REPORT

Steps performed in the code for getting the output are as bellow:-

i. Transformed the input image of 2d array of into 1d .

ii. Implemented the PCA to convert the input into 12 dimensions

iii. Initialized the weights and bias with the random numbers.

iv. Perform the forward pass

v. Calculated the gradients of cost function with respect to weight and bias by the help of chain rule as discussed in class.

vi. Now we performed the back propagation.

vii. We updated the weights and bias.

viii. Calculated the error by the help of entropy loss function for the test data

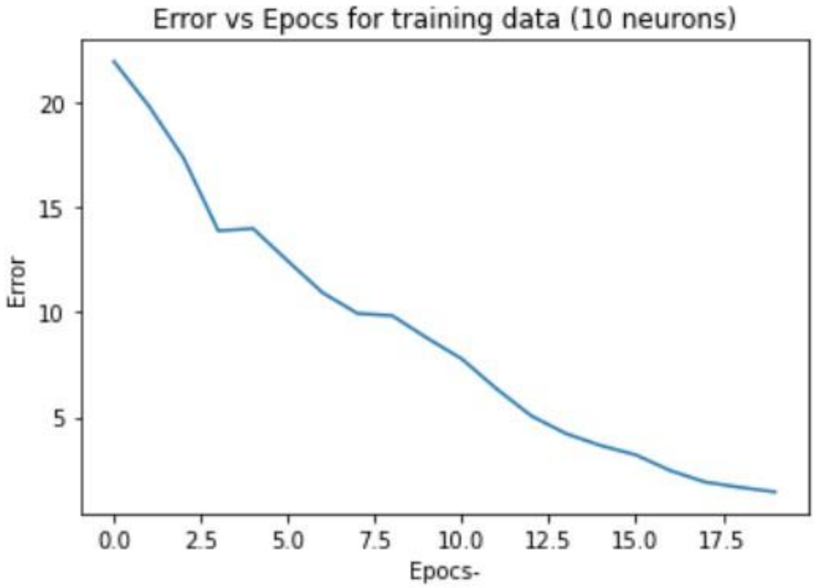
ix. Running the above code for the 20 number of epocs.

x. Now by the help of train data implements the PCA in the test data.

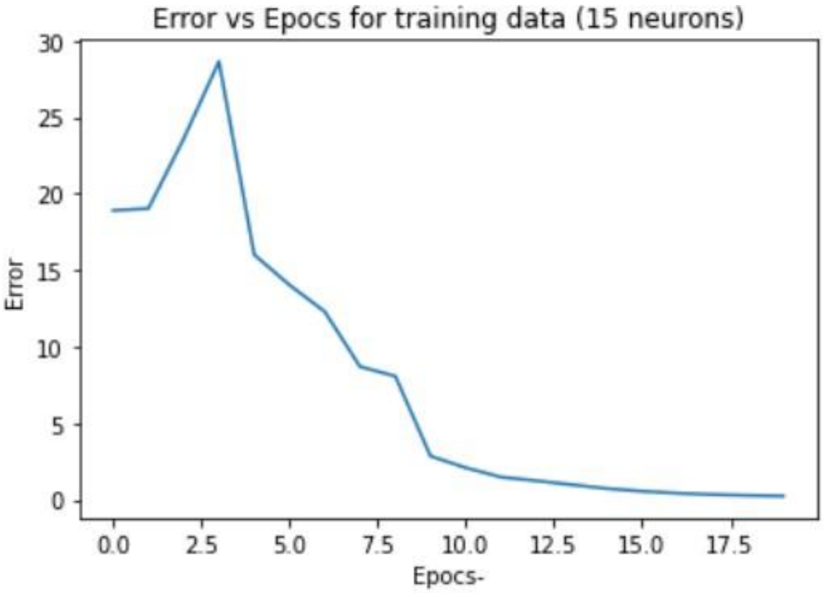
xi. Calculated the output by the help of parameters that we learnt from train data

xii. Now calculated the classification accuracy with the help of true label and the calculated output.

RESULTS:-



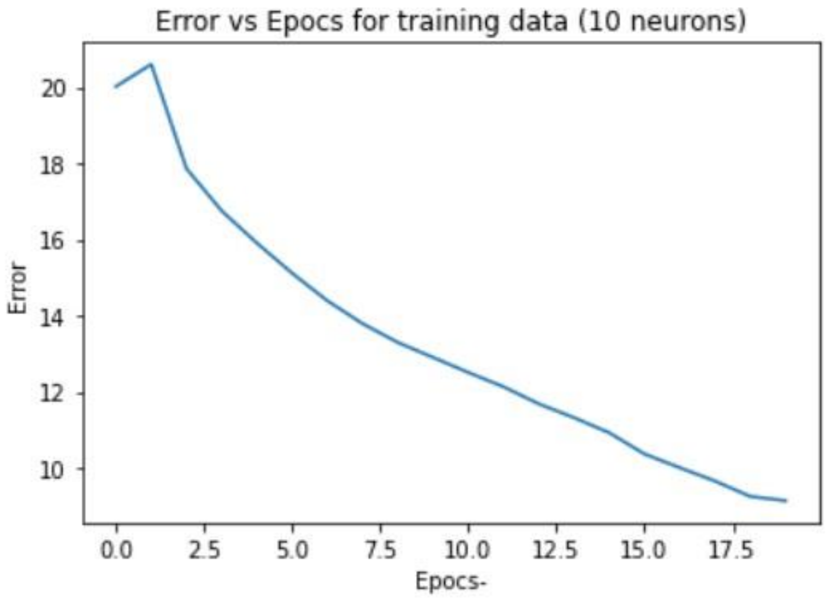
```
Overall Training Error in case of 10 neurons is 176.71852013313264
The classification accuracy obtained in test data is 100 percent
```

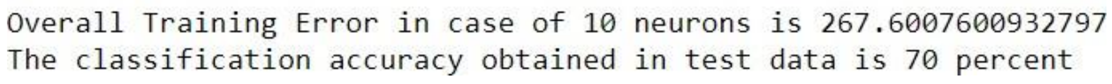


```
Overall Training Error in case of 15 neurons is 160.33729608222203
The classification accuracy obtained in test data is 100 percent
```

OBSERVATION AND CONCLUSION:- Test classification accuracy for both the cases was around 90% if we take the average of taking the 5 observations as we initializing the weights with the random number we didn’t get 100% accuracy everytime mostly it ranges between (80% to 100%), so I have shown the best case in my result . When we increased the number of neurons I got less error and better accuracy but we have to change the learning rate for this. In contrast for the same learning rate accuracy obtained in 15 neurons were less as compared to the 10 neurons case. The reason for this may be the model complexity i.e. increasing the neurons increases the model complexity and hence increases error and decreases the accuracy and another reason can be the overfitting in the train data in case of 15 neurons.

Yes momentum increased the learning process in the code, when we was not using the momentum factor than it was taking much time along with the less performance as compared with the case when we have taken the momentum factor=0.9. The result for without taking the momentum have been shown below.





```
Overall Training Error in case of 10 neurons is 267.6007600932797
The classification accuracy obtained in test data is 70 percent
```

The accuracy in this case was less as compared to the other one and also this code was taken much time as compared to the other case since there was no momentum factor.