CSE4/574 Introduction to Machine Learning

Programming Assignment 2

Handwritten Digits Classification

Team Members:

1. Janhavi Desale (50412898)
2. Krati Sharma (50416821)
3. Pinaz Shaikh (50413528)

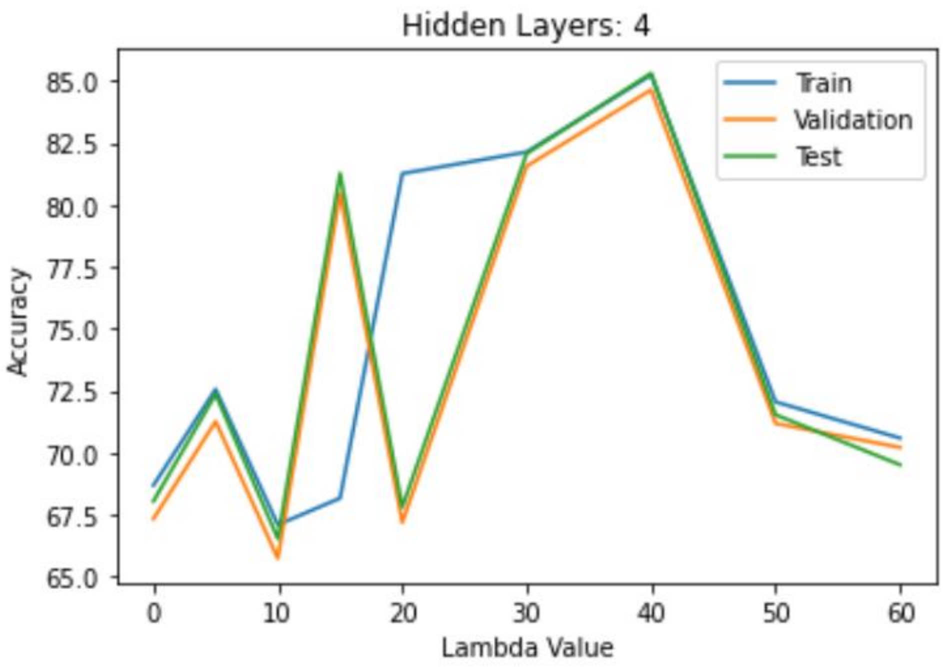
In this assignment, our task was to implement a Multilayer Perceptron Neural Network and evaluate its performance in classifying handwritten digits. We then used the same network to analyze a more challenging face dataset and compare the performance of the neural network against a deep neural network and a convolutional neural network using the TensorFlow library.

Choosing hyper-parameter for Neural Network:

In a neural network we are using three hyper-parameters, λ (regularization parameter), number of hidden nodes, and number of hidden layers. We have regularized the neural network by running the script for different values of λ and number of hidden nodes. We can see from the below table that we attain the maximum validation accuracy when the lambda value is 5 and the number of hidden nodes is 100. The model gives a validation accuracy of 98.39%, training set accuracy of 97.13% and test set accuracy of 97.25%.

Hidden Nodes	**Lambda**	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
4	0	68.67%	67.32%	68.03%
4	5	72.55%	71.25%	72.35%
4	10	67.08%	65.72%	66.52%
4	15	81.26%	80.47%	81.27%
4	20	68.15%	67.17%	67.79%
4	30	82.13%	81.56%	82.07%
4	40	85.24%	84.64%	85.31%
4	50	72.04%	71.17%	71.52%
4	60	70.58%	70.20%	69.50%

