Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Experiment 1- Vary number of hidden units

In this experiment, we have to check the accuracy of the neural network by varying the number of hidden layers and keep the momentum and learning rate as 0.9 and 0.1.

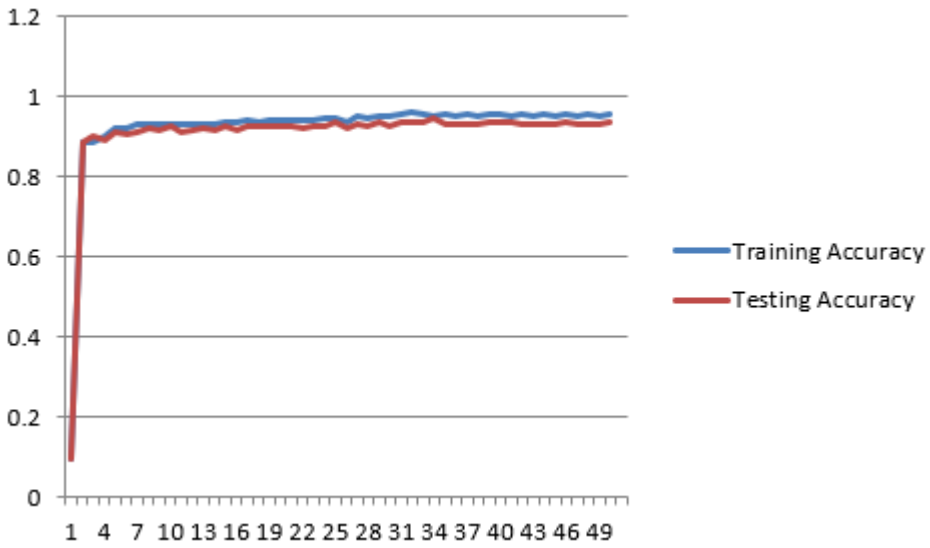
We will be testing this for 3 hidden layers 20, 50 and 100. For all these, we have to calculate accuracies for each epoch and have to plot graph of test and training accuracies, and we also need to create confusion matrix at the end of the training on the test data.

We will be using MNIST data set which will be having 784 inputs and 10 outputs in the size 28 * 28. So there will be 60,000 training data and 10,000 test data.

Graph and confusion matrix for n=20:

The final accuracy obtained was 93-94 %

Graph:



Confusion Matrix:

```
[[ 957    0    1    4    2    4    9    1    1    1]
 [   0 1110    3    6    1    3    4    2    5    1]
 [   8    3  950   28   12    4    7    8   10    2]
 [   1    1   16  962    1   11    0    6    7    5]
 [   1    1    5    1  937    0   11    0    2   24]
 [   6    2    4   30    9  805   18    2   11    5]
 [  11    3    4    3   12   14  903    0    8    0]
 [   3    7   18   14    6    1    0  952    2   25]
 [   5    5    7   41   16   14    7   11  864    4]
 [  12    4    0   19   51    6    1   16    5  895]]
```

