Machine Learning (CSE574)
Programming Assignment 2
Handwritten Digits Classification
Project Group - 36

Shubham Gulati (sgulati3)

Mohammad Umair (m39)

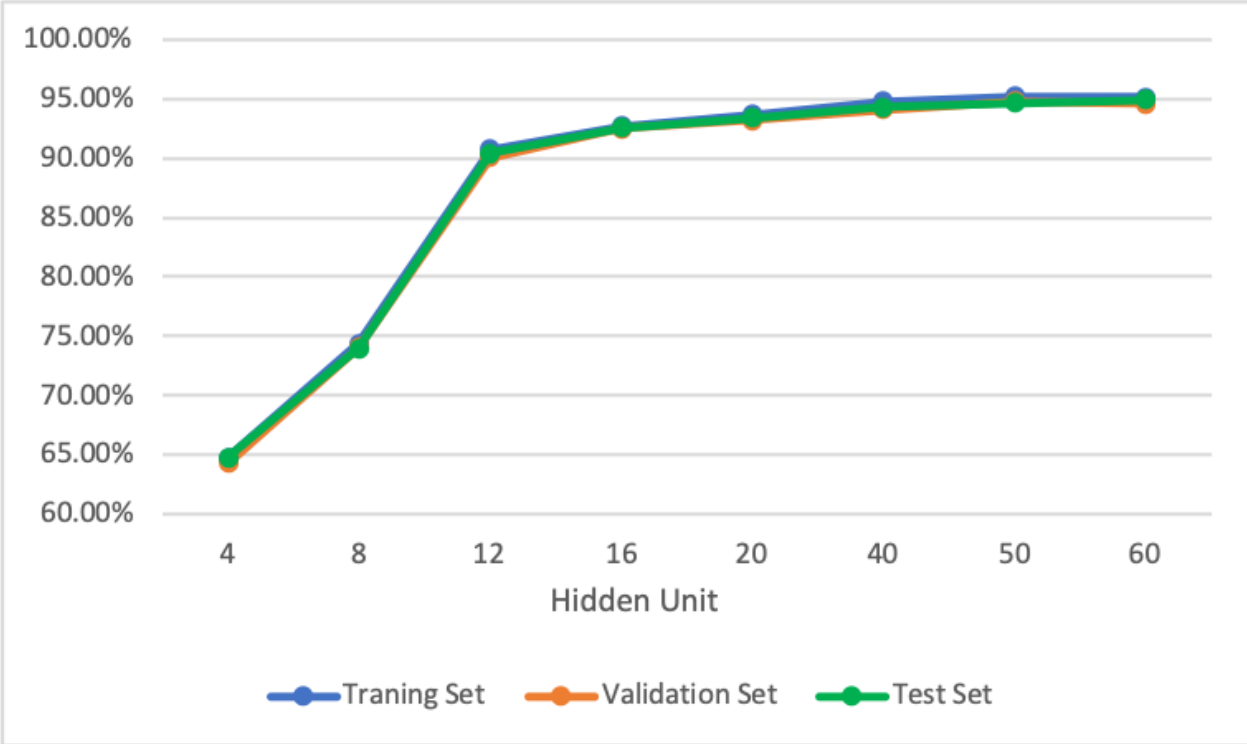
Yichen Wang (yichenwa)

PART 1 - How to choose the hyper-parameter for neural network with one hidden layer:

Task 1: Keeping the Lambda constant, finding the optimal number of hidden units.
Lambda = 0
Number of hidden units = 4,8,12,16,20,40,50,60

