State University of New York at Buffalo

CSE 574 – Introduction to Machine Learning

Spring 2017

Programming Assignment -1

Handwritten Digits Classification Project Report

Group 32

Ajay Gandhi – 50207403 – agandhi3@buffalo.edu

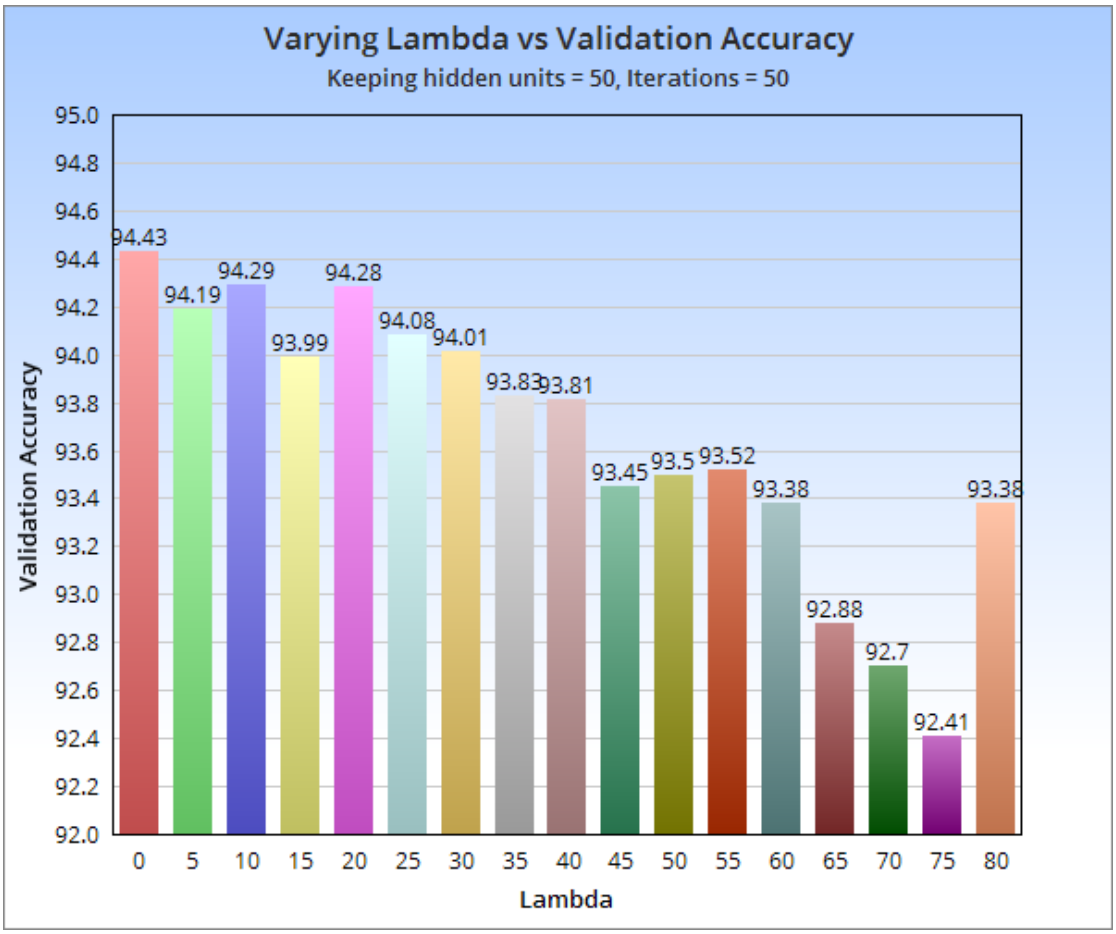
Apoorva Hejib – 50206516 – apoorvah@buffalo.edu

Siddharth Pateriya – 50206348 – spateriy@buffalo.edu

Choosing Hyper-Parameter for Neural Network

The accuracy and speed of a neural network depends on multiple factors such as number of hidden layers, number of hidden units in each layer, size of the data set, and the regularization parameter lambda, and the number of iterations the neural network runs for. Our implementation of the neural network for classifying handwritten digits from the MNIST data set consists of a single hidden layer. Thus we have performed various tests to tune the regularization parameter (lambda) and the find the optimal number of hidden units so that the neural network gives a high accuracy. These tests include varying lambda from 0 to 80, changing the number of hidden units and also determining the effect of training the network with higher iterations.

