Programming Assignment 1

CS 545: Machine Learning, Spring 2024

Srihari Tanmay Karthik Tadala

PSU ID - 918597835

REPORT

Experiment 1: Vary number of hidden units.
Do experiments with *n* = 20, 50, and 100. (Remember to also include a bias unit with weights to every hidden and output node.)

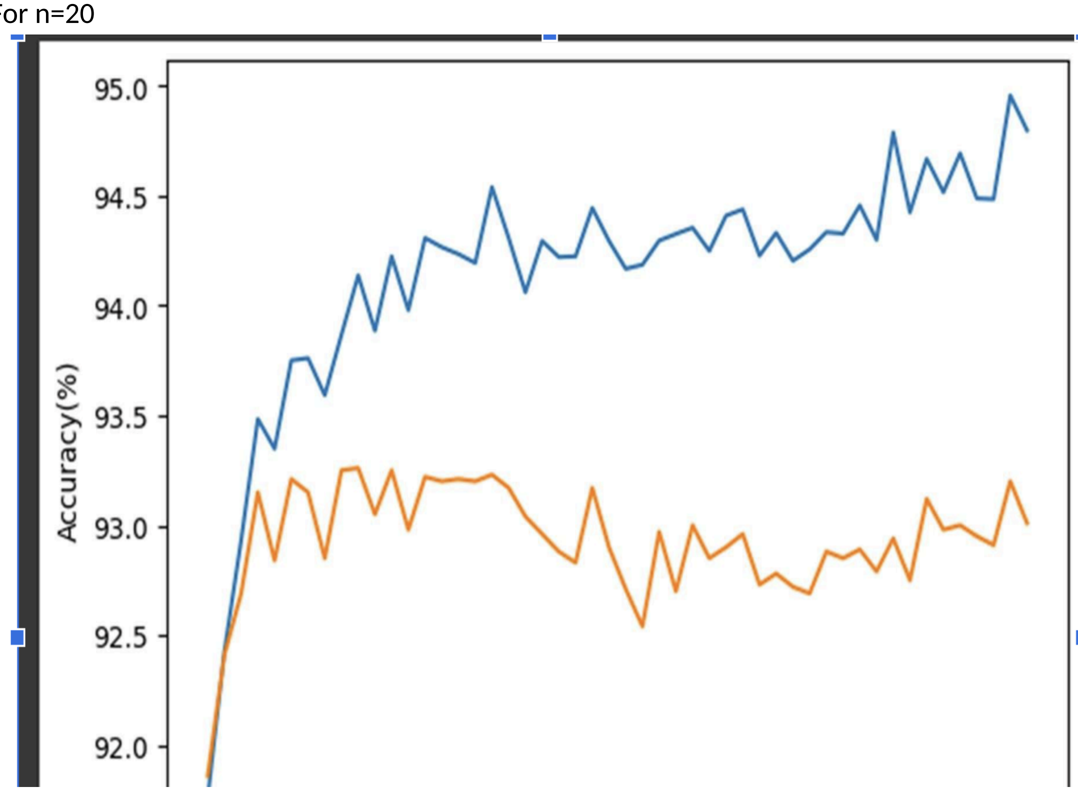
For each value of *n,* train your network on the training set, as described above, changing the weights after each training example. After each epoch, calculate the network's accuracy on the training set **and** the test set for your plot. Train your network for 50 epochs. In your report, give a plot of both training and test accuracy as a function of epoch number (graph both of these in the same plot).

After training is complete, create a confusion matrix for each of your trained networks, summarizing results on the test set.

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

Do experiments with n = 20, 50, and 100.