Table 1.1: Various lambda values for 4 hidden nodes

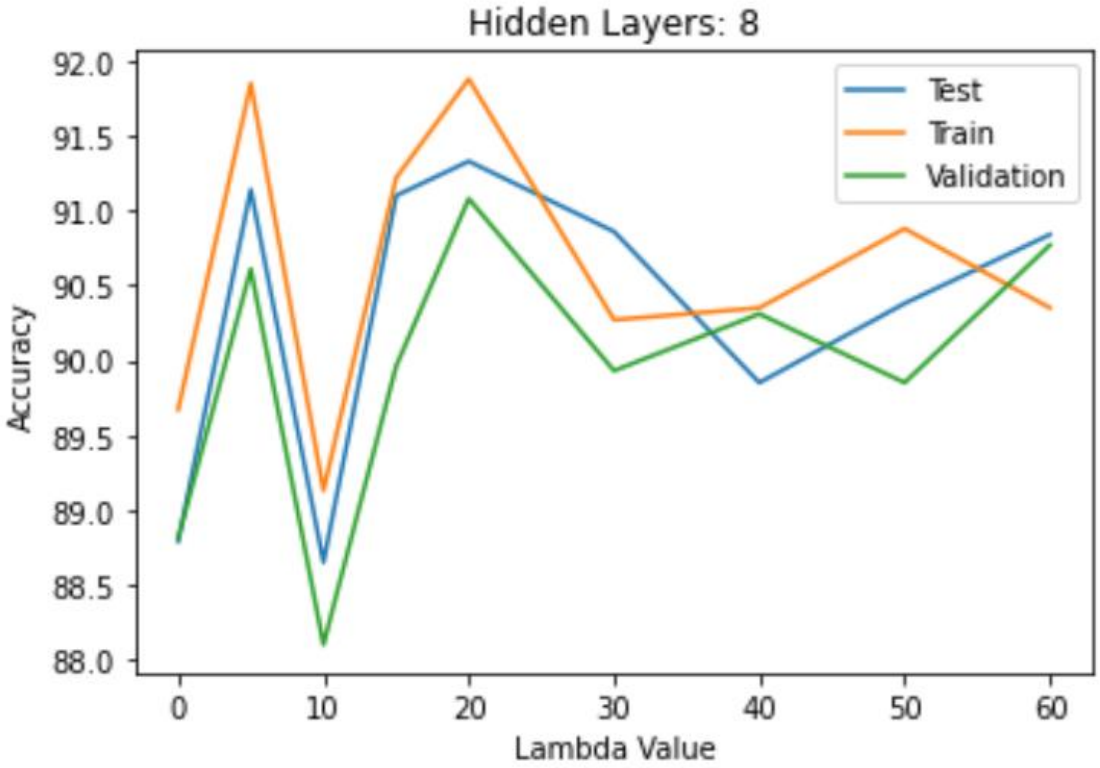


From the above table we can deduce that we achieved the maximum validation accuracy when the lambda value is 10 and the number of hidden nodes is 50. The model takes 0.04859 seconds to train on the MNIST data set and gives a validation accuracy of 94.8%, training set accuracy of 95.406% and test set accuracy of 94.78%.

We plot the graphs for various lambda(λ) values against the accuracy of the Neural Network for each value of the hidden layer, to study the relation between lambda and the performance of our neural network.

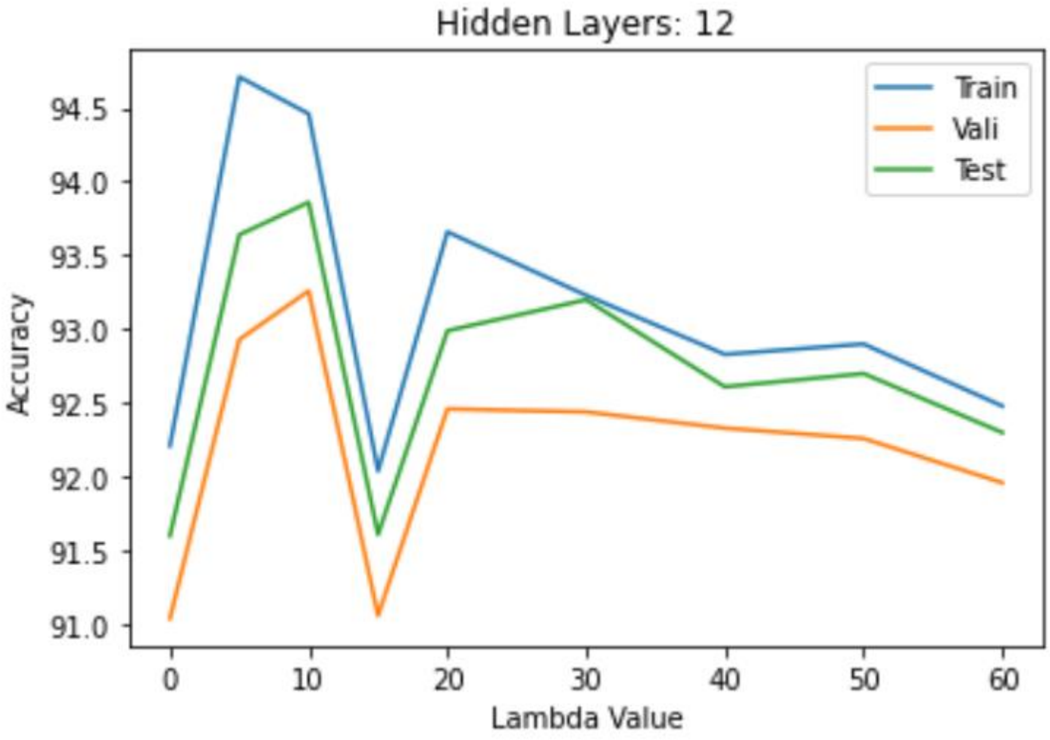
Hidden Nodes	**Lambda**	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
8	0	89.67%	88.81%	88.79%
8	5	91.85%	90.61%	91.14%
8	10	89.13%	88.10%	88.65%
8	15	91.22%	89.96%	91.10%
8	20	91.88%	91.08%	91.33%
8	30	90.27%	89.93%	89.85%
8	40	90.88%	90.31%	90.86%
8	50	90.35%	89.85%	90.38%
8	60	90.84%	90.35%	90.77%