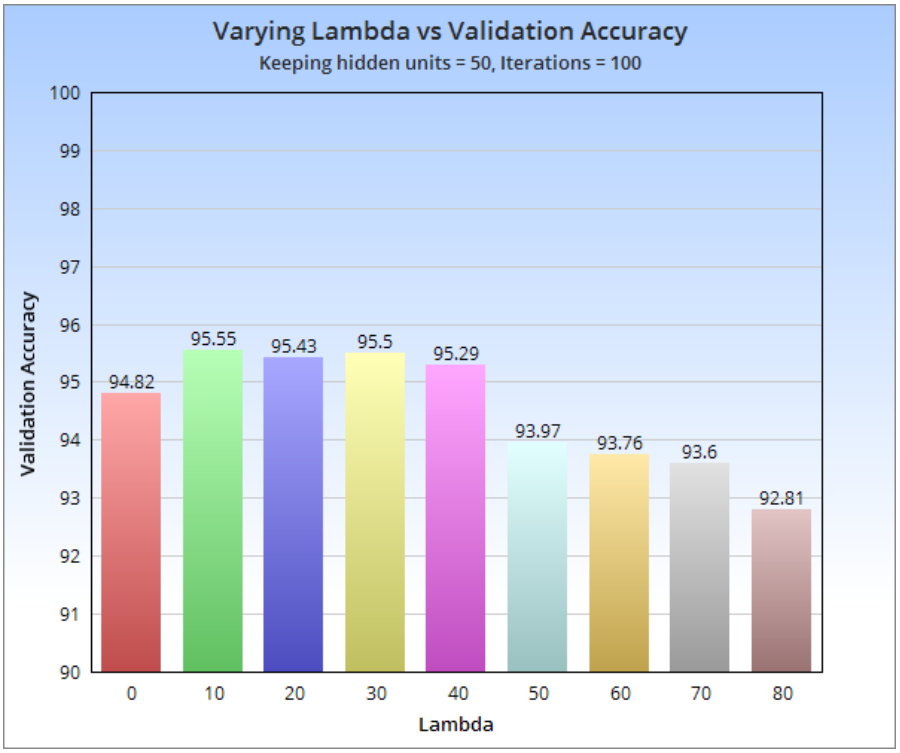
First, we tune Lambda to determine optimal values for it:



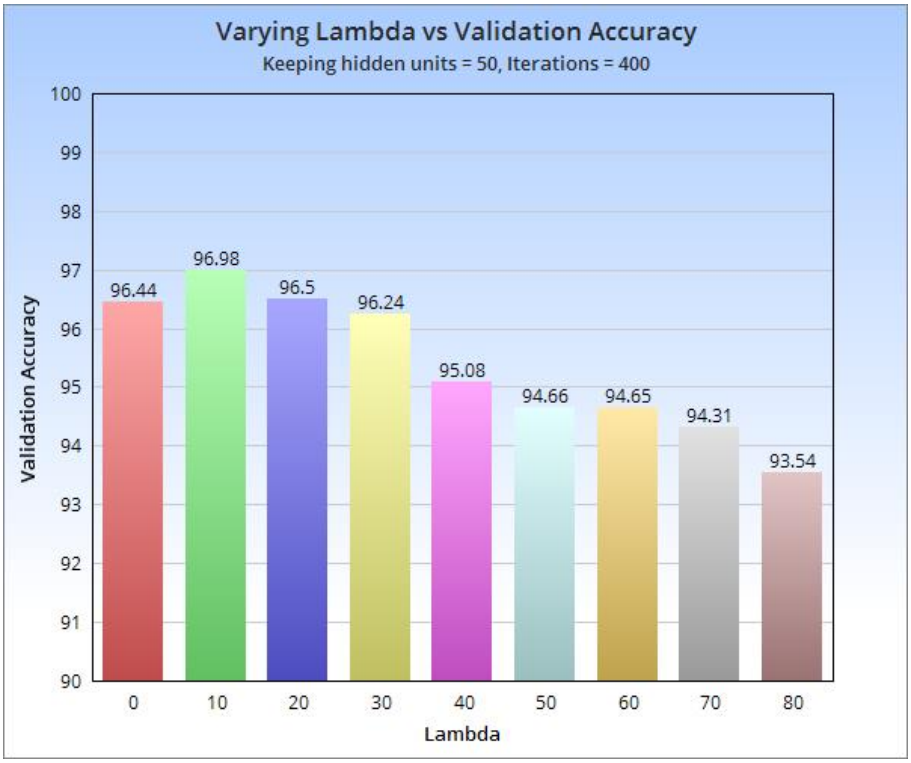
Keeping the number of hidden units as 50, and number of iterations as 50, we vary lambda from 0 to 80. As observed above, increasing lambda gradually resulted in decreased accuracy for the validation data. The top 2 accuracy values were when Lambda was kept 0 and 10.

Increasing number of iterations to 100.

We increased the number of iterations to 100 to see if we see a similar pattern when varying lambda, which we were able to confirm from the results.