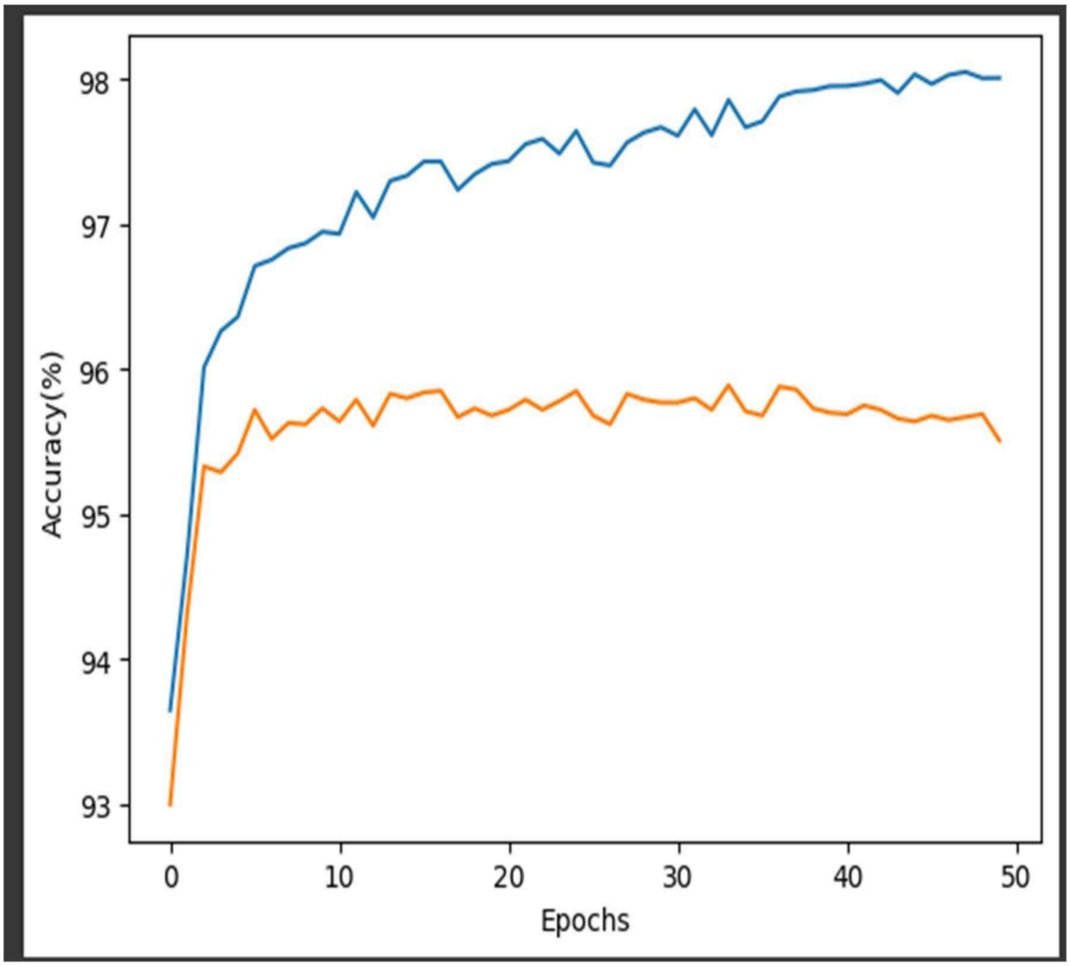
1) For n=20



```
CONFUSION MATRIX ON TEST DATA:
 [[ 952    0    0    0    0    9    8    2    8    1]
 [    0 1113    3    3    1    0    4    1    9    1]
 [   11    5  923   36    6    5   11   10   18    7]
 [    2    0    4  948    0   17    2    9   17   11]
 [    2    2    3    0  889    0   13    1    3   69]
 [    7    3    2   32    2  779   13    3   39   12]
 [   12    3    5    2    7   15  896    0   18    0]
 [    2    5   18    9    5    2    1  931    8   47]
 [    6    9    2    8    5    9    8    5  917    5]
 [    6    5    2   11    9    3    0    5   15  953]]
```