Table 1.2: Various Lambda values for 8 hidden nodes



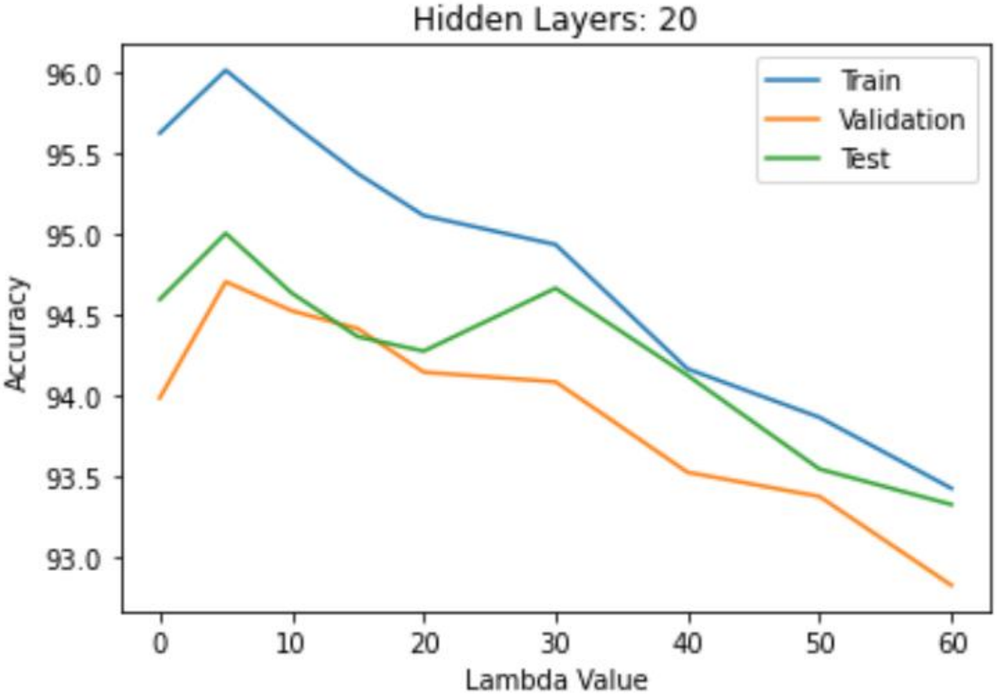
Hidden Nodes	Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy
12	0	92.21%	91.04%	91.69%
12	5	94.71%	92.93%	93.64%
12	10	94.46%	93.26%	93.86%
12	15	92.04%	91.06%	91.61%
12	20	93.66%	92.46%	92.99%
12	30	93.23%	92.44%	93.20%
12	40	92.83%	92.33%	92.61%
12	50	92.90%	92.26%	92.70%
12	60	92.48%	91.96%	92.30%

Table 1.3: Various Lambda values for 12 hidden nodes



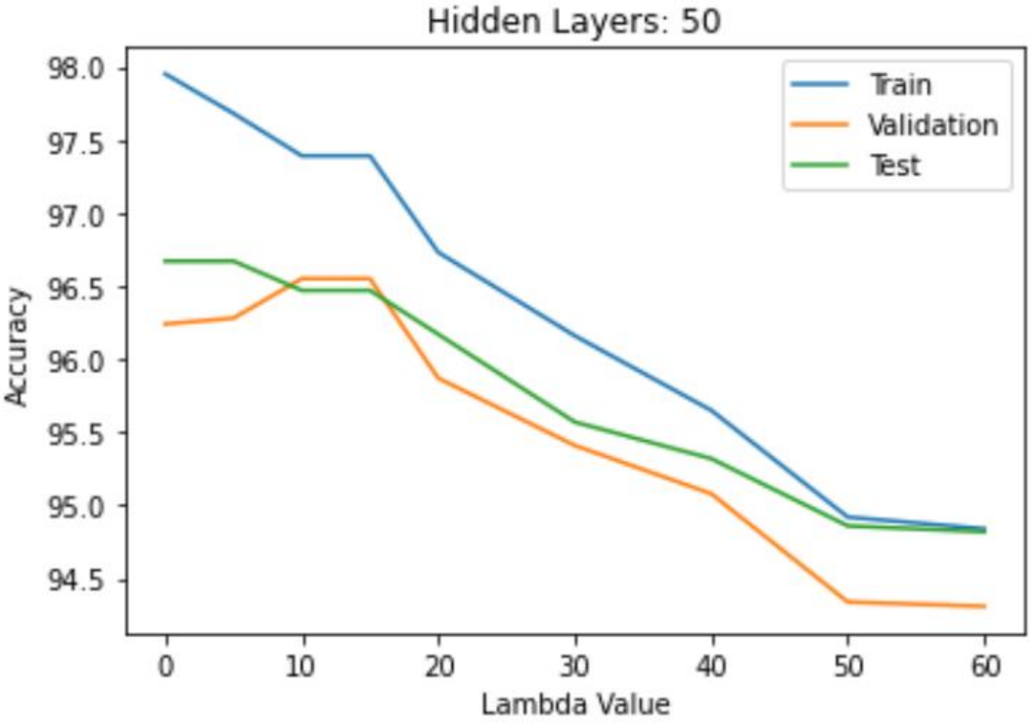
Hidden Nodes	Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy
20	0	95.62%	93.98%	94.59%
20	5	96.01%	94.70%	95.00%
20	10	95.68%	94.41%	94.63%
20	15	95.37%	94.52%	94.36%
20	20	95.11%	94.14%	94.27%
20	30	94.93%	94.08%	94.66%
20	40	94.16%	93.52%	94.12%
20	50	93.86%	93.37%	93.54%
20	60	93.42%	92.82%	93.32%