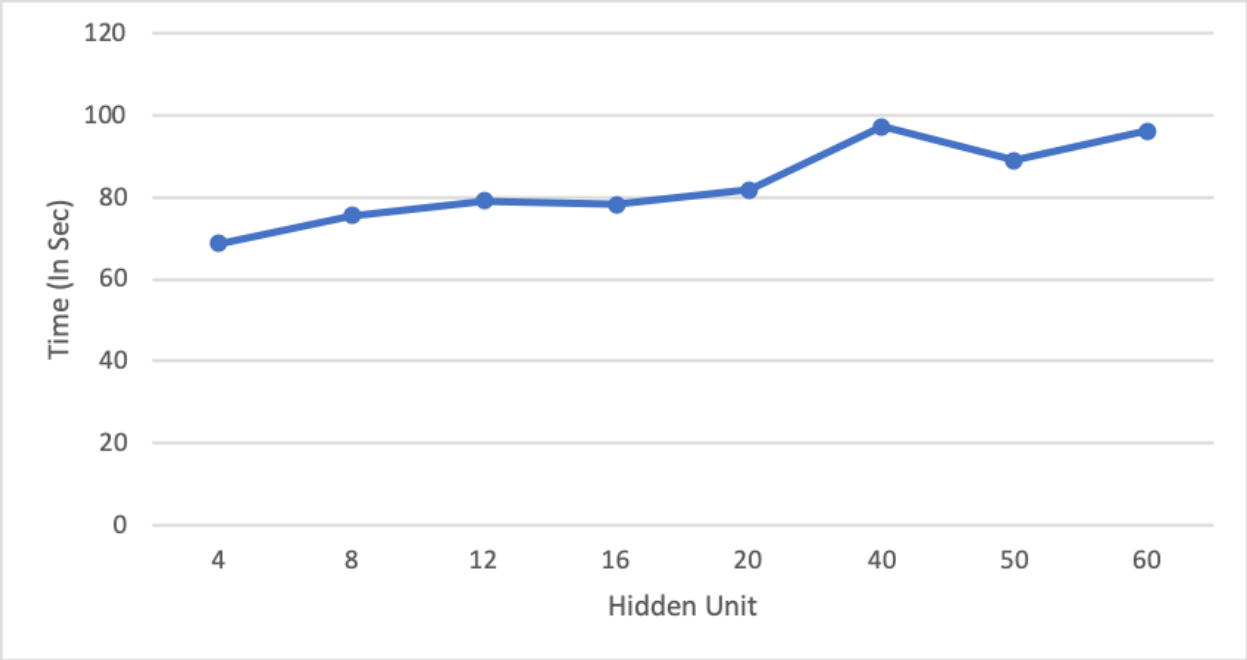
ACCURACY VARIANCE

Hidden Unit	Traning Set	Validation Set	Test Set
4	64.75%	64.28%	64.76%
8	74.37%	74.05%	74.00%
12	90.79%	90.07%	90.40%
16	92.73%	92.50%	92.55%
20	93.72%	93.24%	93.41%
40	94.84%	94.14%	94.35%
50	95.25%	94.82%	94.64%
60	95.18%	94.59%	94.93%



TIME VARIANCE

Hidden Unit	time
4	68.79319382
8	75.51845598
12	79.19816995
16	78.2404983
20	81.90873098
40	97.22336698
50	88.99004006
60	96.16838193



Conclusion: As is evident from the above graphs, our accuracy is maximum for 60 hidden units. However, we get lesser time for 50 hidden units whose accuracy is only a little lesser than the accuracy for 60 hidden units. Therefore, we will take 50 as our optimal number of hidden units.

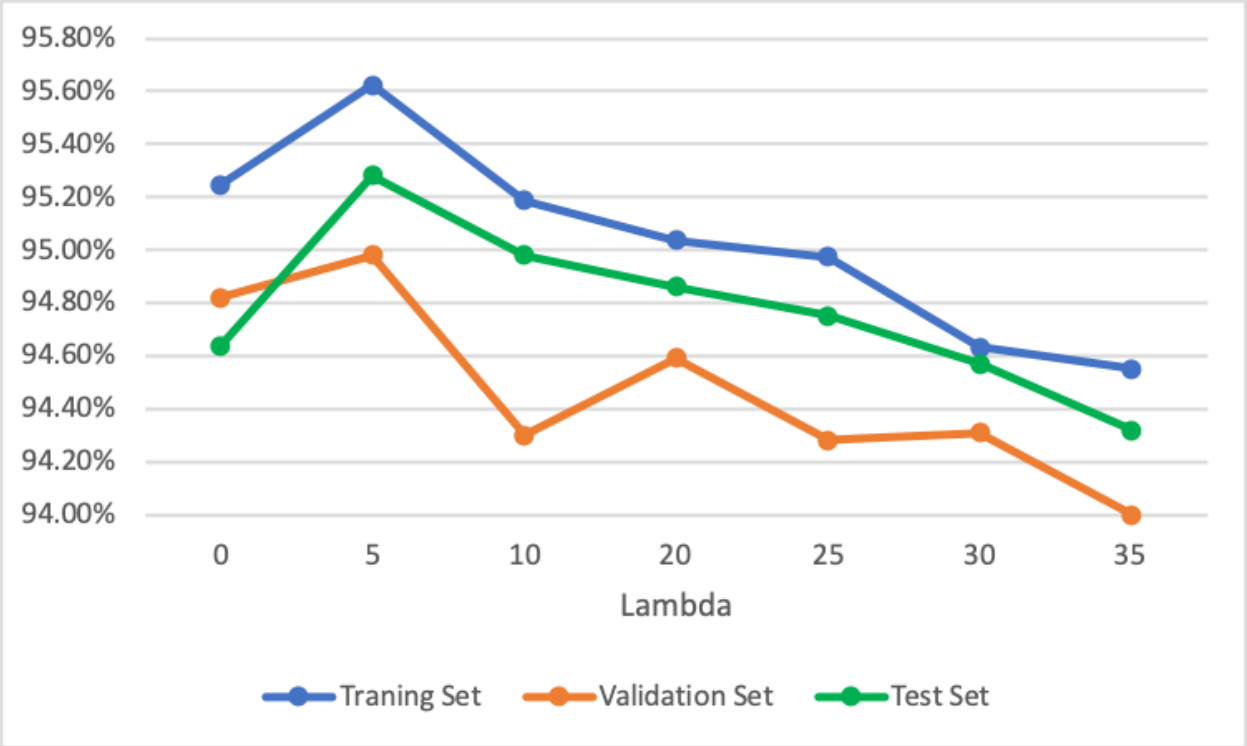
Task 2: Keeping the number of hidden units constant, finding the optimal value of Lambda.

