

## Machine Learning

This assignment was written in Python. MNSIT data was taken from the csv files. The number of epochs is set to 50. After each epoch the data was permuted randomly. The program can be divided into 2 parts:

- Forward propagation
- Backpropagation

Backpropagation is initiated only when the data is of type 'Training' and epoch! = 0 .

On the 49<sup>th</sup> epoch we store the prediction of the network as to which neuron fires and the corresponding labels and create a confusion matrix.

### Experiment 1: Vary number of hidden units

This experiment involves varying the hidden units with n=20,50,100. We then plot both training and test accuracy as a function of epoch number as below.

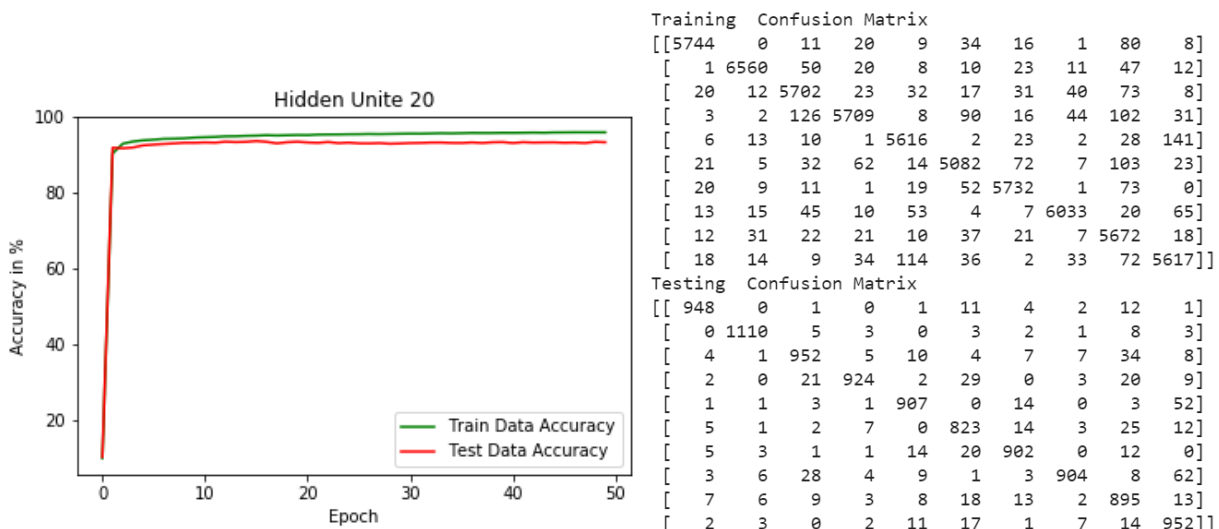
Constant parameters:

Momentum=0.9

Learning rate=0.1

Number of epochs=50

#### Hidden Units: 20

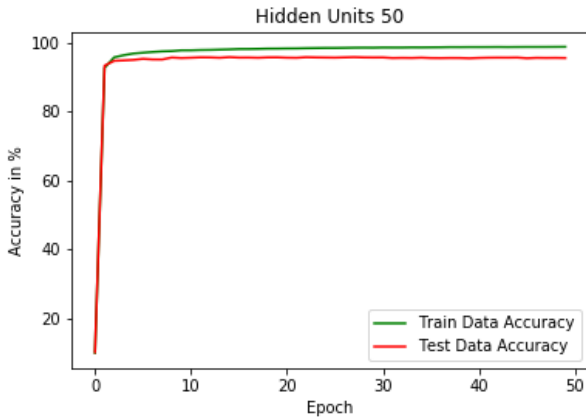


Accuracy of Training data: [9.931666666666667, 90.225, 92.80166666666668, 93.36666666666666, 93.67666666666666, 93.845, 94.06666666666666, 94.11333333333334, 94.19999999999999, 94.40333333333332, 94.49499999999999, 94.57833333333333, 94.72833333333334, 94.75333333333333, 94.855, 94.90833333333333, 95.02333333333334, 94.94500000000001, 95.01166666666667, 95.08, 95.05, 95.18666666666667, 95.21333333333334, 95.22833333333334, 95.27, 95.28833333333333, 95.34666666666666, 95.30999999999999, 95.35833333333333, 95.40666666666667, 95.43833333333333, 95.42666666666668, 95.48, 95.54166666666667, 95.505, 95.55666666666667, 95.625, 95.60333333333332, 95.60666666666665, 95.65333333333334, 95.64166666666667, 95.67166666666667, 95.71, 95.67166666666667, 95.74666666666667, 95.76, 95.78666666666666, 95.78333333333333, 95.78833333333333, 95.77833333333334]