Table 1.4: Various Lambda values for 20 hidden nodes



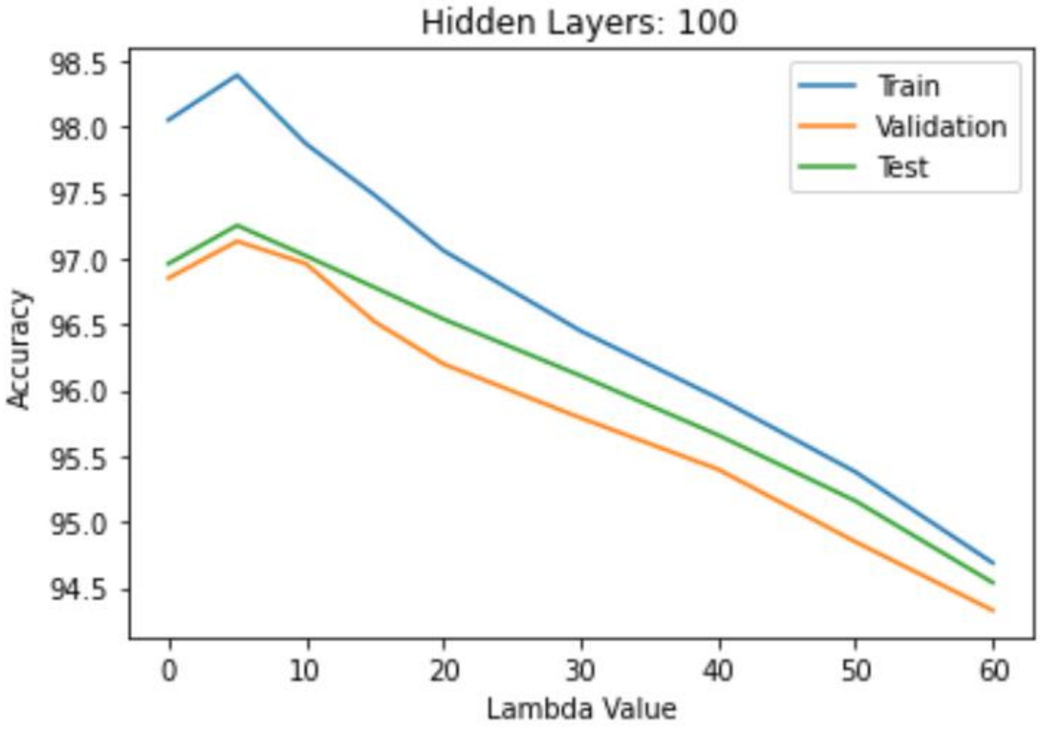
Hidden Nodes	Lambda	Training Accuracy	Validation Accuracy	Testing Accuracy
50	0	97.95	96.24	96.67
50	5	97.68	96.28	96.67
50	10	97.39	96.55	96.47
50	15	97.39	96.55	96.47
50	20	96.73	95.87	96.17
50	30	96.16	95.41	95.57
50	40	95.65	95.08	95.32
50	50	94.92	94.34	94.86
50	60	94.84	94.31	94.82