Number of hidden units = 50

Lambda = 0,5,10,20,25,30,35

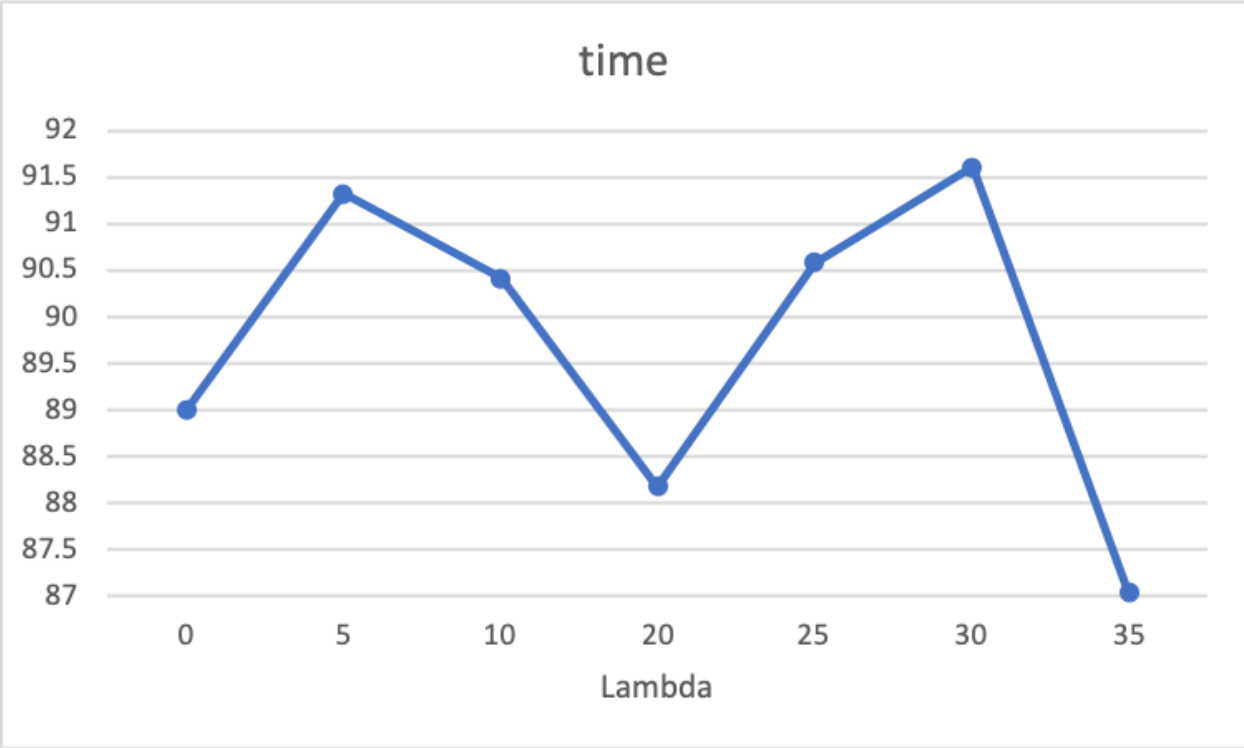
ACCURACY VARIANCE

Lambda	Traning Set	Validation Set	Test Set
0	95.25%	94.82%	94.64%
5	95.62%	94.98%	95.28%
10	95.19%	94.30%	94.98%
20	95.04%	94.59%	94.86%
25	94.97%	94.28%	94.75%
30	94.63%	94.31%	94.57%
35	94.55%	94.00%	94.32%



TIME VARIANCE

Lambda	time
0	88.99004006
5	91.31979012
10	90.40871286
20	88.16774392
25	90.57673192
30	91.60922194
35	87.03225422



Conclusion: As can be seen in the above Lambda vs Accuracy graph, we get maximum accuracy when lambda equals 5. Even though it takes most time at this value of lambda, we will still choose this value because accuracy is more important for us than time. Therefore, optimal value of lambda is 5.

FINAL OPTIMAL VALUES