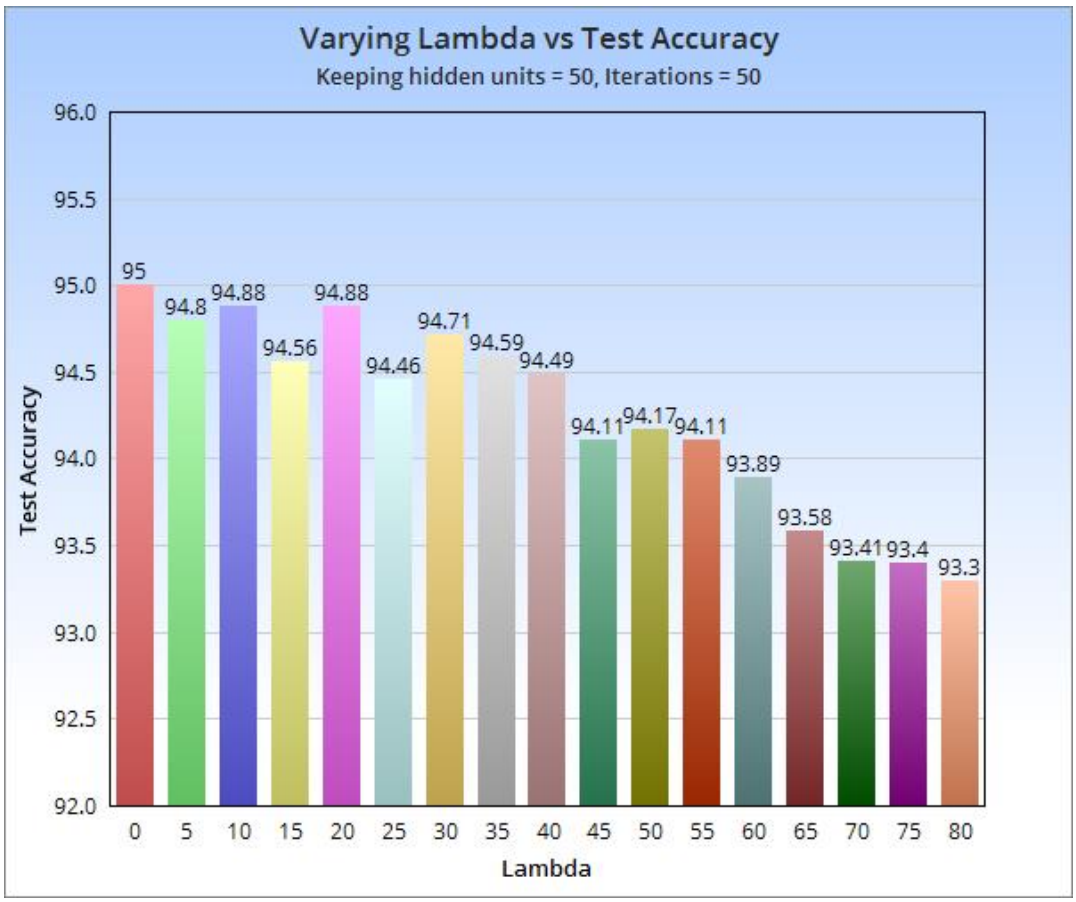


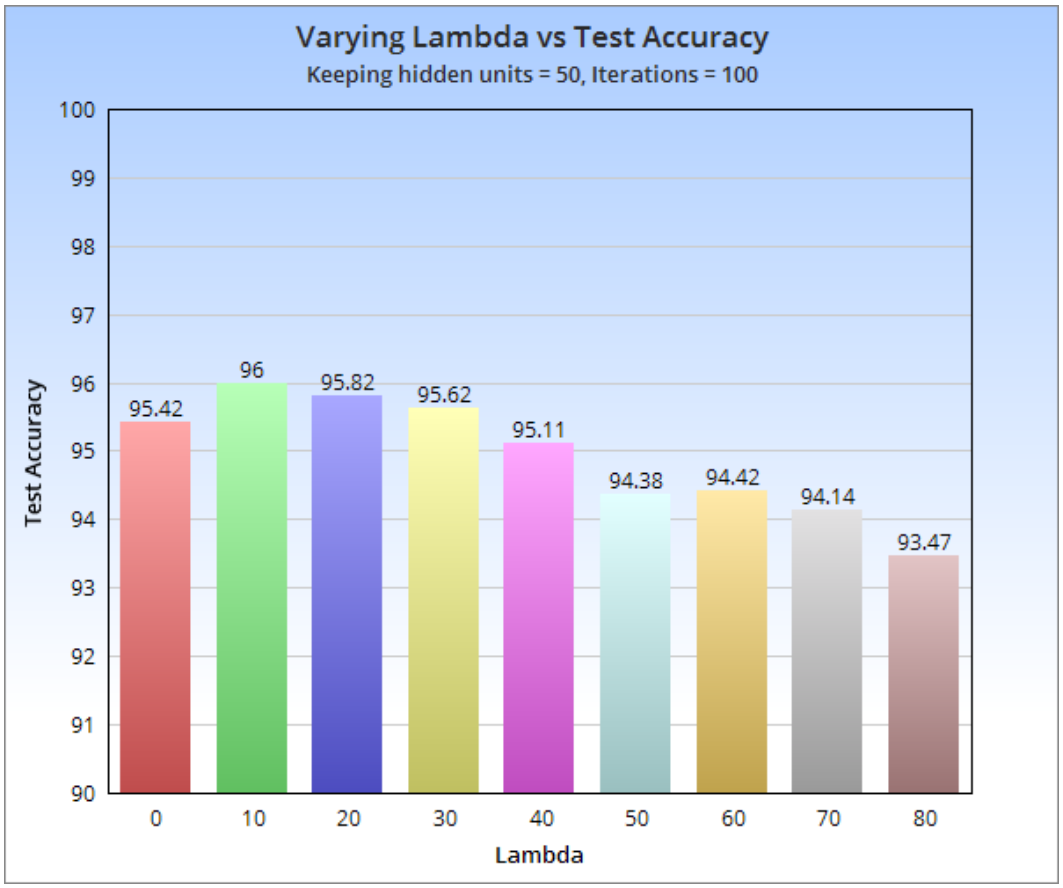
With 100 iterations too, increasing the value of Lambda gradually led to a decrease in the Validation set accuracy. Now we increase the number of iterations to 400.