Table 1.5: Various Lambda values for 50 hidden nodes

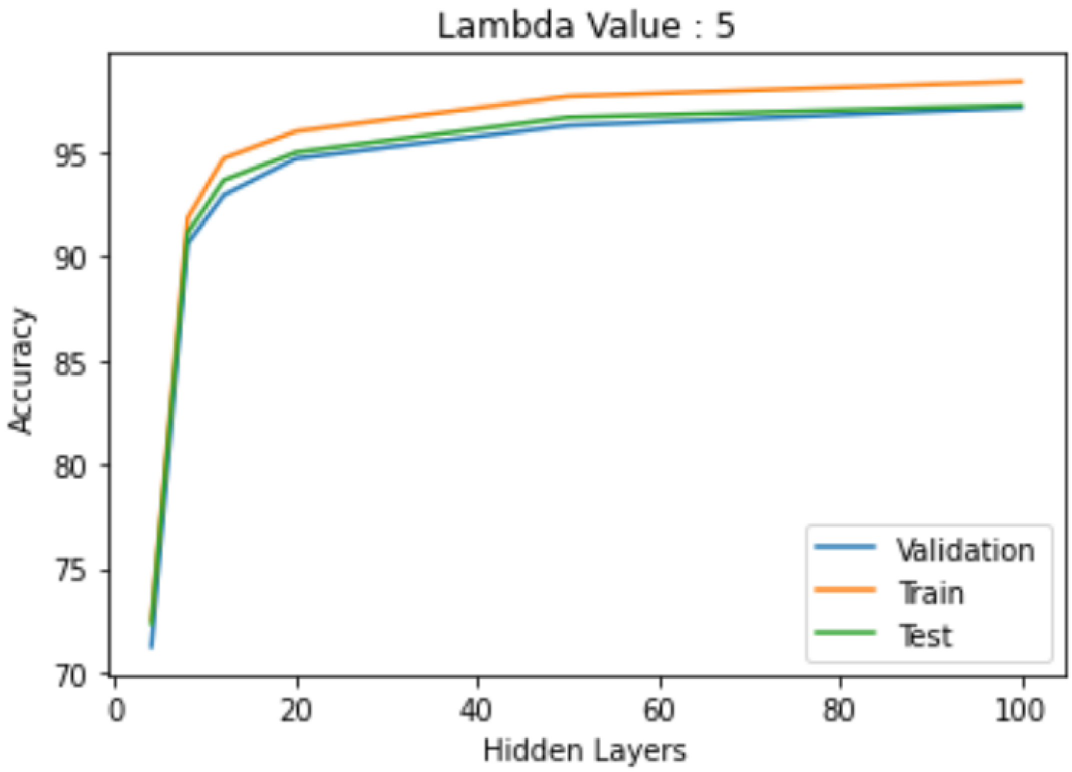


Hidden Nodes	**Lambda**	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
100	0	98.05%	96.85%	96.96%
100	5	98.39%	97.13%	97.25%
100	10	97.87%	96.96%	97.02%
100	15	97.48%	96.52%	96.78%
100	20	97.06%	96.20%	96.54%
100	30	96.45%	95.79%	96.11%
100	40	95.94%	95.40%	95.66%
100	50	95.38%	94.85%	95.16%
100	60	94.69%	94.33%	94.54%

Table 1.6: Various Lambda values for 100 hidden nodes



Inference: From the above graphs we can observe that the highest value of validation accuracy for hidden layer nodes and lambda takes the values 100 and 5 respectively. We achieve an accuracy of 97.13% on the validation set. We also plot the graph for the hidden layer nodes with respect to accuracy for lambda value 5 to understand the performance of the neural network with respect to the number of hidden layer nodes.

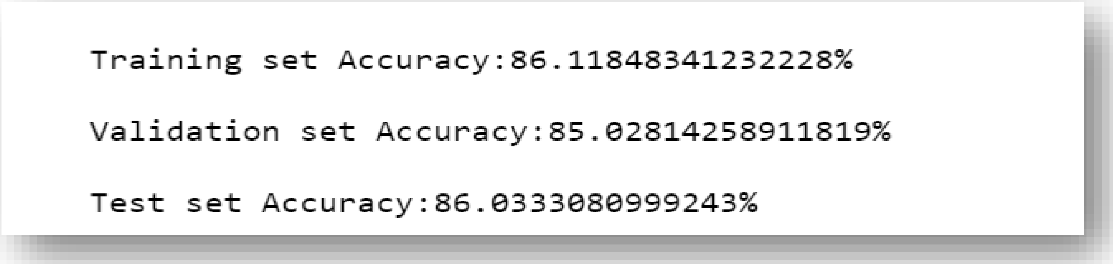


Hence, we can observe that the performance of the neural network gets better as we add more hidden layer nodes and it performs the best at nodes=100.

Accuracy of classification method on the CelebA dataset:

We are using the TensorFlow library to evaluate the accuracy of the single hidden layer Neural Network on CelebA dataset to distinguish between two classes - wearing glasses and not wearing glasses.

Below are the results for test, training and validation accuracy:



```
Training set Accuracy:86.11848341232228%

Validation set Accuracy:85.02814258911819%

Test set Accuracy:86.0333080999243%
```

Lambda Value	Training Set Accuracy	Validation Set	Test Set Accuracy
0	88.00	87.20	87.77
5	88.45	87.91	88.22
10	88.83	87.99	88.79
15	88.18	87.27	88.07
20	88.33	88.14	88.15
25	88.50	87.80	88.30
30	89.02	88.96	88.79
35	88.56	87.72	88.15
40	87.04	86.41	86.90
45	87.96	87.09	87.88
50	87.03	87.42	87.50
55	88.79	88.51	88.79
60	87.80	87.20	87.58

Comparison of your neural network with a deep neural network (using TensorFlow) in terms of accuracy and training time

In this section we analyze and compare the results from single layer neural network and deep neural network with multiple hidden layers. The goal here is to distinguish between two classes, *glasses* and *no glasses*. Using a single layer neural network, we get an accuracy of 81.65% on the validation set. The model takes 177.38 seconds to train data. We also tried to predict the same using deep neural networks with various hidden layers(2, 3, 5, 7) so as to achieve more accuracy. However, we can notice from the below table that as the number of hidden layers increases our model accuracy drops.