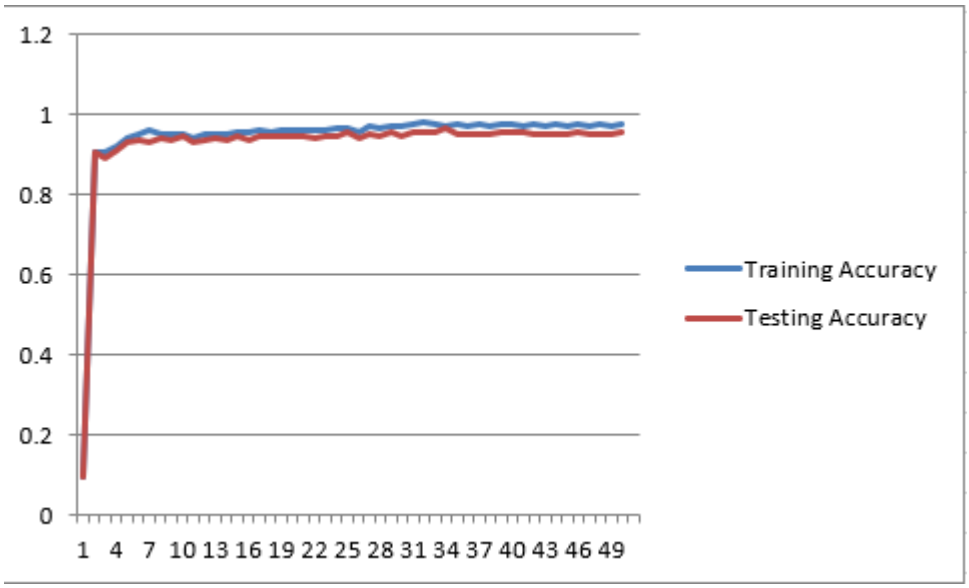
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for n=50:

The final accuracy obtained was 95-96%.

Graph:



Confusion Matrix:

```
[[ 964    0    0    3    1    2    2    3    2    3]
 [   0 1118    3    4    0    0    2    0    8    0]
 [  10    1  976   15    5    1    6   11    6    1]
 [   1    0    9  971    1    7    1   10    8    2]
 [   2    1    3    0  949    0    8    1    6   12]
 [  10    1    0   30    4  815   14    4    9    5]
 [  13    3    2    2    5    7  919    0    7    0]
 [   4    8   13    7    3    0    0  979    4   10]
 [  12    3    5   20    5   11    2    5  908    3]
 [  11    4    0   13   20    4    1   14   11  931]]
```

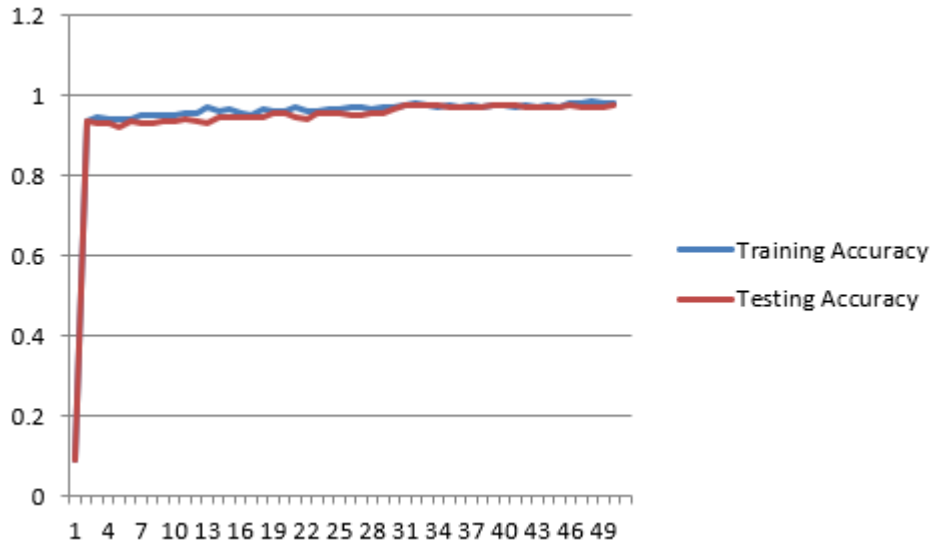
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for n=100:

The final accuracy obtained was 96-97%

Graph:



Confusion Matrix:

```
[[ 960    1    1    3    1    2    5    4    2    1]
 [   0 1121    4    3    0    1    1    0    5    0]
 [   6    1 1000    3    5    0    3    8    6    0]
 [   0    1    4  990    0    4    0    4    4    3]
 [   1    0    4    1  944    0    4    5    3   20]
 [   4    0    1   19    2  850    7    3    4    2]
 [   5    3    1    1    5    6  931    1    4    1]
 [   2    6   15    2    2    0    0  994    1    6]
 [   4    1    9    7    3    3    7    3  934    3]
 [   8    3    0    7   13    3    1    4    5  965]]
```

Discuss your results in a paragraph in your report. Include answers to the following questions:

In this experiment we have checked the accuracy for different number of hidden layers and have seen that as the number of hidden layers increases the accuracy rate also increases, It is evident from the fact that we have got the accuracy of about 97% with the 100 layers where as we have got 93-94% for 20 and 95-96% for 50 hidden layers.

Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

(1) How does the number of hidden units affect the final accuracy on the test data?

As discussed above the accuracy increases as the number of hidden layers increases.

(2) How does it affect the number of epochs needed for training to converge?

As the number of layers increases it takes less number of epochs to converge, the 20-hidden layer took around 37-40 epochs to reach the maximum accuracy and then was revolving around the maximum, for n=50 it took around 25-30 epochs and for n=100 it took around 20-22 epochs to converge. Also, as the number of the layers increases the accuracy reaches from minimum value at epoch 0 to comparatively large value at epoch 1, as compared to the one with less number of hidden units.

(3) Is there evidence that any of your networks has over-fit to the training data? If so, what is that evidence?

For few of the runs and as it can be seen from above graphs as well, the training accuracy is more than that of test accuracy, so, yes it over-fitted sometimes.

(4) How do your results compare to the results obtained by your perceptron in HW 1?

The accuracy achieved in this experiment was more than that of perceptron and the convergence was also happening earlier as compared to that of perceptron. The accuracy in this case was 94-97% and for perceptron it was 90-92 %, also, the convergence for perceptron occurred late around 50-55 epochs and for neural network with hidden layer it was 22-40 epochs.

Experiment 2- Vary the momentum value.

In this experiment, we have to check the accuracy of the neural network by varying the momentum and keep the hidden layer number and learning rate as 100 and 0.1.

We will test the training and test accuracy for 0, 0.25 and 0.5 momentum value. For all these, we have to calculate accuracies for each epoch and have to plot graph of test and training accuracies, and we also need to create confusion matrix at the end of the training on the test data.

We will be using MNIST data set which will be having 784 inputs and 10 outputs in the size 28 * 28. There will be 60,000 training data and 10,000 test data.

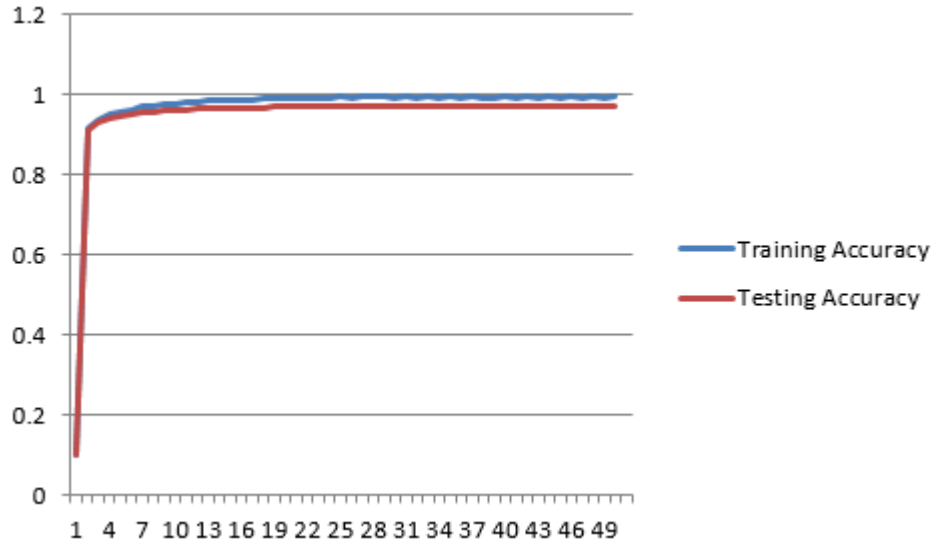
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for momentum=0