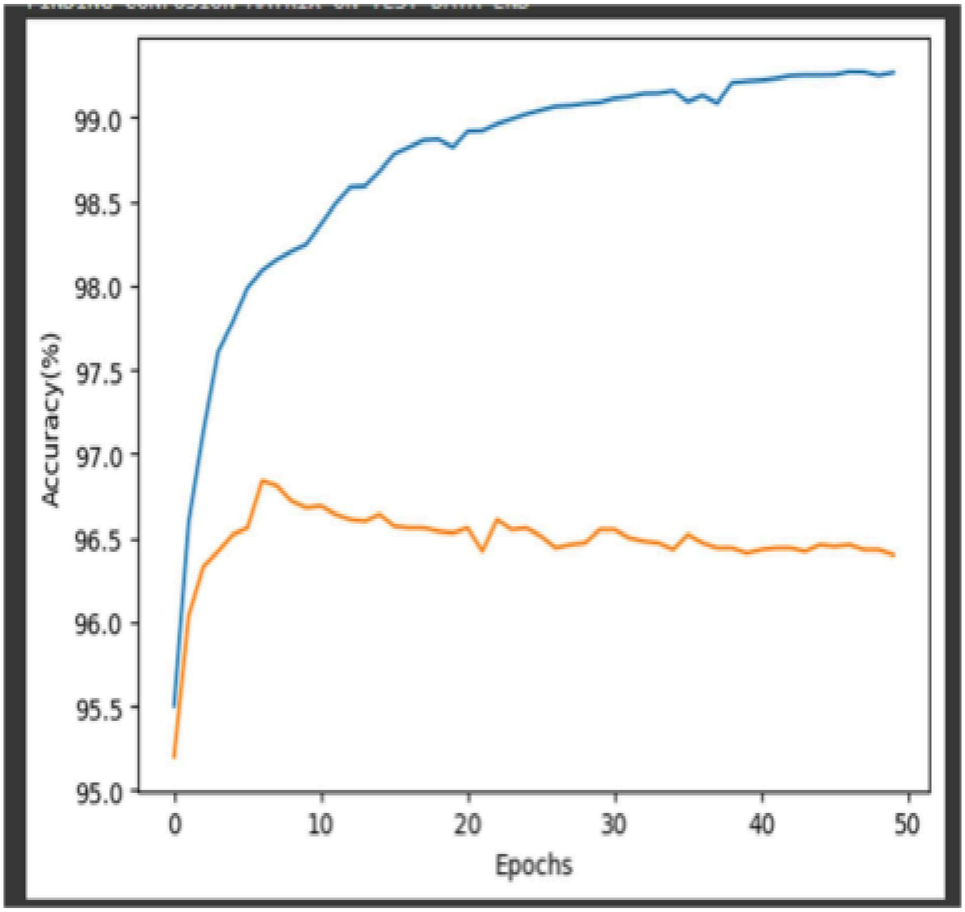
2) For n=50

```
CONFUSION MATRIX ON TEST DATA:
 [[ 961    0    2    1    3    2    0    4    7    0]
 [    0 1113    1    4    0    1    3    1   12    0]
 [    6    6  973   11    2    2    4    8   19    1]
 [    1    0    9  972    0    8    0    4   11    5]
 [    3    1    3    2  921    0    8    1    4   39]
 [    3    1    0   21    1  829    9    3   20    5]
 [   11    4    1    0    2    8  923    0    9    0]
 [    0    4   16   15    2    1    0  962    9   19]
 [    4    0    2    3    5   10    6    3  937    4]
 [    5    6    0   13   10    2    2    5    6  960]]
```



98
97
96
95
94
93
Accuracy(%)
0
10
20
30
40
50
Epochs

3) For n=100




```
CONFUSION MATRIX ON TEST DATA:
 [[ 969    1    1    1    0    0    4    2    1    1]
 [    0 1120    4    1    1    1    1    0    7    0]
 [    6    2  990   10    1    2    2    5   13    1]
 [    3    2    2  973    1   13    0    7    6    3]
 [    2    0    4    0  948    0    3    2    1   22]
 [    8    1    1    9    0  845    8    2   13    5]
 [    8    3    1    1    2   11  926    0    5    1]
 [    0    5   15    4    6    0    0  976    8   14]
 [    4    2    2    6    3    7    6    3  934    7]
 [    5    6    0   11   11    3    2    5    7  959]]
```

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

questions:

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

Ans- The test data's accuracy improves as the number of hidden layers increases, as demonstrated by the graphs above.