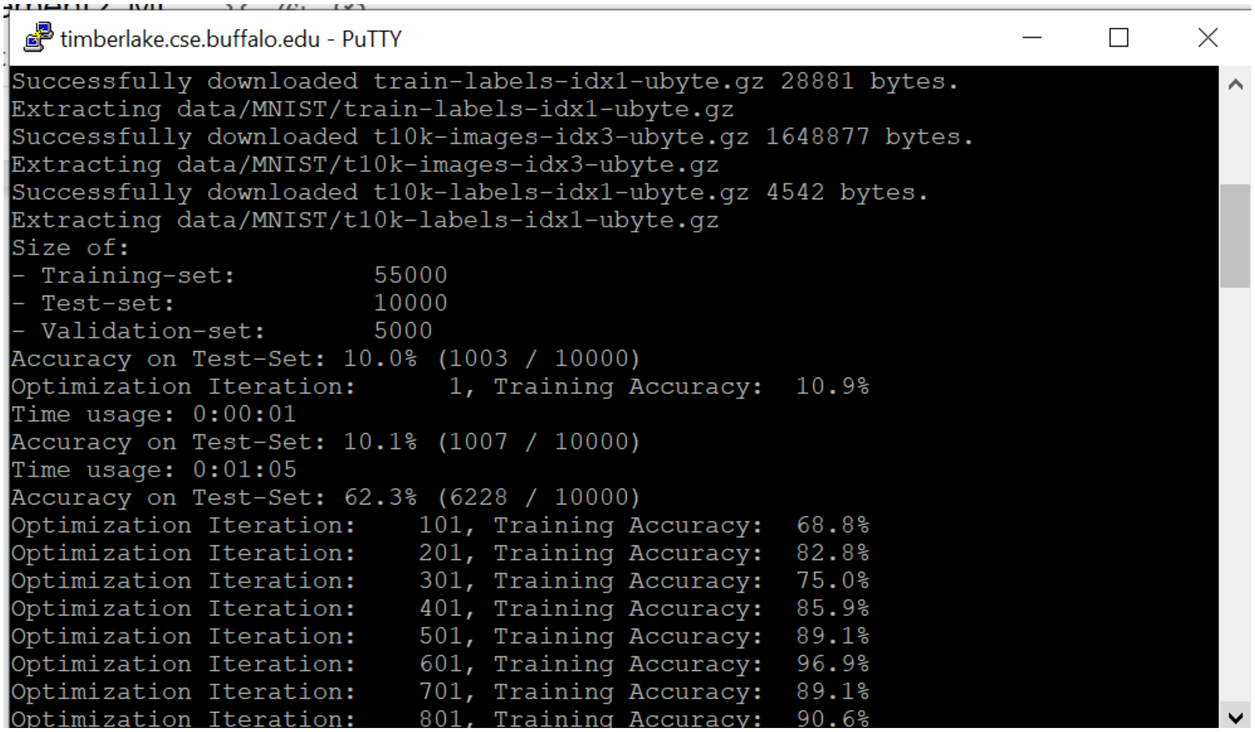
Number of Hidden Layers	**Number of Units in each Layer**	**Time taken for Training**	**Validation Accuracy**
2	50	77.91	81.65%
3	256	88.43	79.62%
3	512	91.43	79.06%
3	1024	93.24	81.31%
5	256	119.34	76.34%
5	512	122.86	78.01%
5	1024	122.86	80.33%
7	256	126.3	71.72%
7	512	137.89	75.12%
7	1024	178.64	76.92%

Hence, we can say that adding more layers may increase or decrease the accuracy depending upon the complexity of the problem. As the problem statement is not relatively complex, we can notice that we get the best accuracy from the single layer neural network. Therefore, from the above observations we can interpret that this might be the optimal number of neural networks for this problem.