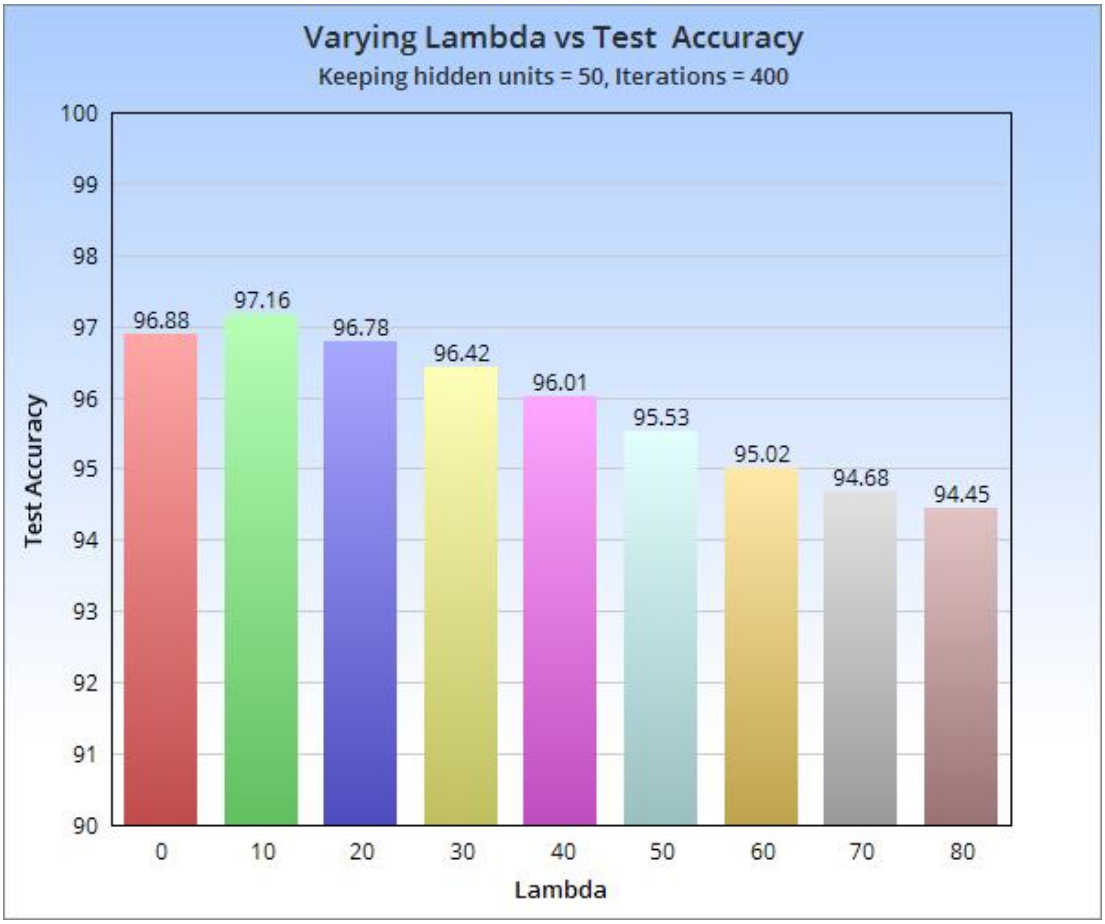


Again, it can be seen that increasing Lambda over a certain range leads to decrease in the accuracy of the validation data set**. Lambda from 0-30 more or less results in similar values, whereas as we go higher we see a decrease in the accuracy as we increase Lambda**. A similar trend was observed on the accuracy for the test data as well, as shown below.

Varying Lambda vs Test Data Accuracy







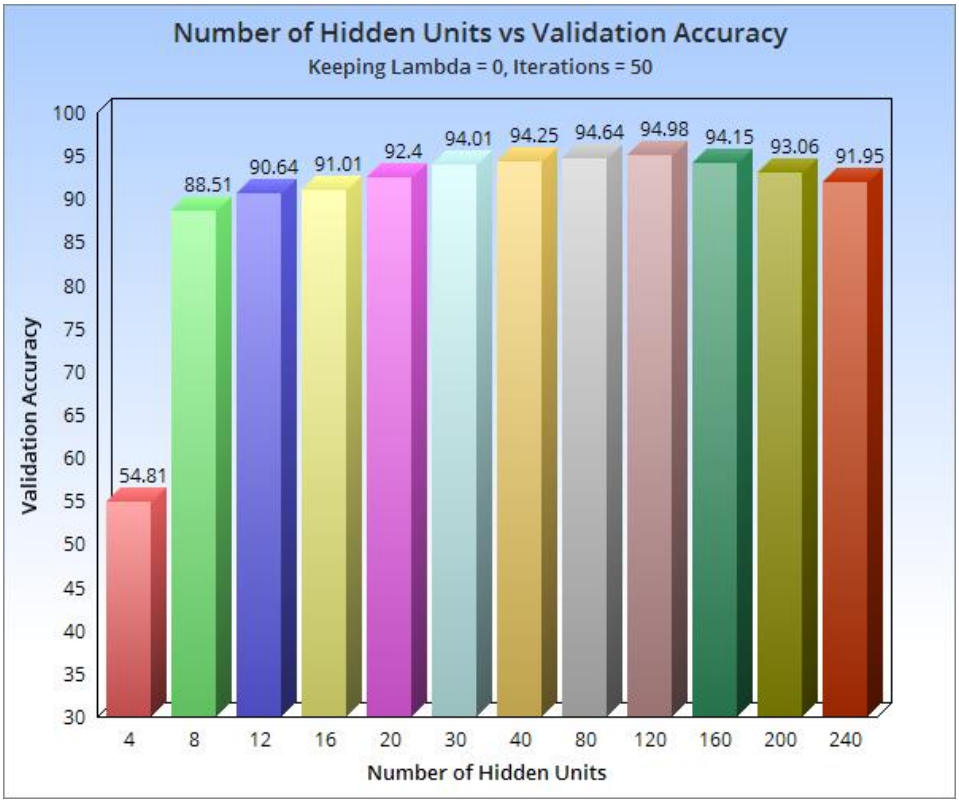
Looking at the performance above, we decided to further perform our tests keeping Lambda equal to 0 and 10, changing the number of hidden units and iterations, and also measuring the time taken to train the neural network.