Hyper-Parameters	Optimal Value
Hidden Unit	50
Lambda	5
Training Set Accuracy	95.62%
Validation Set Accuracy	94.98%
Test set Accuracy	95.28
Time (In Secs)	91.31979012

Ignored features in the MNIST dataset (the index value of each column): 125 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 82, 83, 84, 85, 86, 87, 88, 110, 111, 112, 113, 114, 115, 139, 140, 141, 142, 167, 168, 169, 196, 224, 252, 308, 335, 336, 363, 364, 391, 392, 419, 420, 448, 449, 476, 477, 503, 504, 505, 532, 559, 560, 587, 588, 615, 616, 617, 643, 644, 645, 670, 671, 672, 673, 698, 699, 700, 701, 702, 725, 726, 727, 728, 729, 730, 731, 752, 753, 754, 755, 756, 757, 758, 759, 760, 779, 780, 781, 782, 783]

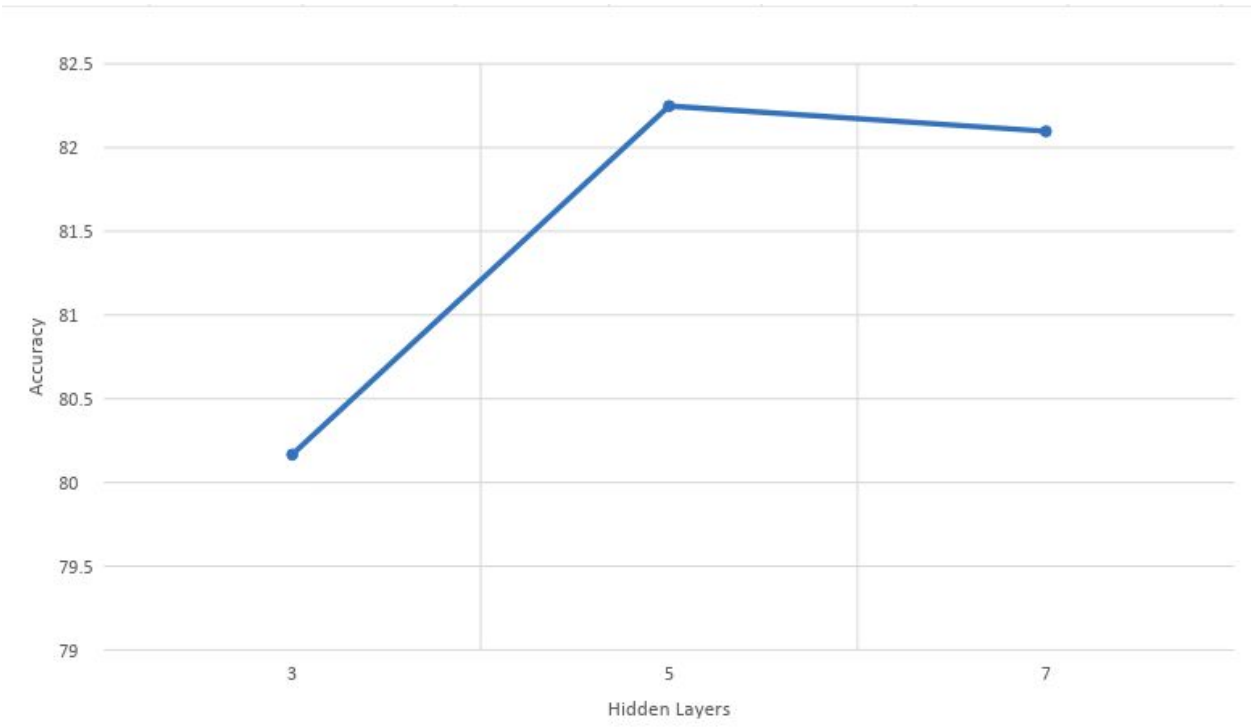
Part 2 - Compare results of normal neural network with deep neural network

