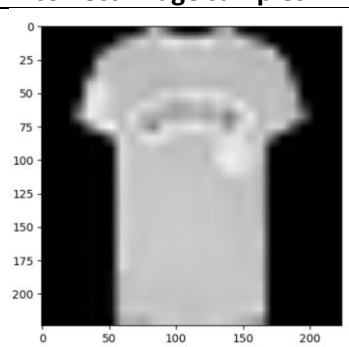
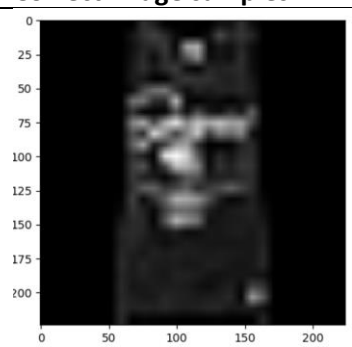
2	 Actual label - 2 (Pullover) Predicted label- 3	 Actual label - 2 Predicted label- 2
3	 Actual label - 3 (Dress) Predicted label- 6	 Actual label - 3 Predicted label- 3
4	 Actual label - 4 (Coat) Predicted label- 6	 Actual label - 4 Predicted label- 4
5	 Actual label - 5 (Sandal) Predicted label- 7	 Actual label - 5 Predicted label- 5

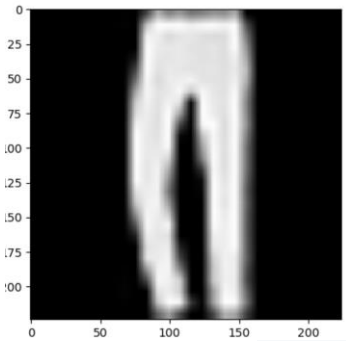
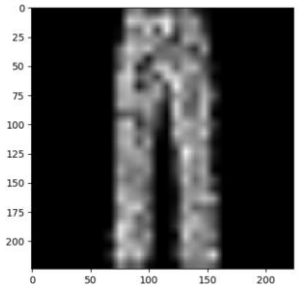
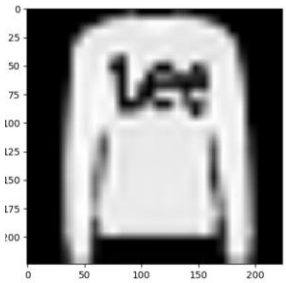
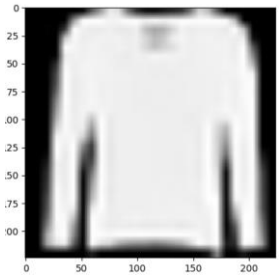
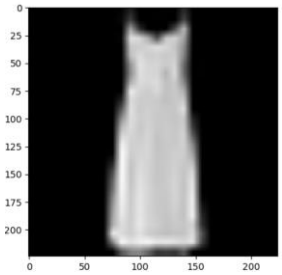
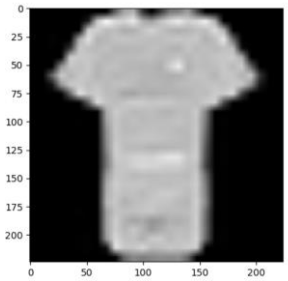
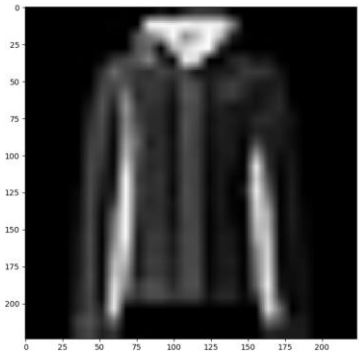
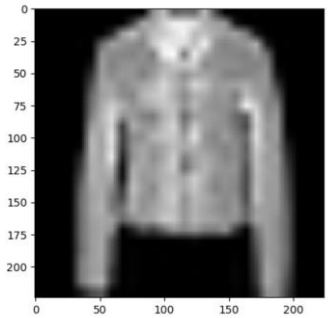
6	 Actual label - 6 (Shirt) Predicted label- 2	 Actual label - 6 Predicted label- 6
7	 Actual label - 7 (Sneaker) Predicted label- 8	 Actual label - 7 Predicted label- 7
8	 Actual label - 8 (Bag) Predicted label- 2	 Actual label - 8 Predicted label- 8
9	 Actual label - 9 (Ankle boot) Predicted label- 5	 Actual label - 9 Predicted label- 9