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

Ans- Adding more hidden layers can result in faster convergence, as demonstrated by the fact that n=50 achieves faster convergence than n=20. Nevertheless, continuing to add more hidden layers beyond a specific threshold leads to a slowdown in the rate at which the model converges. This implies that increasing the number of hidden layers does not necessarily result in better convergence. Determining the ideal number of hidden units is essential for achieving optimal performance. Within this context, a setup featuring a value of n=50 demonstrated superior performance compared to configurations with smaller (n=20) and larger (n=100) quantities of hidden layers.

3). Is there evidence that any of your networks has overfit to the training

data? Ifso, what is that evidence?

Ans-There is a small degree of overfitting observed when n=100.

4). How do your results compare to the results obtained by your

perceptron in HW1?

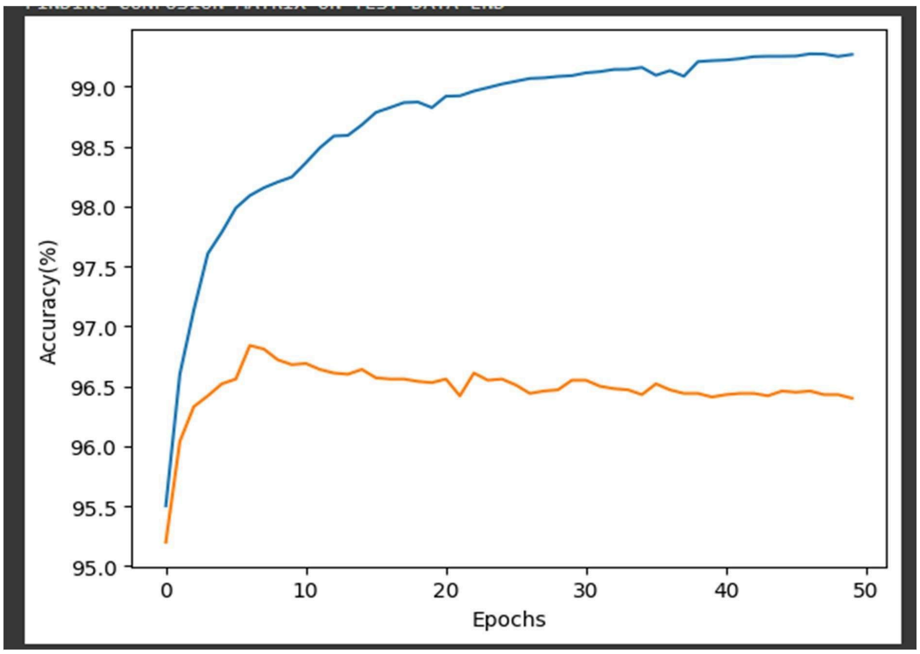
Ans- By incorporating hidden layers, we were able to significantly improve the accuracy compared to Homework 1.

Experiment 2: Vary the momentum value. Here, fix the number of hidden units to 100, and vary the momentum value during training. Use momentum values of 0, 0.25, and 0.5.
Train networks using these values as in Experiment 1, and plot the results as in Experiment 1. (Include your plot for *n*=100 and momentum = 0.9 that you completed in Experiment 1.)

Create a confusion matrix for each of your trained networks, summarizing results on the test set.

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

From Experiment 1 , for n=100 and momentum = 0.9




```
CONFUSION MATRIX ON TEST DATA:
 [[ 969    1    1    1    0    0    4    2    1    1]
 [    0 1120    4    1    1    1    1    0    7    0]
 [    6    2  990   10    1    2    2    5   13    1]
 [    3    2    2  973    1   13    0    7    6    3]
 [    2    0    4    0  948    0    3    2    1   22]
 [    8    1    1    9    0  845    8    2   13    5]
 [    8    3    1    1    2   11  926    0    5    1]
 [    0    5   15    4    6    0    0  976    8   14]
 [    4    2    2    6    3    7    6    3  934    7]
 [    5    6    0   11   11    3    2    5    7  959]]
```