Results from convolutional neural network in terms of accuracy and training time:

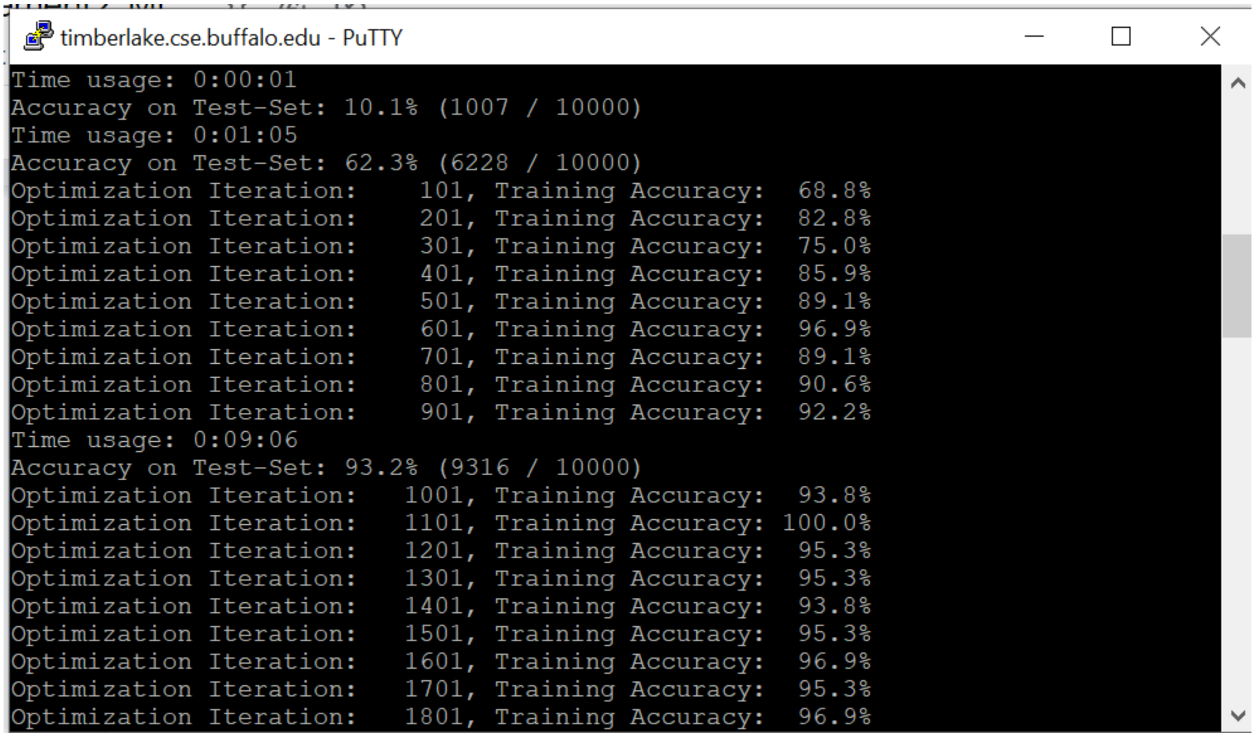
When we run the Convolutional Neural Network on MNIST dataset, at the end of 9,000 iterations, we get a test set accuracy of 96.9%, which is higher than our single layer neural networks.

With the first 100 iterations, we get an accuracy of 68.8%. We further increase iterations to see if accuracy keeps on growing.



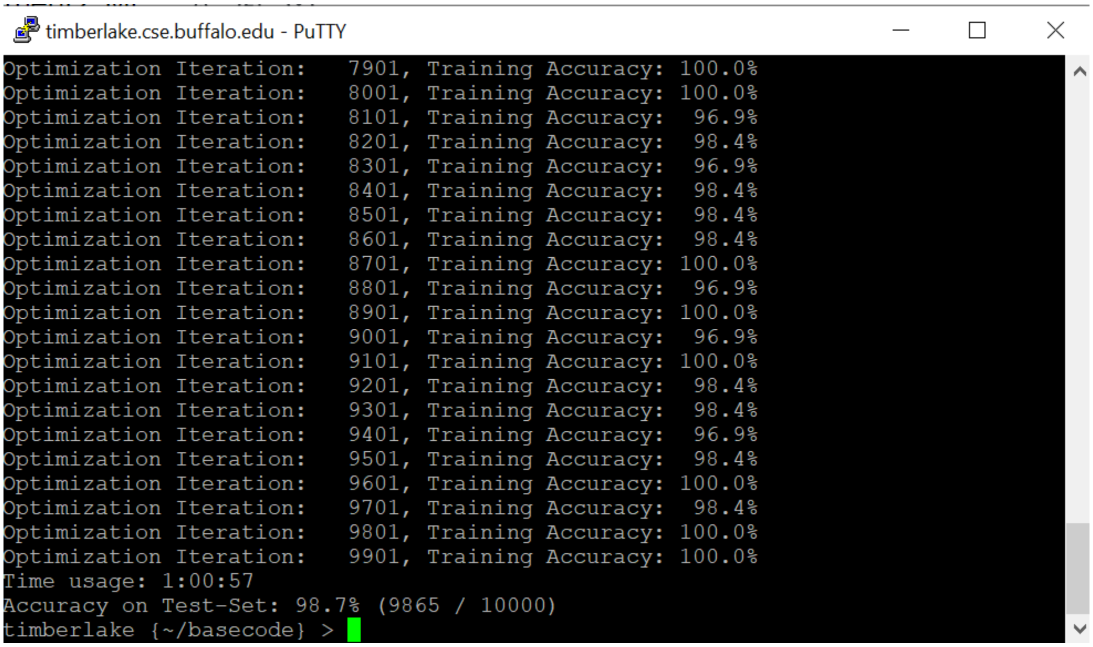
```
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
Extracting data/MNIST/train-labels-idx1-ubyte.gz
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting data/MNIST/t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting data/MNIST/t10k-labels-idx1-ubyte.gz
Size of:
- Training-set:         55000
- Test-set:             10000
- Validation-set:       5000
Accuracy on Test-Set: 10.0% (1003 / 10000)
Optimization Iteration:      1, Training Accuracy:  10.9%
Time usage: 0:00:01
Accuracy on Test-Set: 10.1% (1007 / 10000)
Time usage: 0:01:05
Accuracy on Test-Set: 62.3% (6228 / 10000)
Optimization Iteration:    101, Training Accuracy:  68.8%
Optimization Iteration:    201, Training Accuracy:  82.8%
Optimization Iteration:    301, Training Accuracy:  75.0%
Optimization Iteration:    401, Training Accuracy:  85.9%
Optimization Iteration:    501, Training Accuracy:  89.1%
Optimization Iteration:    601, Training Accuracy:  96.9%
Optimization Iteration:    701, Training Accuracy:  89.1%
Optimization Iteration:    801, Training Accuracy:  90.6%
```