Normal neural network with 1 hidden layer:

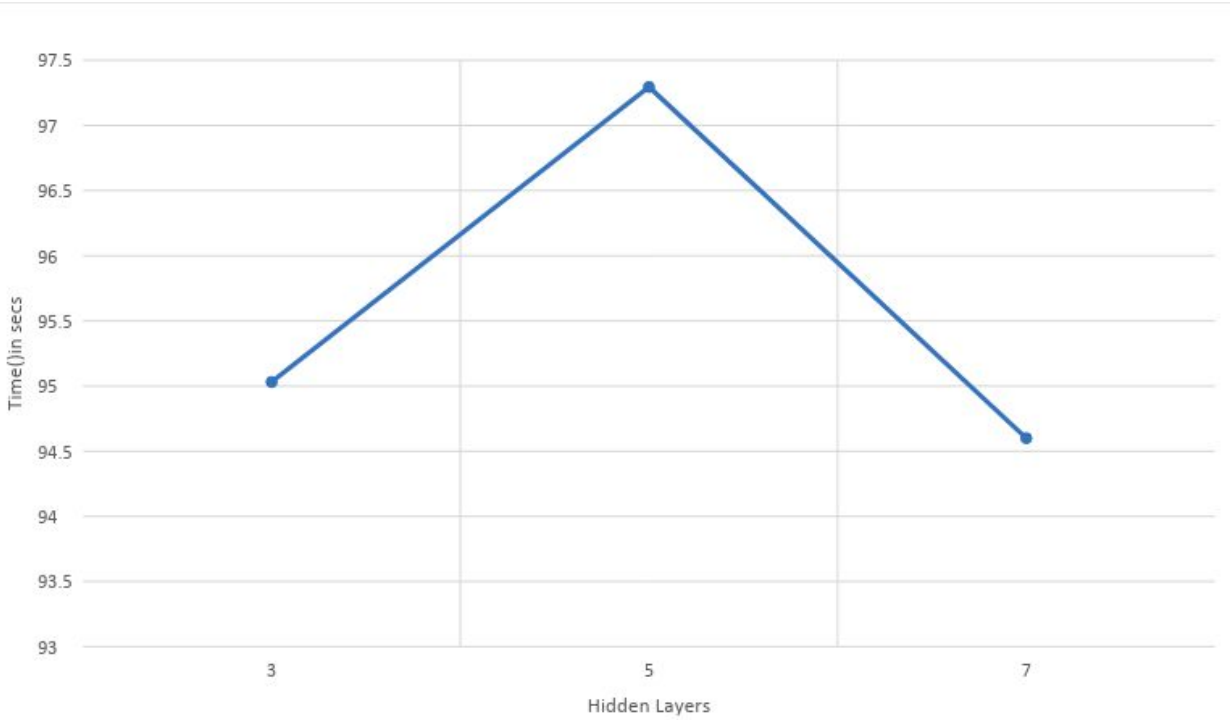
Hyper-parameters	Default Values	Optimal Values
Number of hidden units	256	50
Lambda	10	5
Training accuracy	85.12%	84.71%
Validation set accuracy	84.39%	83.18%
Test set accuracy	85.88%	85.12%

Deep neural network:

Layers	Accuracy
3	80.16%
5	82.24%
7	82.09%



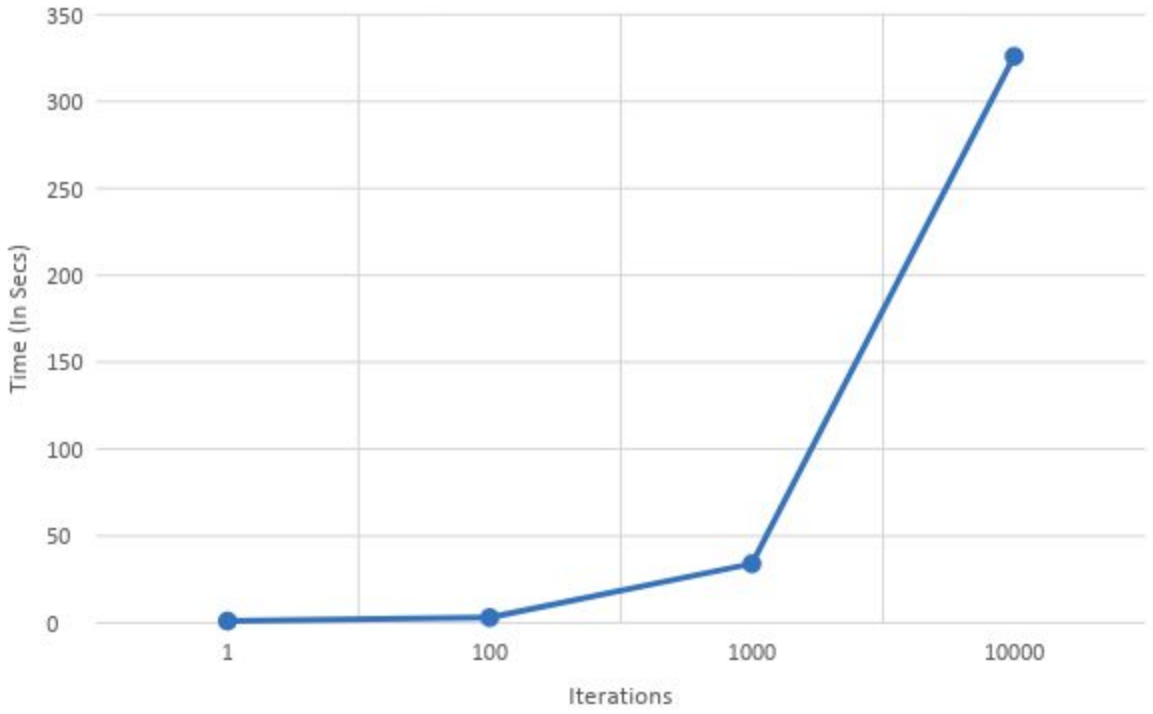
Layers	Time (in seconds)
3	95.03
5	97.29
7	94.60



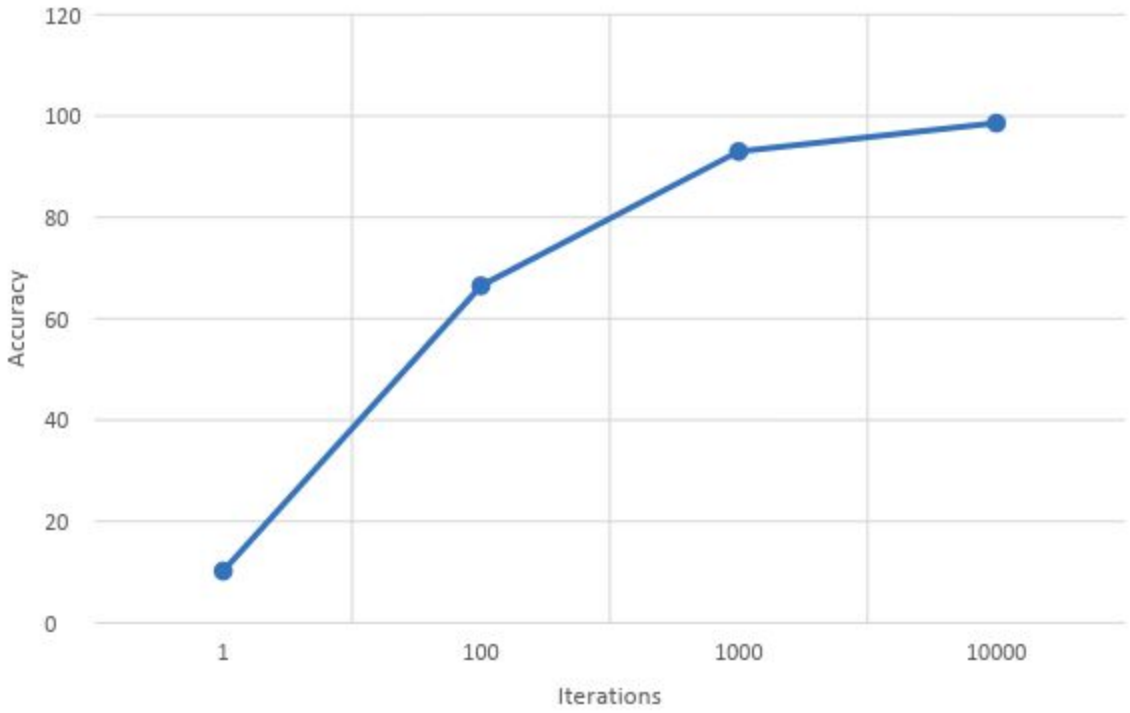
Conclusion: As is evident from the above graphs that the normal neural network with one hidden layer gives better accuracy than the deep neural network with any of 3, 5 or 7 hidden layers. This is because detecting spectacles is a slightly simpler problem which can be easily solved with just one hidden layer. Adding multiple hidden layers makes the neural network more complex which ends up overfitting the training data.