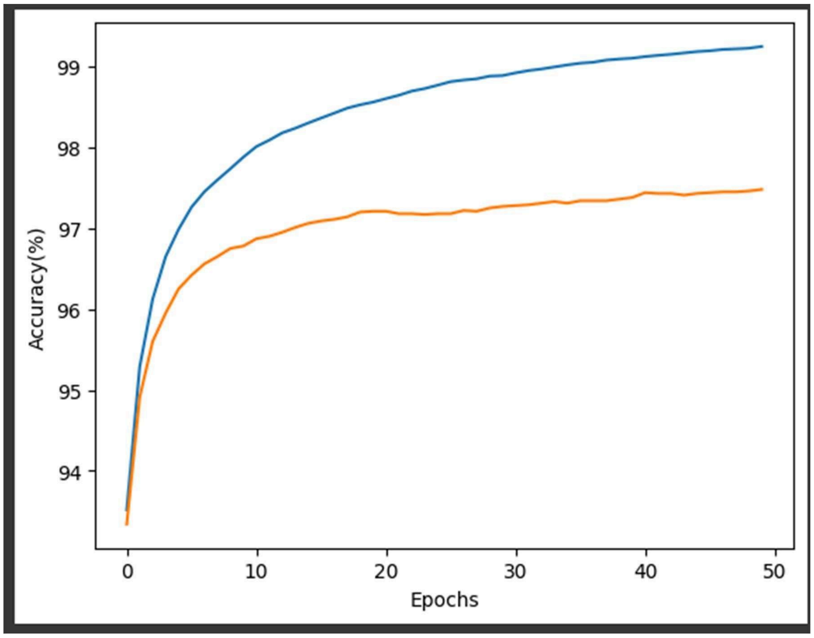
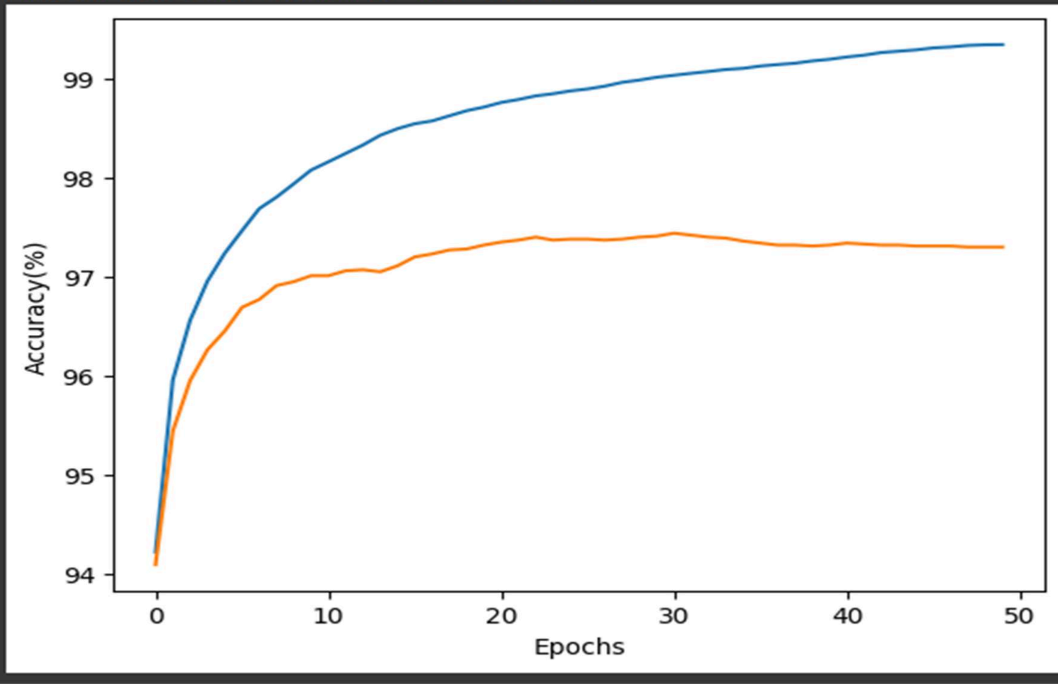