The final accuracy obtained was 97-98 %

Graph:



Confusion Matrix:

```
[[ 964    0    0    3    0    4    2    2    2    3]
 [   0 1124    4    1    0    1    3    1    1    0]
 [   5    2 1001    3    3    1    3    8    6    0]
 [   0    0    6  990    0    3    1    4    3    3]
 [   1    0    2    1  953    0    5    2    2   16]
 [   2    1    3   17    2  855    6    1    4    1]
 [   6    2    3    1    3    7  931    1    4    0]
 [   0    9   10    5    2    1    0  990    2    9]
 [   4    0    4    7    5    5    3    4  941    1]
 [   4    3    0   11   18    3    2    8    3  957]]
```

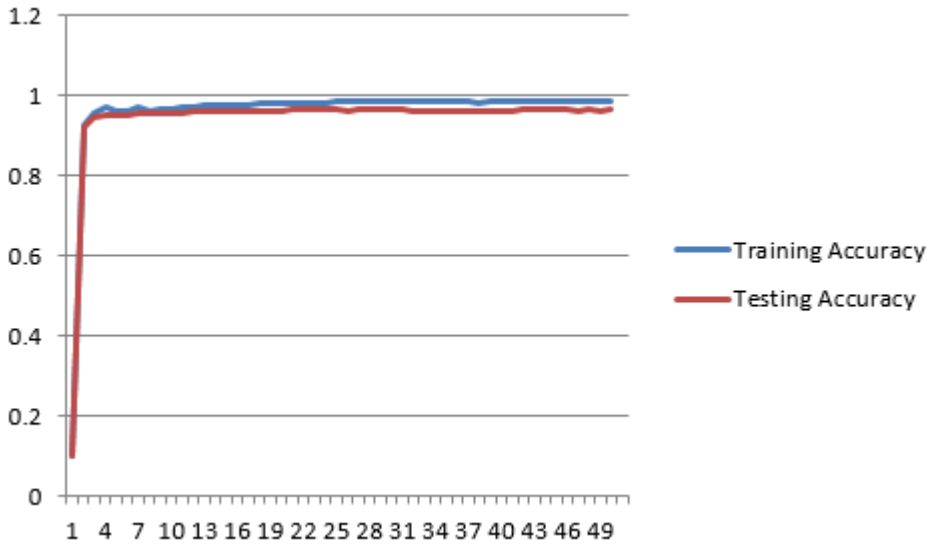
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for momentum=0.25

The final accuracy obtained was 96.45 % (sometimes goes to 97%)

Graph:



Confusion Matrix:

```
[[ 966    0    2    1    0    2    3    2    3    1]
 [   0 1121    3    2    0    1    4    1    3    0]
 [   5    0 1004    4    2    0    1    8    7    1]
 [   0    0    3  985    0    6    1    8    6    1]
 [   1    0    5    0  954    0    4    2    2   14]
 [   4    1    1   12    1  863    5    0    3    2]
 [   6    2    2    1    2    4  938    0    3    0]
 [   1    8   14    5    2    0    0  987    1   10]
 [   4    1    5    7    4    3    5    3  940    2]
 [   4    4    0    7   14    4    1    7    3  965]]
```