Part 3 - Convolutional Neural Networks

Iterations	Time (seconds)
1	1
100	3
1000	34
10000	326

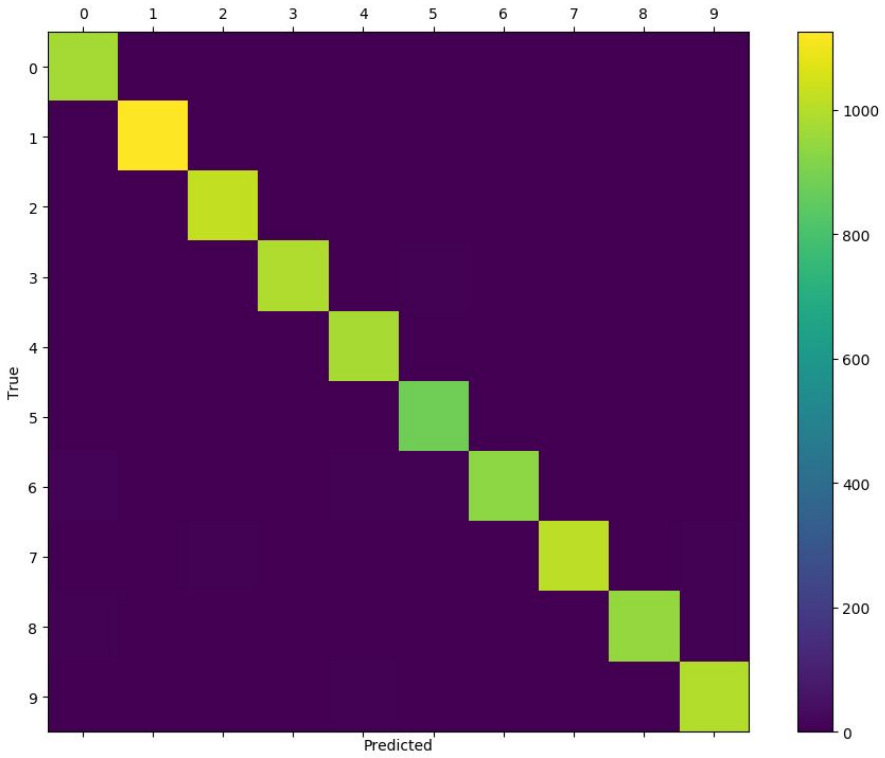


Iterations	Accuracy (%)
1	10.20
100	66.50
1000	93.00
10000	98.60



Confusion Matrix:

```
[[ 973    0    1    0    0    1    1    1    3    0]
 [   0 1134    1    0    0    0    0    0    0    0]
 [   1    2 1022    2    1    0    0    3    1    0]
 [   1    0    2 1002    0    2    0    1    2    0]
 [   0    0    2    0  975    0    0    3    0    2]
 [   2    0    0    8    0  880    1    0    1    0]
 [   9    4    0    0    3    6  936    0    0    0]
 [   0    3    5    1    0    0    0 1018    1    0]
 [   2    2    7    6    2    2    0    3  948    2]
 [   5    6    1    8    7    3    0    7    1  971]]
```



Conclusion: As is apparent from the above tables, convolutional neural networks give almost the same accuracy with much lesser time as compared to time time taken by our normal neural network with one hidden layer. Therefore, we can conclude that convolutional neural networks are more efficient.