Image for Momentum = 0   the top and bottom image

```
CONFUSION MATRIX ON TEST DATA:
 [[ 969    1    1    0    0    2    4    1    2    0]
 [    0 1122    2    3    0    1    3    0    4    0]
 [    2    0 1000    8    1    0    3    7    7    4]
 [    1    1    1  988    0    3    0    4    4    8]
 [    1    0    3    0  958    0    4    0    0   16]
 [    3    0    1   10    1  853   15    1    4    4]
 [    6    3    0    1    1    1  940    0    6    0]
 [    0    4   18    4    1    0    1  981    5   14]
 [    4    0    0    4    4    2    2    2  953    3]
 [    2    5    1    4    5    2    1    4    1  984]]
```

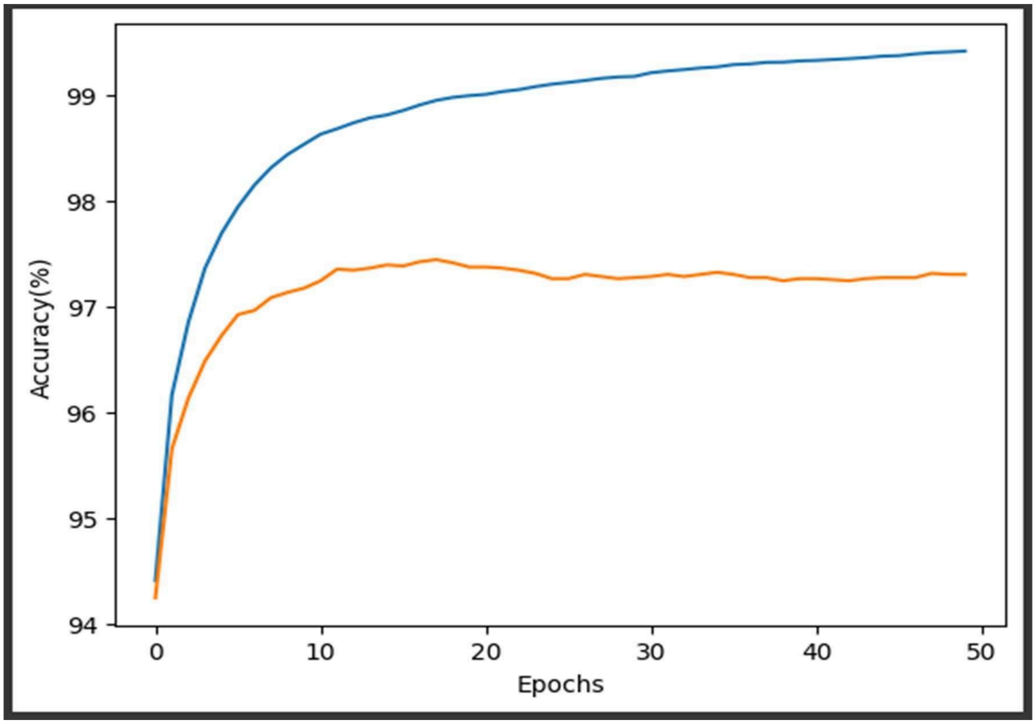
Image for Momentum = 0.25

```
FINDING CONFUSION MATRIX ON TEST DATA END
```




```
CONFUSION MATRIX ON TEST DATA:
 [[ 972    1    0    1    0    0    2    2    2    0]
 [    0 1123    3    2    0    1    2    1    3    0]
 [    2    3  999    6    2    2    3    8    5    2]
 [    0    0    2  985    1    8    0    5    3    6]
 [    1    0    3    0  950    0    5    0    2   21]
 [    5    1    0   12    0  857    6    2    3    6]
 [    8    4    0    2    3    6  932    0    3    0]
 [    1    2   13    4    3    0    0  986    3   16]
 [    3    1    1    6    2    1    3    1  951    5]
 [    2    6    0    6    8    6    0    5    1  975]]
```

<u>Image for Momentum = 0.5</u>




```
CONFUSION MATRIX ON TEST DATA:
 [[ 971    1    1    2    0    1    1    1    1    1]
 [   0 1122    2    5    0    0    1    1    3    1]
 [   1    1 1003    9    1    0    2    8    6    1]
 [   1    0    2  989    0    5    0    2    6    5]
 [   1    0    1    1  951    0    6    0    2   20]
 [   5    2    0   15    1  844   11    2    5    7]
 [   5    3    1    1    0    4  938    0    5    1]
 [   0    7   11    3    1    0    0  990    1   15]
 [   4    2    0    6    4    2    3    2  946    5]
 [   3    4    0    5    6    2    0    6    7  976]]
```

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

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

Ans- Conversely, the accuracy remained consistent regardless of the momentum values.