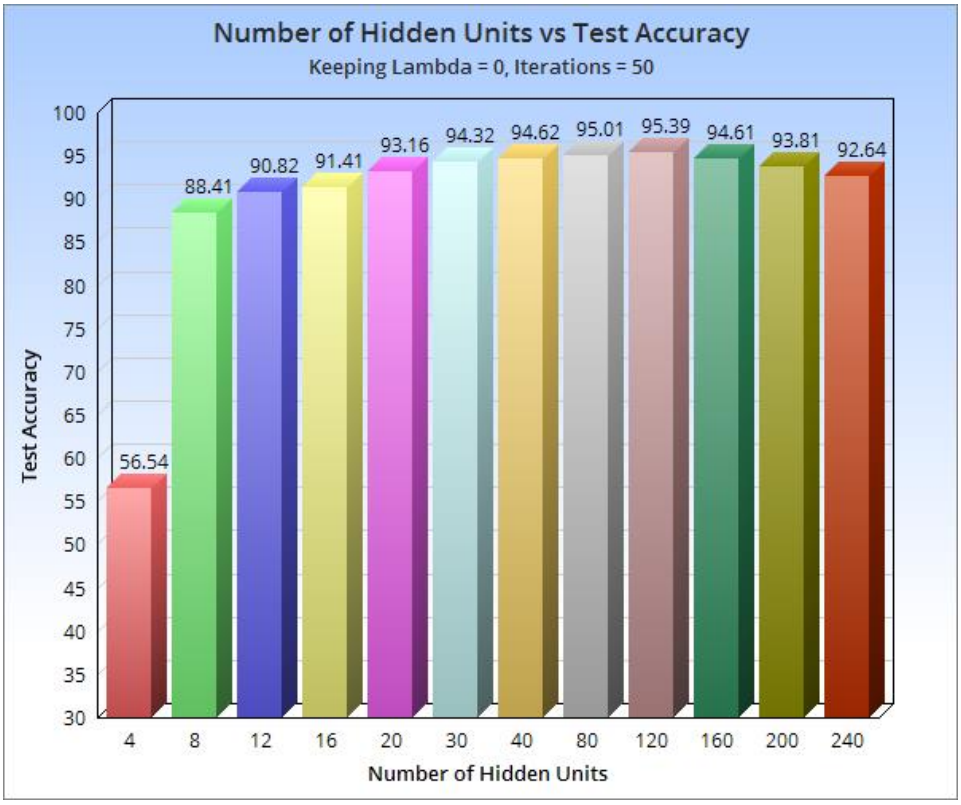
Effect of changing number of hidden units