Accuracy of Test data : [10.32, 91.73, 91.57, 91.79, 92.36, 92.55, 92.72, 92.86, 93.03, 93.02, 93.12, 93.06, 93.32000000000001, 93.21000000000001, 93.28, 93.47999999999999, 93.28, 92.86999999999999, 93.17999999999999, 93.31, 93.11, 93.01, 93.24, 92.93, 93.07, 92.88, 92.86, 92.94, 92.77, 92.84, 92.95, 92.99, 93.08999999999999, 93.10000000000001, 92.99, 92.97, 93.12, 92.97, 93.16, 93.21000000000001, 92.96, 93.19, 93.08, 93.12, 93.14, 93.03, 93.08999999999999, 92.95, 93.27, 93.17]

The convergence takes place after the 11<sup>th</sup> epoch. The final accuracy after 50 epochs is nearly 93%

### Hidden Units:50



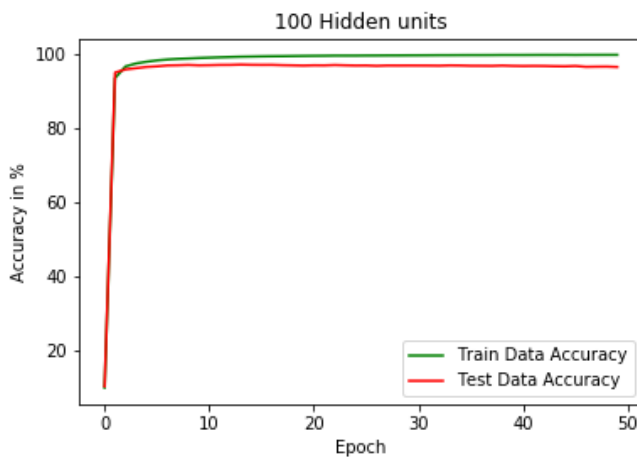
Training Confusion Matrix									
[	5896	0	0	0	0	1	2	0	21
[	1	6722	5	0	2	1	2	2	5
[	5	0	5937	2	1	1	2	3	6
[	0	0	9	6092	0	2	0	2	15
[	2	0	4	0	5826	0	0	1	2
[	2	1	2	9	2	5386	7	0	9
[	2	0	1	0	1	0	5904	0	9
[	0	2	8	1	0	0	0	6241	3
[	4	0	3	1	0	0	2	0	5839
[	5	0	1	10	4	3	0	3	18
Testing Confusion Matrix									
[	968	0	1	1	1	1	2	1	3
[	1	1120	3	1	0	1	2	1	6
[	4	3	987	3	2	1	3	7	17
[	0	1	7	979	1	5	0	3	9
[	1	1	5	0	929	0	7	2	3
[	5	1	0	15	1	844	5	1	11
[	4	4	2	0	1	7	930	0	10
[	0	6	13	4	5	1	0	977	6
[	2	1	3	5	4	5	5	5	935
[	6	5	1	10	11	3	0	3	9

Accuracy of Train data : [9.915000000000001, 92.68666666666667, 95.73333333333333, 96.36166666666666, 96.84, 97.075, 97.28833333333333, 97.46333333333334, 97.54833333333333, 97.76666666666667, 97.775, 97.86, 97.895, 97.97166666666666, 98.04, 98.14500000000001, 98.15, 98.19833333333334, 98.265, 98.275, 98.31, 98.32333333333332, 98.38, 98.41333333333333, 98.45, 98.45166666666667, 98.48666666666666, 98.53666666666666, 98.55833333333334, 98.54333333333334, 98.595, 98.58833333333334, 98.58833333333334, 98.62666666666667, 98.615, 98.63833333333332, 98.65833333333333, 98.7, 98.695, 98.71166666666666, 98.71, 98.71833333333333, 98.735, 98.71166666666666, 98.73333333333333, 98.74666666666667, 98.74666666666667, 98.75333333333333, 98.76666666666667, 98.785]