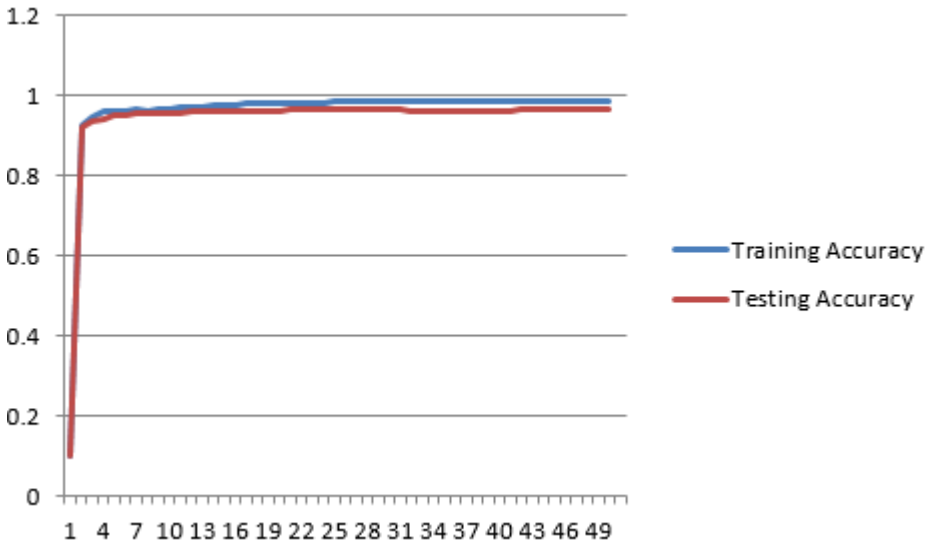
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for momentum=0.5

The final accuracy obtained was 96.7 % (sometimes goes to 97%)

Graph:



Confusion Matrix:

```
[[ 966    0    0    1    0    4    5    1    2    1]
 [   0 1118    2    2    0    1    5    1    6    0]
 [   4    2 1007    3    3    0    1    7    5    0]
 [   0    0    9  984    0    3    0    5    7    2]
 [   1    1    6    1  953    0    3    2    1   14]
 [   2    1    0   14    3  855    7    2    4    4]
 [   6    2    2    1    2    7  932    0    6    0]
 [   2    8   15    3    0    0    0  990    1    9]
 [   4    1    5    6    6    4    4    1  941    2]
 [   3    4    2   10   12    5    1    8    4  960]]
```

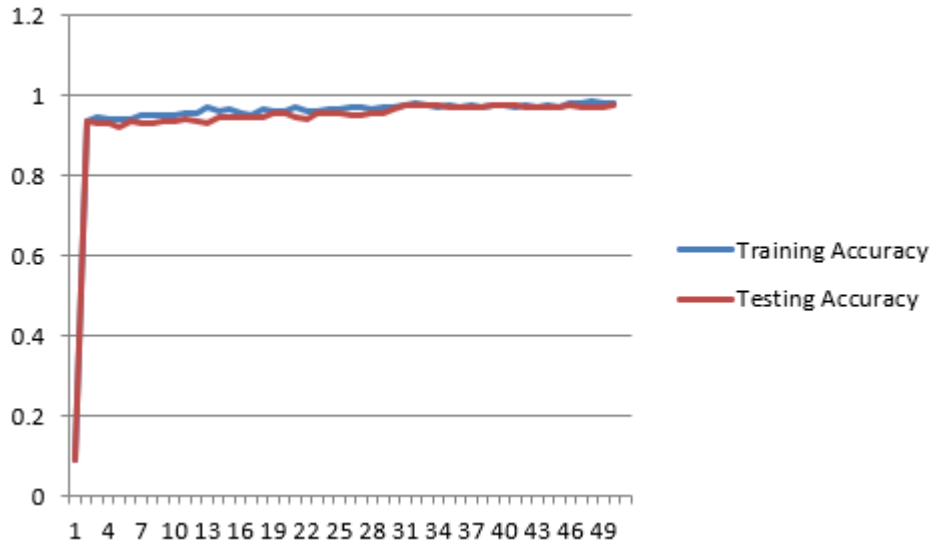
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for momentum=0.9

The final accuracy obtained was 96.9 % (sometimes goes to 97%)

Graph:



Confusion Matrix:

```
[[ 960    1    1    3    1    2    5    4    2    1]
 [   0 1121    4    3    0    1    1    0    5    0]
 [   6    1 1000    3    5    0    3    8    6    0]
 [   0    1    4  990    0    4    0    4    4    3]
 [   1    0    4    1  944    0    4    5    3   20]
 [   4    0    1   19    2  850    7    3    4    2]
 [   5    3    1    1    5    6  931    1    4    1]
 [   2    6   15    2    2    0    0  994    1    6]
 [   4    1    9    7    3    3    7    3  934    3]
 [   8    3    0    7   13    3    1    4    5  965]]
```

Discuss your results in a paragraph in your report. Include answers to the following questions:

In this experiment we can conclude that the accuracy obtained was maximum for the one where there was no momentum i.e. 0, and for other values the accuracy was almost same as we have reached the convergence point with the difference that one with the lower value of momentum converged last and one with higher value converged first.

Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

(1) How does the momentum value affect the final accuracy on the test data?

The accuracy obtained was maximum for the one where there was no momentum i.e. 0, and for other values the accuracy was almost same (very little difference) as we have reached the convergence point. For .25-96.45, .5-96.7 and for .9-96.9. For 0 it was between 97-98.

(2) How does it affect the number of epochs needed for training to converge?

The network which was using the higher value of the momentum converged earlier i.e. with .9 value converged in 20-22 epochs, .5 – 25-27 epochs, .25 – 30-34 epochs, 0- 37-39 epochs.

(3) Again, is there evidence that any of your networks has overfit to the training data? If so, what is that evidence

We can say that there was overfitting as for some instances the testing accuracies was less than that of the training accuracies. It can be seen from the graph as well.

Experiment 3 - Vary the number of training examples.

In this experiment, we will check the accuracy of the neural network by varying the training set and keep the momentum as .9 and hidden layer number as 100.

We will test the training and test accuracy for one quarter and one half of the training set (even distribution of 0-9-digit classes in both the sets). For both these cases, we will calculate accuracies for each epoch and will plot graph of test and training accuracies, and we also need to create confusion matrix at the end of the training on the test data.

We will be using MNIST data set which will be having 784 inputs and 10 outputs in the size 28 * 28. There will be 60,000 training data and 10,000 test data.

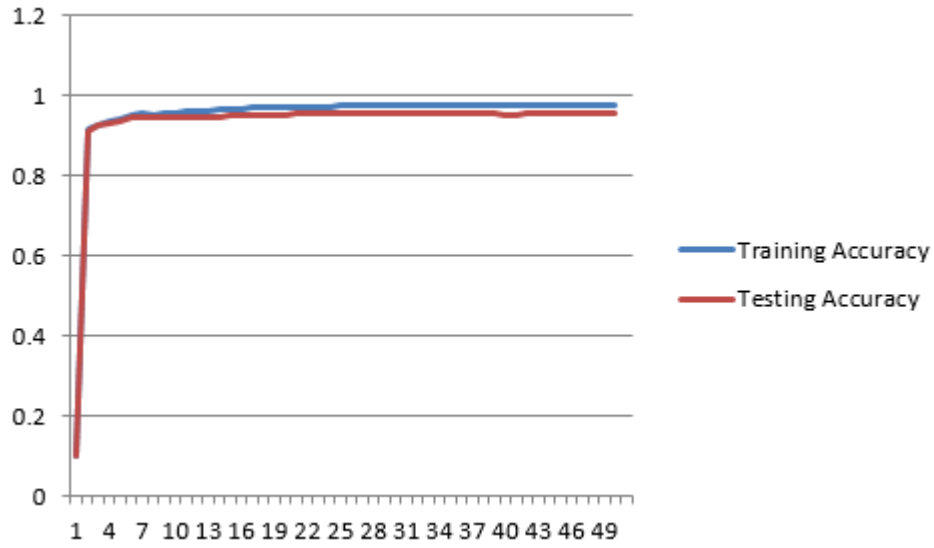
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for Half