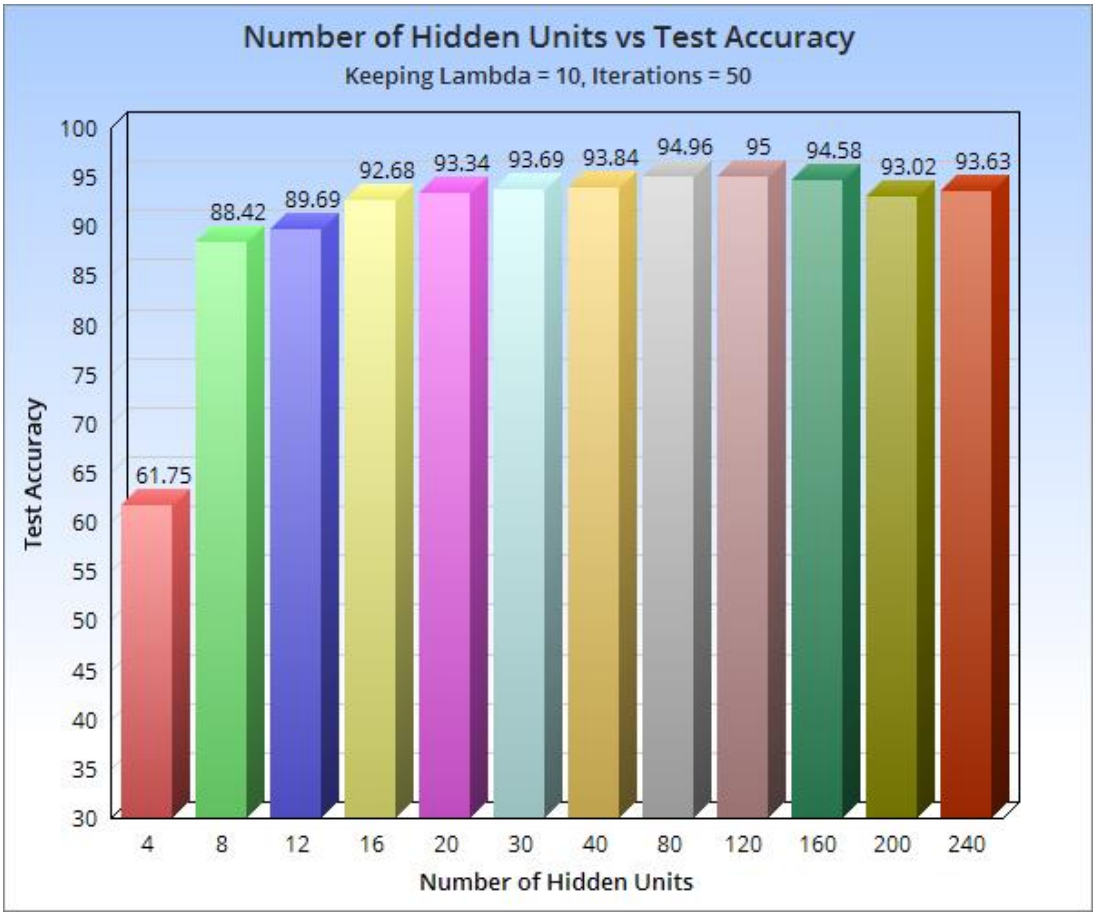
Keeping Lambda equal to 0 and 10, we varied the number of hidden units from 4 to 240, increasing the intervals as we moved forward, and ran the tests for iterations 50 and 400.

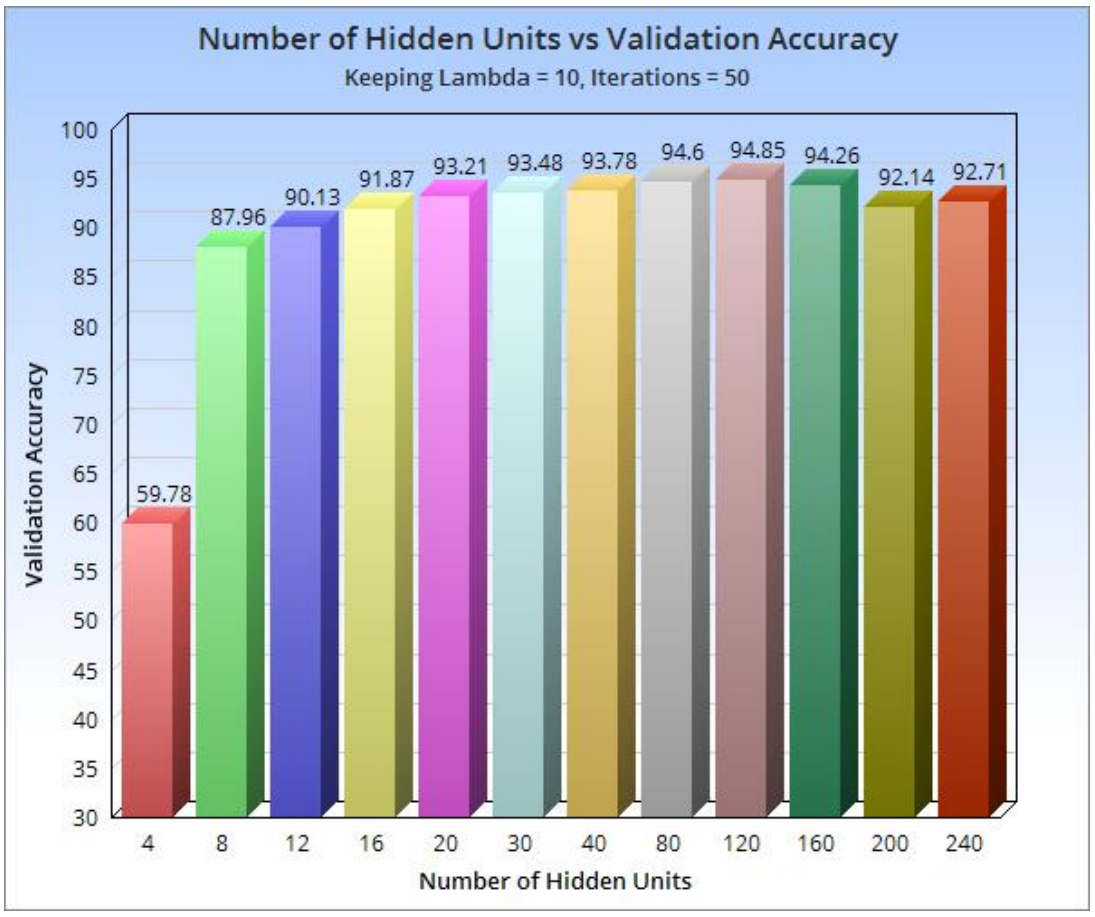
We see an initial increase in accuracy when increasing the number of hidden nodes, which decreased after a point. In this case, **we achieve max accuracy with hidden nodes = 120, and it decreases gradually as we increase the number of hidden nodes**.