Accuracy of Test data : [10.09, 93.32000000000001, 94.72, 94.84, 94.99, 95.3, 95.13000000000001, 95.11, 95.67999999999999, 95.53, 95.62, 95.73, 95.71, 95.61, 95.78999999999999, 95.66, 95.67, 95.61, 95.72, 95.73, 95.63000000000001, 95.6, 95.77, 95.71, 95.67999999999999, 95.65, 95.72, 95.78999999999999, 95.71, 95.69, 95.7, 95.53, 95.58, 95.55, 95.64, 95.53, 95.50999999999999, 95.55, 95.54, 95.46, 95.57, 95.63000000000001, 95.65, 95.64, 95.67, 95.47, 95.59, 95.56, 95.58, 95.54]

The convergence takes place after the 9<sup>th</sup> epoch. The final accuracy after 50 epochs is nearly 95%

### Hidden Units:100



Training Confusion Matrix

```

[[5887 1 5 1 0 0 2 0 23 4]
 [ 0 6724 6 0 2 1 0 2 6 1]
 [ 5 0 5944 0 0 0 0 1 6 2]
 [ 0 1 6 6093 2 2 0 1 20 6]
 [ 0 2 2 0 5824 0 0 1 5 8]
 [ 2 1 4 7 2 5391 5 0 6 3]
 [ 5 2 0 0 1 1 5899 0 8 2]
 [ 1 3 5 0 0 1 0 6245 1 9]
 [ 1 0 2 1 0 2 5 0 5837 3]
 [ 1 1 1 8 3 1 0 2 17 5915]]

```

Testing Confusion Matrix

```

[[ 965 0 0 3 0 1 2 1 7 1]
 [ 1 1120 1 5 1 1 1 2 3 0]
 [ 4 2 986 8 1 1 1 11 14 4]
 [ 2 0 4 955 1 19 1 3 22 3]
 [ 1 0 3 0 949 0 6 3 6 14]
 [ 4 1 0 10 0 852 10 0 9 6]
 [ 4 3 1 2 3 17 922 0 5 1]
 [ 0 3 8 4 4 1 0 980 12 16]
 [ 7 1 1 1 5 10 4 3 939 3]
 [ 2 6 0 2 10 4 1 5 15 964]]

```

Accuracy Train Data [9.776666666666667, 93.32833333333333, 96.47, 97.27166666666666, 97.73166666666665, 98.09666666666666, 98.335, 98.50666666666666, 98.61333333333333, 98.74166666666667, 98.83833333333332, 98.935, 99.015, 99.09, 99.13166666666666, 99.18, 99.21333333333334, 99.23333333333333, 99.27666666666667, 99.31166666666667, 99.33833333333332, 99.35833333333333, 99.39333333333333, 99.39333333333333, 99.395, 99.41666666666666, 99.425, 99.445, 99.44833333333334, 99.465, 99.47333333333333, 99.48666666666666, 99.50666666666666, 99.50833333333333, 99.52333333333333, 99.53166666666667, 99.53166666666667, 99.53833333333333, 99.55833333333334, 99.55833333333334, 99.55833333333334, 99.57000000000001, 99.58166666666666, 99.575, 99.58833333333334, 99.56666666666666, 99.58833333333334, 99.58833333333334, 99.59833333333333, 99.59833333333333]

Accuracy Test Data [10.16, 94.81, 95.66, 96.0, 96.33, 96.49, 96.74000000000001, 97.89999999999999, 97.88, 97.77, 97.81, 97.88, 97.89, 97.97, 97.91, 97.89, 97.89999999999999, 97.81, 97.76, 97.7, 97.77, 97.75, 97.85000000000001, 97.77, 97.71, 97.73, 97.63000000000001, 97.72, 97.71, 97.72, 97.71, 97.7, 97.67, 97.75, 97.72, 97.65, 97.64, 97.61999999999999, 97.71, 96.71999999999999, 97.58, 97.61, 97.59, 97.52, 97.49, 97.59, 97.36, 97.39999999999999, 97.41999999999999, 97.32]

The convergence takes place after the 8<sup>th</sup> epoch. The final accuracy after 50 epochs is nearly 97%

Hidden Units	Train Accuracy in %	Test Accuracy in %
20	95.77	93.17
50	98.78	95.54
100	99.59	97.32

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

With increase in number of hidden units the final accuracy on the test data also increases.