The final accuracy obtained was 96.4%

Graph



Confusion Matrix

```
[[ 967    1    0    1    0    3    3    3    1    1]
 [   0 1126    3    2    0    0    1    0    3    0]
 [  10    2  993    2    5    0    5    9    5    1]
 [   0    1    8  980    0    4    1    8    5    3]
 [   2    1    2    0  941    0   10    4    3   19]
 [   7    0    2   18    2  843   10    3    3    4]
 [   8    3    3    1    6    3  929    0    5    0]
 [   1    8   12    4    2    1    0  989    1   10]
 [   7    1    5    8    4    7    4    6  927    5]
 [   6    7    1    7   21    2    1   10    3  951]]
```

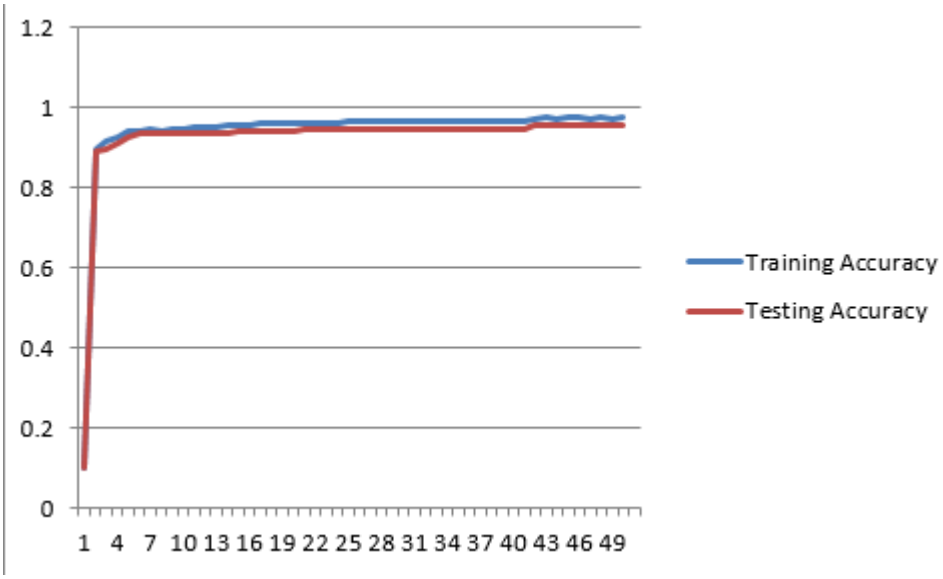
Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

Graph and confusion matrix for quarter

The final accuracy obtained was 96.6

Graph



Confusion Matrix

```
[[ 965    1    2    0    1    1    4    2    4    0]
 [   0 1115    5    4    0    2    3    1    5    0]
 [  10    1  983   12    6    1    6    8    4    1]
 [   1    1   16  962    3    6    0    7    8    6]
 [   2    0    3    0  935    0   11    1    6   24]
 [   5    1    1   34    1  823   10    1   11    5]
 [  11    3    2    2   10    6  917    1    5    1]
 [   3    7   17    2    6    0    0  977    1   15]
 [   5    1    3   14    6    5    8    7  921    4]
 [   3    6    0   10   22    9    1   13    9  936]]
```

Discuss your results in a paragraph in your report. Include answers to the following questions:

For both the cases the value of the accuracy was quite similar towards the end, but for the quarter train set there was more overfitting as compared to the half set and the convergence happened little earlier for the quarter set.

Name – Amarjit Jadhav PSU ID – 963839497

MACHINE LEARNING PROGRAMMING #1

(1) How does the size of the training data affect the final accuracy on the test data?

The Final accuracies for the both the sets (quarter and half) was almost same for few runs and for few runs there was little difference, for half it was slightly more than that of quarter, for quarter it was 95.6-95.8 and for half it was 95.9 -96.2.

(2) How does it affect the number of epochs needed for training to converge?

The epochs for the convergence for the quarter was around 24-26 and for the half was 28-30, So for quarter the convergence occurred earlier comparatively.

(3) Again, is there evidence that any of your networks has overfit to the training data? If so, what is that evidence.

Yes, there was overfitting as can be seen from graph the training accuracies are more than the testing accuracies.