Also, we observe an increase in the time to train the network as we increase the number of hidden nodes. **Increasing the number of iterations from 50 to 400 resulted in the instances of higher nodes giving much higher accuracy (upto 98%), but also took a lot of time to train (around 2 hours).**

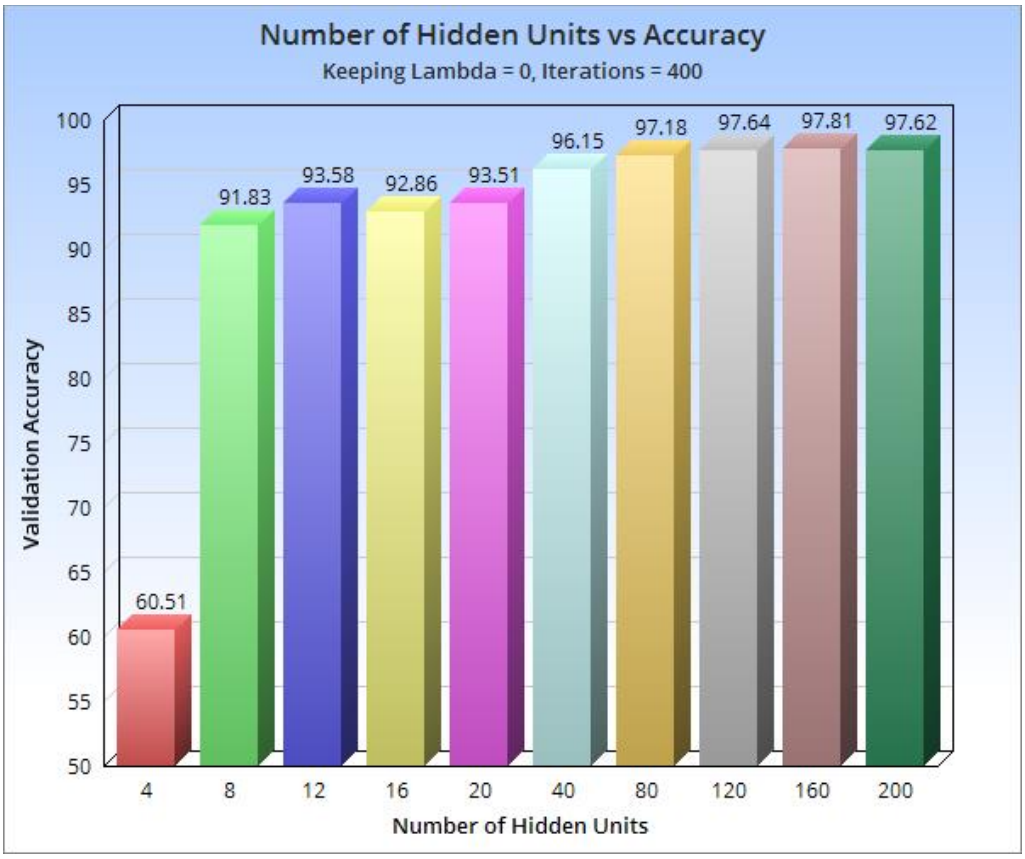

Number of Hidden Units vs Validation Accuracy
Keeping Lambda = 0, Iterations = 50
Validation Accuracy
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
54.81
88.51
90.64
91.01
92.4
94.01
94.25
94.64
94.98
94.15
93.06
91.95
4
8
12
16
20
30
40
80
120
160
200
240
Number of Hidden Units