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

As the number of hidden units are increased from 20 to 100, the number of epochs needed to converge is reduced. With 20 hidden units convergence takes place after 11<sup>th</sup> epoch, with 50 hidden units convergence takes place after 9<sup>th</sup> epoch and with 100 hidden units convergence takes place after 8<sup>th</sup> epoch.

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

Yes, my network has overfit to the training data. From the above three graphs I can see that the test data curve comes below the training data curve after a certain point. The difference in the final accuracy of the test data and training data is nearly 3%.

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

The test and train data accuracy are much more with hidden units when compared to the perceptron of homework 1 where we did not have hidden units. With no hidden units and learning rate of 0.1, single perceptron accuracy on test data was 86% after 70 epochs however with hidden units the test data accuracy has increased to 93% with 20 hidden units ,95% with 50 hidden units and 97% for 100 hidden units.

### Experiment 2: Vary the momentum value

This experiment involves varying the momentum with values 0, 0.25 and 0.5 keeping the number of hidden unit's constant with  $n = 100$ . We then plot both training and test accuracy as a function of epoch number as below.

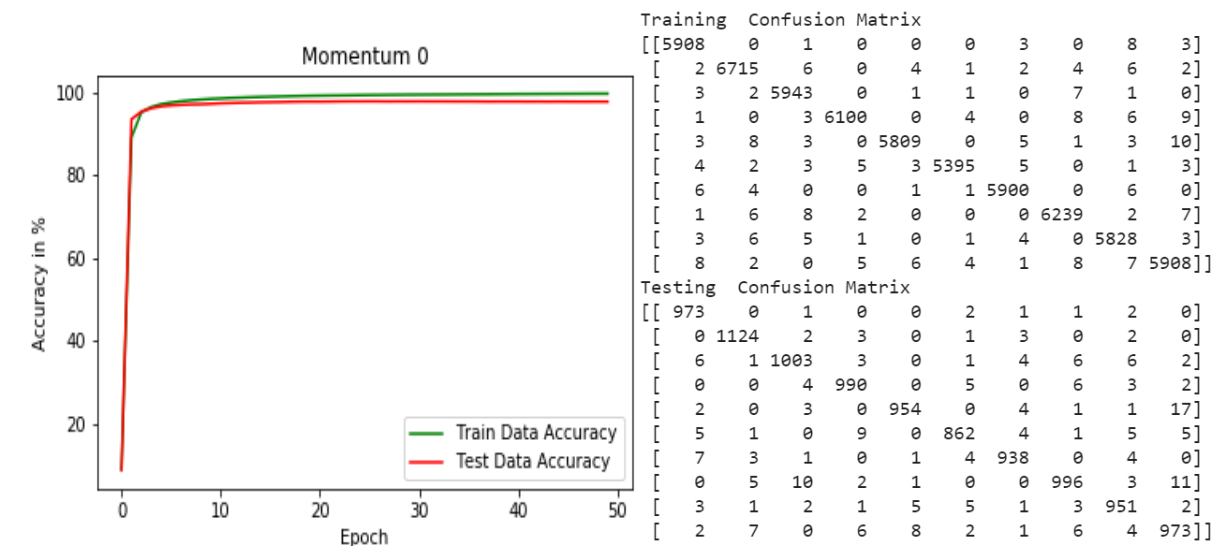
Constant parameters:

Hidden Units=100

Learning rate=0.1

Number of epochs=50

#### **Momentum:0**



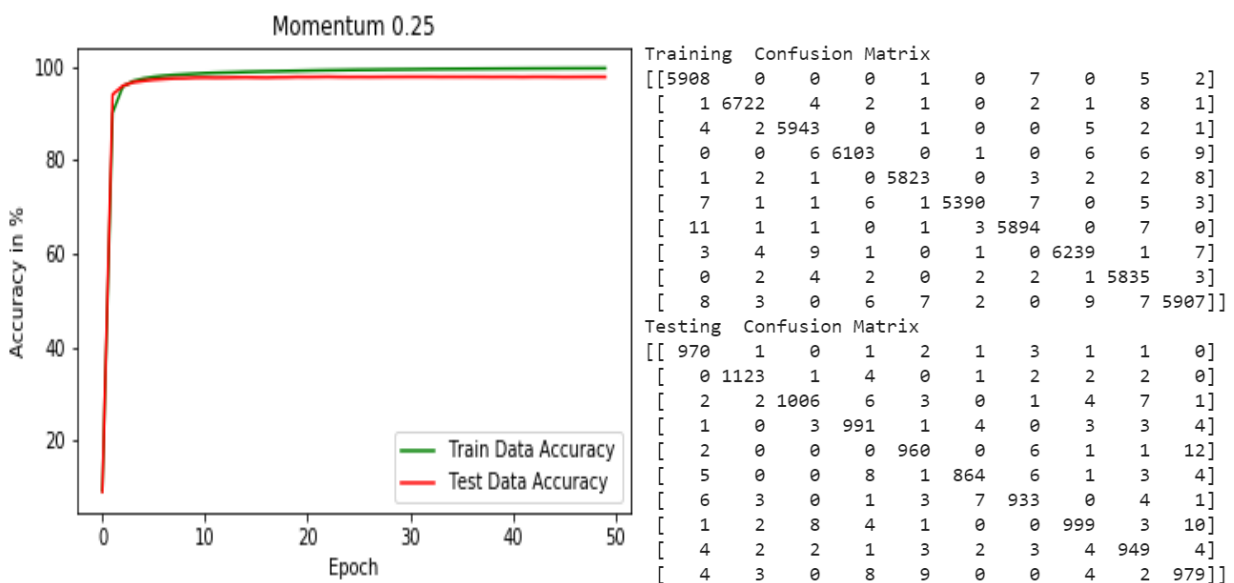
Accuracy Train Data [9.22333333333333, 89.01333333333334, 95.10499999999999, 96.34333333333333, 97.04333333333334, 97.44333333333334, 97.745, 97.97666666666667, 98.15333333333334, 98.30333333333333, 98.41833333333334, 98.54166666666667, 98.61333333333333, 98.70666666666666, 98.78166666666667, 98.85000000000001, 98.905, 98.965, 99.01833333333333, 99.06166666666667, 99.09666666666666, 99.125, 99.15, 99.18, 99.21,

99.24166666666666, 99.26, 99.28166666666667, 99.30499999999999, 99.32166666666666,  
 99.33333333333333, 99.35000000000001, 99.35833333333333, 99.37166666666667,  
 99.38833333333334, 99.40333333333334, 99.41666666666666, 99.42833333333333,  
 99.44833333333334, 99.46166666666667, 99.48166666666667, 99.49, 99.50166666666667,  
 99.51333333333334, 99.52333333333333, 99.535, 99.54166666666666, 99.55833333333334,  
 99.56333333333333, 99.575]

Accuracy Test Data [8.97, 93.41000000000001, 95.24000000000001, 96.07, 96.52, 96.74000000000001,  
 96.88, 96.98, 97.02, 97.11999999999999, 97.25, 97.32, 97.39, 97.43, 97.49, 97.52, 97.55, 97.61, 97.64,  
 97.66, 97.65, 97.67, 97.7, 97.72, 97.72, 97.72999999999999, 97.72, 97.7, 97.72, 97.71, 97.72, 97.71,  
 97.71, 97.7, 97.69, 97.7, 97.68, 97.66, 97.67, 97.67, 97.67, 97.65, 97.66, 97.64, 97.66, 97.65, 97.64, 97.64,  
 97.65, 97.64]

Convergence takes place after 20 epochs.

### Momentum:0.25



Accuracy Train Data [9.335, 90.115, 95.69833333333332, 96.83, 97.37166666666667,  
 97.73833333333334, 97.985, 98.16166666666668, 98.28166666666667, 98.41, 98.53166666666667,  
 98.60833333333333, 98.67, 98.72999999999999, 98.81, 98.86, 98.91166666666666,  
 98.94666666666667, 99.0, 99.045, 99.08833333333334, 99.13333333333333, 99.17,  
 99.19666666666667, 99.22833333333332, 99.26, 99.27666666666667, 99.30499999999999, 99.33,  
 99.34833333333334, 99.36833333333334, 99.39, 99.41666666666666, 99.42833333333333, 99.445,  
 99.45666666666668, 99.47333333333333, 99.485, 99.49666666666667, 99.51, 99.52,  
 99.53166666666667, 99.54499999999999, 99.55166666666668, 99.56, 99.57333333333334,  
 99.58166666666666, 99.58833333333334, 99.59666666666666, 99.60666666666667]