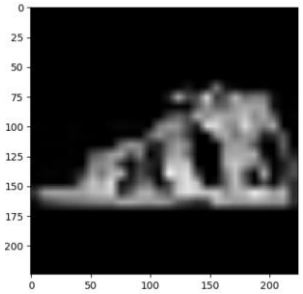
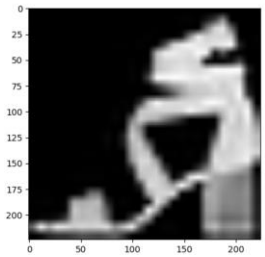
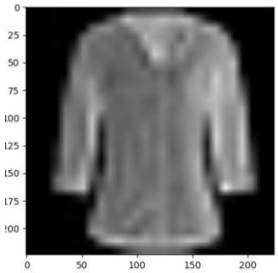
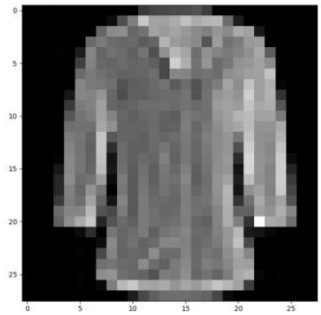
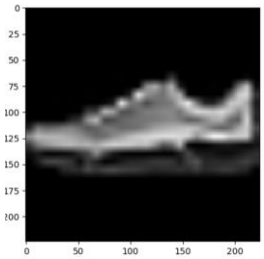
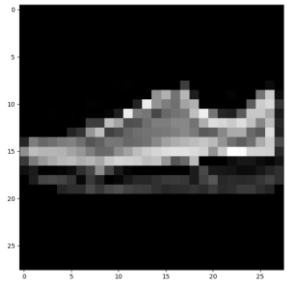
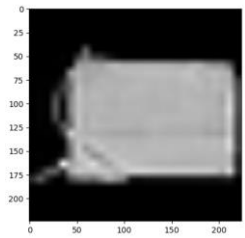
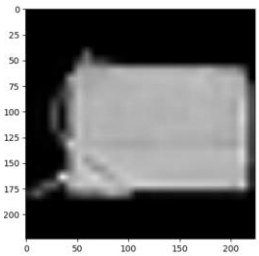
From the above table of correct and incorrect samples we can understand that the model is good in predicting the labels of images which are clear and poor in predicting the labels of images which are visually not clear or may be blurred or may be similar to other images like label 9 and 5 which are ankle boot and sandal respectively.

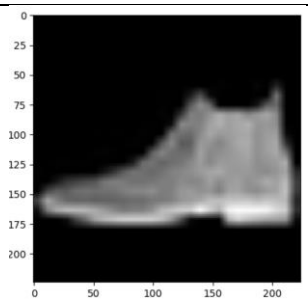
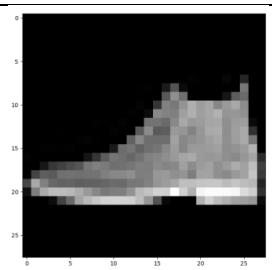
Question 3)

a.) Table of predictions of test samples:-

Class labels	Incorrect Image samples	Correct Image samples
0		

	Actual label - 0 (T-shirt/top) Predicted label- 3	Actual label - 0 Predicted label- 0
1	 Actual label - 1 (Trouser) Predicted label- 5	 Actual label - 1 Predicted label- 1
2	 Actual label - 2 (Pullover) Predicted label- 5	 Actual label - 2 Predicted label- 2
3	 Actual label - 3 (Dress) Predicted label- 5	 Actual label - 3 Predicted label- 3
4	 Actual label - 4 (Coat) Predicted label- 1	 Actual label - 4 Predicted label- 4