With the first 900 iterations, we get a test accuracy of 92.2 %. Also, for the training set we get an accuracy of 100%. Further, we look at the model after it has been optimized over 9000 iterations.



```
Time usage: 0:00:01
Accuracy on Test-Set: 10.1% (1007 / 10000)
Time usage: 0:01:05
Accuracy on Test-Set: 62.3% (6228 / 10000)
Optimization Iteration:    101, Training Accuracy:  68.8%
Optimization Iteration:    201, Training Accuracy:  82.8%
Optimization Iteration:    301, Training Accuracy:  75.0%
Optimization Iteration:    401, Training Accuracy:  85.9%
Optimization Iteration:    501, Training Accuracy:  89.1%
Optimization Iteration:    601, Training Accuracy:  96.9%
Optimization Iteration:    701, Training Accuracy:  89.1%
Optimization Iteration:    801, Training Accuracy:  90.6%
Optimization Iteration:    901, Training Accuracy:  92.2%
Time usage: 0:09:06
Accuracy on Test-Set: 93.2% (9316 / 10000)
Optimization Iteration:   1001, Training Accuracy:  93.8%
Optimization Iteration:   1101, Training Accuracy: 100.0%
Optimization Iteration:   1201, Training Accuracy:  95.3%
Optimization Iteration:   1301, Training Accuracy:  95.3%
Optimization Iteration:   1401, Training Accuracy:  93.8%
Optimization Iteration:   1501, Training Accuracy:  95.3%
Optimization Iteration:   1601, Training Accuracy:  96.9%
Optimization Iteration:   1701, Training Accuracy:  95.3%
Optimization Iteration:   1801, Training Accuracy:  96.9%
```

At the end of 9,000 iteration, we have a test-set accuracy of 96.9%, which is higher than the accuracies obtained from any of the other models that we ran on the handwriting dataset.



```
Optimization Iteration:   7901, Training Accuracy: 100.0%
Optimization Iteration:   8001, Training Accuracy: 100.0%
Optimization Iteration:   8101, Training Accuracy:  96.9%
Optimization Iteration:   8201, Training Accuracy:  98.4%
Optimization Iteration:   8301, Training Accuracy:  96.9%
Optimization Iteration:   8401, Training Accuracy:  98.4%
Optimization Iteration:   8501, Training Accuracy:  98.4%
Optimization Iteration:   8601, Training Accuracy:  98.4%
Optimization Iteration:   8701, Training Accuracy: 100.0%
Optimization Iteration:   8801, Training Accuracy:  96.9%
Optimization Iteration:   8901, Training Accuracy: 100.0%
Optimization Iteration:   9001, Training Accuracy:  96.9%
Optimization Iteration:   9101, Training Accuracy: 100.0%
Optimization Iteration:   9201, Training Accuracy:  98.4%
Optimization Iteration:   9301, Training Accuracy:  98.4%
Optimization Iteration:   9401, Training Accuracy:  96.9%
Optimization Iteration:   9501, Training Accuracy:  98.4%
Optimization Iteration:   9601, Training Accuracy: 100.0%
Optimization Iteration:   9701, Training Accuracy:  98.4%
Optimization Iteration:   9801, Training Accuracy: 100.0%
Optimization Iteration:   9901, Training Accuracy: 100.0%
Time usage: 1:00:57
Accuracy on Test-Set: 98.7% (9865 / 10000)
timberlake {~/basecode} >
```