Accuracy Test Data [9.11, 94.0, 95.84, 96.6, 96.89999999999999, 97.16, 97.31, 97.44, 97.47,  
 97.57000000000001, 97.6, 97.57000000000001, 97.59, 97.63, 97.65, 97.61999999999999, 97.6, 97.65,  
 97.67, 97.72999999999999, 97.72, 97.74000000000001, 97.77, 97.72999999999999, 97.72,

97.72999999999999, 97.72, 97.72999999999999, 97.75, 97.75, 97.76, 97.76, 97.76, 97.74000000000001, 97.75, 97.74000000000001, 97.75, 97.74000000000001, 97.72999999999999, 97.74000000000001, 97.74000000000001, 97.72, 97.75, 97.75, 97.74000000000001, 97.72, 97.72999999999999, 97.74000000000001, 97.75, 97.74000000000001]

Convergence takes place after 11 epochs.

## Momentum 0.5



Accuracy of Train data : [9.931666666666667, 91.42, 96.21666666666667, 97.19833333333334, 97.67, 97.96166666666667, 98.17666666666666, 98.35166666666667, 98.48666666666666, 98.595, 98.67666666666666, 98.74166666666667, 98.81333333333333, 98.89166666666667, 98.94833333333334, 99.00999999999999, 99.04666666666667, 99.08833333333334, 99.13666666666666, 99.18333333333334, 99.225, 99.24, 99.27, 99.29666666666667, 99.32166666666666, 99.35000000000001, 99.37166666666667, 99.39, 99.41166666666666, 99.42666666666666, 99.45166666666667, 99.46833333333333, 99.47666666666667, 99.49166666666667, 99.505, 99.51333333333334, 99.51833333333333, 99.53166666666667, 99.54666666666667, 99.55833333333334, 99.57166666666667, 99.57833333333333, 99.59, 99.59833333333333, 99.60666666666667, 99.61333333333333, 99.62, 99.625, 99.63333333333333, 99.65166666666667]

Accuracy of Test data : [10.11, 94.75, 96.28999999999999, 96.84, 96.95, 97.11999999999999, 97.23, 97.22, 97.32, 97.41, 97.43, 97.46000000000001, 97.47, 97.52, 97.50999999999999, 97.55, 97.54, 97.53, 97.5, 97.53, 97.52, 97.50999999999999, 97.50999999999999, 97.50999999999999, 97.52, 97.53, 97.52, 97.50999999999999, 97.55, 97.6, 97.61999999999999, 97.61999999999999, 97.65, 97.66, 97.63, 97.61999999999999, 97.59, 97.58, 97.56, 97.57000000000001, 97.58, 97.59, 97.58, 97.56, 97.57000000000001, 97.57000000000001, 97.55, 97.54, 97.54, 97.57000000000001]

Convergence takes place after 9 epochs.

Momentum	Training accuracy in %	Testing accuracy in %
0	99.57	97.64
0.25	99.6	97.74
0.5	99.65	97.57
0.9	99.59	97.32

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

With increase in momentum, convergence happen quicker. But the momentum value does not have much effect on accuracy of the test data. For momentum value of 0,0.25,0.5 and 0.9 the final accuracy after 50 epochs is 97%

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

With increase in momentum we can see that the number of epochs needed to converge has reduced slightly. For 0 momentum the number of epochs need to converge is 20, for 0.25 it is 11 and for 0.5 it is 9 and for 0.9 it is 8.

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

For momentum 0 I have observed that the test data and train data underfit since the graphs are almost overlapping. For momentum 0.25,0.5 and 0.9 I have observed that the train data is overfitting since after reaching a certain point the test data accuracy is decreasing when compared to train data accuracy.