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

Ans- A higher momentum value (n=0.9) leads to a superior convergence rate in comparison to lower momentum. This suggests that greater momentum values have a significant impact on the convergence of the data set.

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

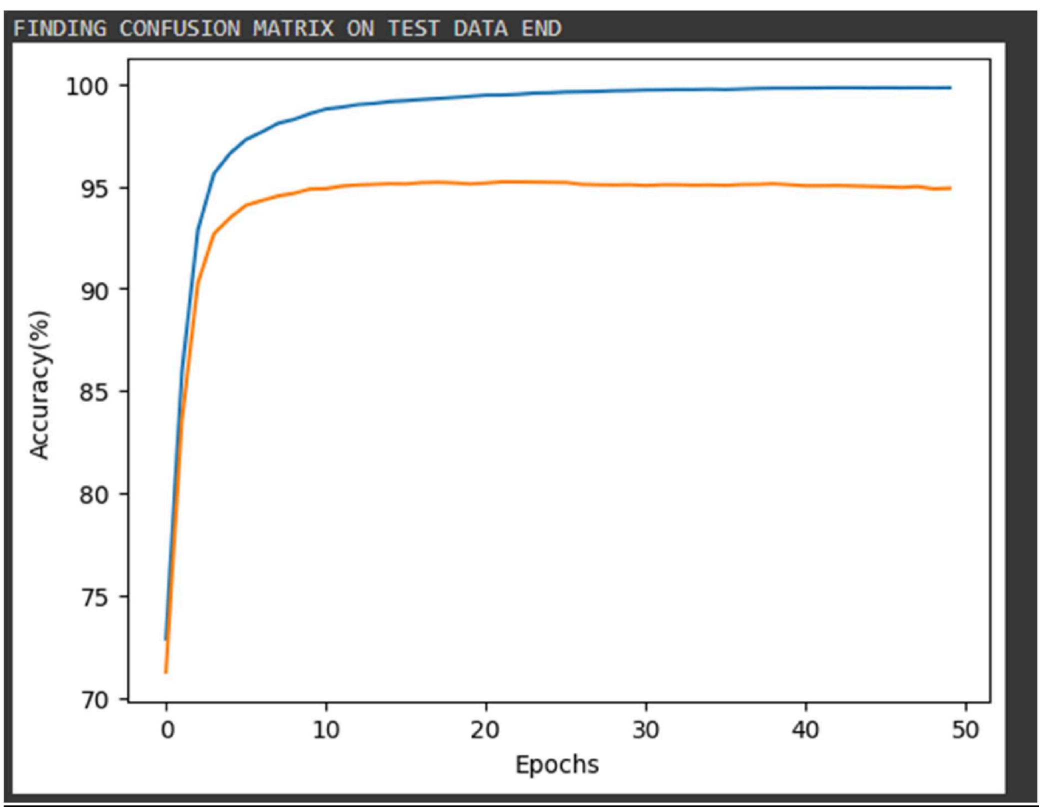
Ans- No overfitting is found.

Experiment 3: Vary the number of training examples. In this experiment, fix the number of hidden units to 100 and momentum 0.9. Instead of using all of the training examples, train two networks, using respectively one quarter and one half of the training examples for training. Make sure that in each case your training data is approximately balanced among the 10 different classes. Plot the results, as in the previous experiments, plotting accuracy on both the training and test data at the end of each epoch.

Create a confusion matrix for each of your trained networks, summarizing results on the test set.