5	 Actual label - 5 (Sandal) Predicted label- 6	 Actual label - 5 Predicted label- 5
6	 Actual label - 6 (Shirt) Predicted label- 8	 Actual label - 6 Predicted label- 6
7	 Actual label - 7 (Sneaker) Predicted label- 5	 Actual label - 7 Predicted label- 7
8	 Actual label - 8 (Bag) Predicted label- 9	 Actual label - 8 Predicted label- 8

9	 Actual label - 9 (Ankle boot) Predicted label- 5	 Actual label - 9 Predicted label- 9

From the above table of correct and incorrect samples we can observe that the model can predict the sample if the image is clear and those images which can be easily interpreted visually and performs classification much better than previous LeNet5 models

2.) Accuracies :

Accuracy of the network before training

```
Accuracy of the network on the  test images: 10 %

Accuracy of     0 :  0 %  Incorrect samples  999  correct samples  1
Accuracy of     1 : 10 %  Incorrect samples  898  correct samples  102
Accuracy of     2 :  0 %  Incorrect samples  1000  correct samples  0
Accuracy of     3 :  3 %  Incorrect samples  969  correct samples  31
Accuracy of     4 :  5 %  Incorrect samples  943  correct samples  57
Accuracy of     5 :  4 %  Incorrect samples  952  correct samples  48
Accuracy of     6 :  0 %  Incorrect samples  996  correct samples  4
Accuracy of     7 : 67 %  Incorrect samples  324  correct samples  676
Accuracy of     8 : 15 %  Incorrect samples  841  correct samples  159
Accuracy of     9 :  0 %  Incorrect samples  1000  correct samples  0
```

Since the accuracy is very low as 10% the model was trained for 10 epochs and the results obtained are as follows:-

Accuracy of the network after training:-

```
Accuracy of the network on the  test images: 85 %

Accuracy of     0 : 81 %  Incorrect samples  182  correct samples  818
Accuracy of     1 : 95 %  Incorrect samples  44  correct samples  956
Accuracy of     2 : 83 %  Incorrect samples  167  correct samples  833
Accuracy of     3 : 85 %  Incorrect samples  150  correct samples  850
Accuracy of     4 : 77 %  Incorrect samples  225  correct samples  775
Accuracy of     5 : 92 %  Incorrect samples  74  correct samples  926
Accuracy of     6 : 51 %  Incorrect samples  484  correct samples  516
Accuracy of     7 : 92 %  Incorrect samples  72  correct samples  928
Accuracy of     8 : 96 %  Incorrect samples  33  correct samples  967
Accuracy of     9 : 93 %  Incorrect samples  63  correct samples  937
```

The model performs better in predicting the labels with accuracy of only 85% which is better than 80% accuracy of previous model. The maximum accuracy among all classes is accuracy of class 8.

The pretrained model which was used is resnet18. The finally fully connected layer was reset to predict the 10 classes. The model accepts only 3 channel RGB so the Fashion-MNIST images which are gray scale were transformed to convert them to 3-channel RGB by repeating them to each band. The final layer was freezed and linear layer was added after this to predict the 10 classes and this layer is not freezed and the model accepts only a 224x224 images and according the tensor was resized and the code is mentioned in data_transform() function in file q3.py . The model was run in 10 epochs but may perform better if we increase the no of epochs.

```
Training started
Epoch 1, Train_loss: 0.9350937334696452
Epoch 2, Train_loss: 0.6225956614812215
Epoch 3, Train_loss: 0.57153009446462
Epoch 4, Train_loss: 0.5440888873934746
Epoch 5, Train_loss: 0.5259170797864596
Epoch 6, Train_loss: 0.5126328510840734
Epoch 7, Train_loss: 0.5023286398847898
Epoch 8, Train_loss: 0.4940095581213633
Epoch 9, Train_loss: 0.48709528569380445
Epoch 10, Train_loss: 0.4812202115058899
Finished Training
Time to converge 786.5500919818878
```

But the time required to train this pretrained model to train and improve the accuracy was 786.555 seconds which is much larger than the previous models used.