**Experiment 3: Vary the number of training examples**

Constant parameters:

Momentum=0.9

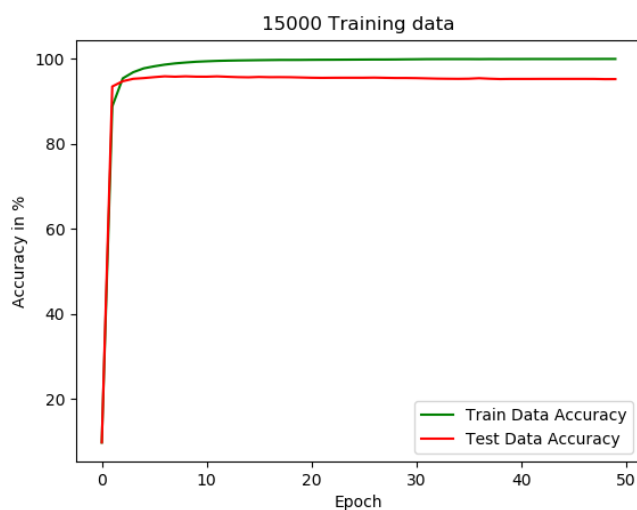
Learning rate=0.1

Number of epochs=50

Hidden Units=100

**Using one quarter of 60000 training data which is 15000.**

15000 training data was chosen randomly such that it was well distributed.



Training Confusion Matrix

```
[[1495  0  0  0  0  0  1  0  0  0]
 [  0 1688  0  1  0  0  0  0  1  0]
 [  0  0 1462  0  0  0  0  0  0  0]
 [  0  0  0 1546  0  0  0  1  0  1]
 [  0  0  0  0 1468  0  0  0  0  0]
 [  0  0  1  1  0 1316  0  0  0  0]
 [  1  0  0  0  0  0 1489  0  0  0]
 [  0  0  0  1  0  0  0 1592  0  0]
 [  0  0  0  0  0  0  0  0 1432  0]
 [  0  0  0  0  0  0  0  0  0 1501]]
```

Testing Confusion Matrix

```
[[ 964  0  1  0  1  1  3  1  9  0]
 [  1 1115  2  1  0  1  4  2  9  0]
 [  6  0 980  5  1  0  2  9 25  4]
 [  2  0 13 956  0  7  0  7 21  4]
 [  1  3  7  0 908  0 11  0  5 47]
 [  8  0  3 37  4 797  8  2 30  3]
 [ 11  3  3  1  8  8 916  0  8  0]
 [  0  4 12  5  2  0  1 987  4 13]
 [  5  0  4  1  2  3  6  5 941  7]
 [  4  3  2  4  7  4  2 10 18 955]]
```

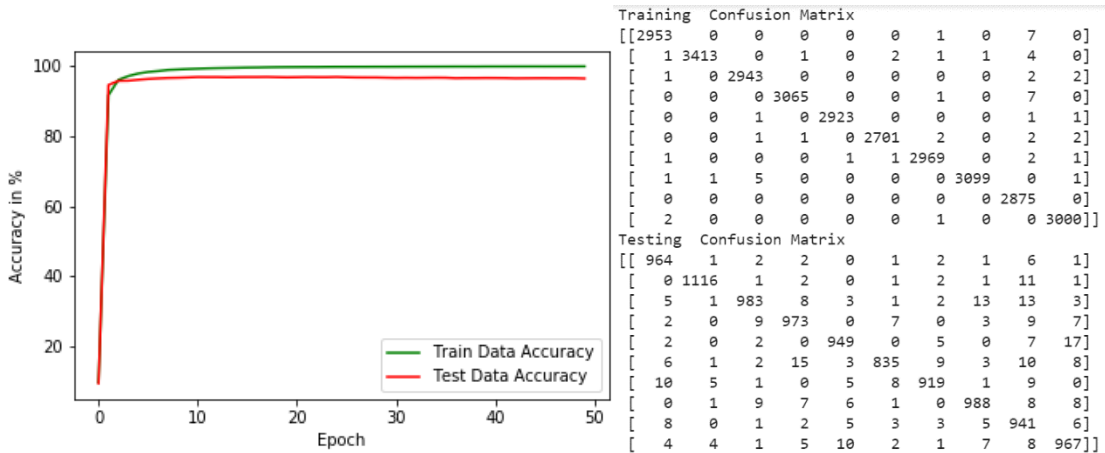
Accuracy of Train data : [9.786666666666667, 88.78, 95.36, 96.80666666666666, 97.71333333333332, 98.19333333333333, 98.58666666666667, 98.88, 99.08666666666667, 99.26666666666667, 99.37333333333333, 99.46666666666667, 99.52666666666666, 99.57333333333334, 99.60666666666667, 99.64, 99.66666666666667, 99.69333333333333, 99.69333333333333, 99.7, 99.72, 99.73333333333333, 99.74, 99.75333333333334, 99.76666666666667, 99.78, 99.79333333333334, 99.79333333333334, 99.80666666666667, 99.83333333333333, 99.85333333333334, 99.88, 99.88666666666667, 99.89333333333333, 99.89333333333333, 99.89333333333333, 99.88, 99.9, 99.89333333333333, 99.9, 99.9, 99.90666666666667, 99.90666666666667, 99.90666666666667, 99.90666666666667, 99.91333333333333, 99.92, 99.92, 99.92666666666666, 99.92666666666666]