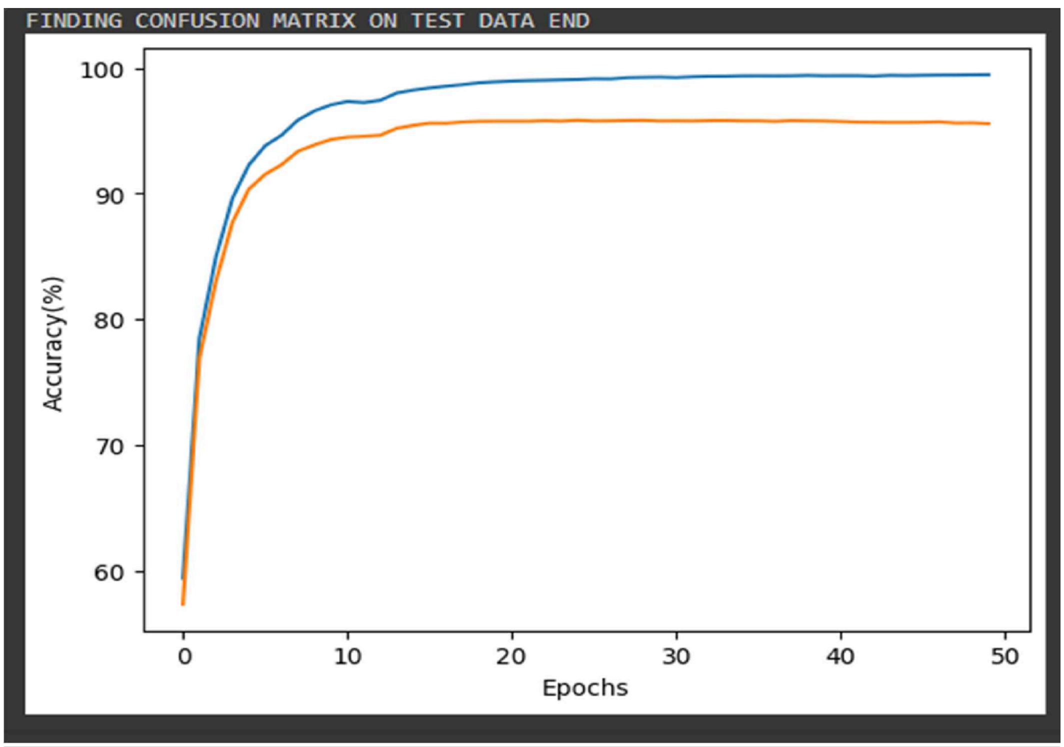
Discuss your results in a paragraph in your report. Include answers to the following questions:

<u>Images for Data Set = QUARTER</u>



```
CONFUSION MATRIX ON TEST DATA:
 [[ 963    0    3    1    0    4    1    2    5    1]
 [   1 1110    3    5    0    2    3    2    8    1]
 [   9    1  980    7    3    4    0    7   18    3]
 [   2    0   16  932    1   21    0    5   26    7]
 [   2    2    2    1  925    0    6    4    5   35]
 [   5    0    2   13    1  843    8    2   16    2]
 [  17    3    3    2    9   13  900    0   10    1]
 [   1    5   17    4    5    3    1  959    9   24]
 [   7    0    4    8    4   10    2    4  928    7]
 [   3    4    2    7    9    8    0   10   16  950]]
```

Images for Data Set = HALF



```
CONFUSION MATRIX ON TEST DATA:
 [[ 959    0    2    2    0    6    1    1    4    5]
 [    0 1107    5    2    0    4    3    2   11    1]
 [    2    0  982    5    3    5    1   11   19    4]
 [    1    0   21  920    0   34    1    4   25    4]
 [    1    1    3    1  945    0    5    1    4   21]
 [    3    0    1   12    0  863    3    2    8    0]
 [   10    2    4    2    3   15  907    1   13    1]
 [    1    2   14    5    7    4    0  977    3   15]
 [    6    0    1    1    3    8    2    2  949    2]
 [    4    3    1   10   11    7    1    4   22  946]]
```

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

questions:

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

Ans- As the size of the data set increased from a quarter to a half, we noticed a marginal improvement in accuracy. Training on the entire data set yielded greater accuracy in comparison to training on partial data sets, such as those comprising half or a quarter of the total data.

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

Ans- It is evident that the training set with half the data size exhibits slightly superior performance compared to the dataset that is one-fourth the size.

3). Again, is there evidence that any of your networks has overfit

to the training data? If so, what is that evidence?

Ans- No evidence of overfitting is seen.