Accuracy of Test data : [9.82, 93.45, 94.69, 95.26, 95.43, 95.65, 95.84, 95.75, 95.84, 95.76, 95.75, 95.83, 95.73, 95.65, 95.61, 95.69, 95.64, 95.65, 95.63000000000001, 95.57, 95.50999999999999, 95.48, 95.5, 95.50999999999999, 95.50999999999999, 95.50999999999999, 95.54, 95.49, 95.45, 95.45, 95.42, 95.36, 95.3, 95.27, 95.25, 95.28, 95.39999999999999, 95.28, 95.19999999999999, 95.23, 95.23, 95.23, 95.24000000000001, 95.24000000000001, 95.24000000000001, 95.24000000000001, 95.24000000000001, 95.23, 95.17999999999999, 95.19]

### **Using one quarter of 60000 training data which is 30000.**

30000 training data was chosen randomly such that it was well distributed.





Accuracy of Train data : [9.916666666666666, 91.60000000000001, 95.95333333333333, 97.03666666666668, 97.76666666666667, 98.20666666666666, 98.46333333333334, 98.78, 98.91, 99.03999999999999, 99.14333333333335, 99.24, 99.30333333333333, 99.35333333333334, 99.4, 99.45666666666668, 99.48333333333333, 99.53666666666666, 99.57666666666667, 99.6, 99.61333333333333, 99.63, 99.62666666666667, 99.65666666666667, 99.67, 99.67333333333333, 99.68, 99.68333333333334, 99.69666666666667, 99.69333333333333, 99.70666666666666, 99.71666666666667, 99.72, 99.72999999999999, 99.73333333333333, 99.74, 99.75, 99.75333333333334, 99.74666666666666, 99.76666666666667, 99.77000000000001, 99.77000000000001, 99.77666666666667, 99.78, 99.78333333333333, 99.78, 99.78333333333333, 99.78666666666666, 99.79, 99.80333333333333]

Accuracy of Test data : [9.58, 94.5, 95.67, 95.71, 95.99, 96.22, 96.37, 96.5, 96.56, 96.65, 96.77, 96.75, 96.76, 96.7, 96.76, 96.77, 96.77, 96.8, 96.73, 96.67999999999999, 96.72, 96.77, 96.73, 96.73, 96.78999999999999, 96.7, 96.65, 96.64, 96.64, 96.58, 96.53, 96.56, 96.53, 96.55, 96.58, 96.56, 96.44, 96.48, 96.47, 96.5, 96.49, 96.47, 96.39999999999999, 96.41999999999999, 96.44, 96.45, 96.41999999999999, 96.44, 96.43, 96.35000000000001]

Train Data Samples	Train Data Accuracy	Test Data Accuracy
15000	99.92	95.19
30000	99.79	96.35

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

By taking one quarter and one half of the training samples we see that the test data accuracy reduces.

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

The number of epochs needed to converge increases by reducing the size of the training data.

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

Yes, I do observe overfitting. In both the graphs the test data accuracy goes below train data accuracy after a certain point.

**How to run:**

The code was run on Google Collaboratory. The train and test data were loaded from the csv file through the lines in the main function of the program.

```
train=np.genfromtxt('/comtent/mnist_train.csv',delimiter=",")
```

```
test=np.genfromtxt('/content/mnist_test.csv',delimiter=",")
```