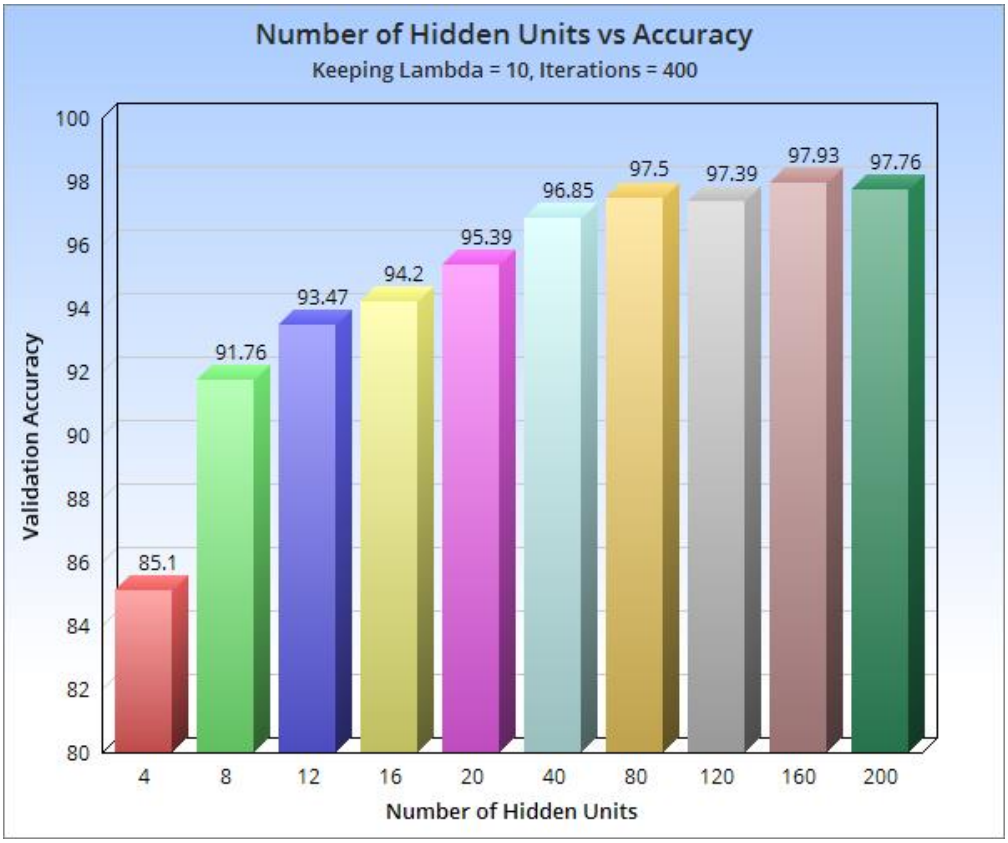

Number of Hidden Units vs Test Accuracy
Keeping Lambda = 0, Iterations = 50
Test Accuracy
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
56.54
88.41
90.82
91.41
93.16
94.32
94.62
95.01
95.39
94.61
93.81
92.64
4
8
12
16
20
30
40
80
120
160
200
240
Number of Hidden Units


Number of Hidden Units vs Test Accuracy
Keeping Lambda = 10, Iterations = 50
Test Accuracy
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
61.75
88.42
89.69
92.68
93.34
93.69
93.84
94.96
95
94.58
93.02
93.63
4
8
12
16
20
30
40
80
120
160
200
240
Number of Hidden Units


Number of Hidden Units vs Validation Accuracy
Keeping Lambda = 10, Iterations = 50
Validation Accuracy
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
59.78
87.96
90.13
91.87
93.21
93.48
93.78
94.6
94.85
94.26
92.14
92.71
4
8
12
16
20
30
40
80
120
160
200
240
Number of Hidden Units

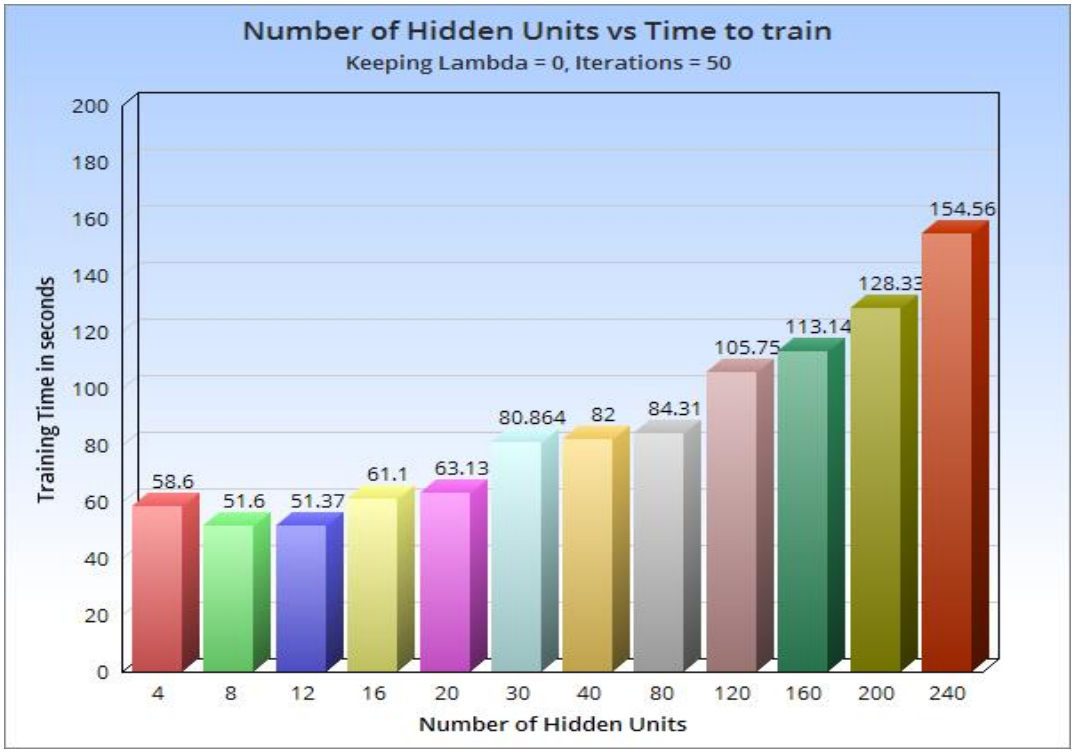
From the results presented above, we were able to conclude that increasing the number of hidden units leads to increase in accuracy upto a threshold, after which the accuracy starts decreasing. Another interesting observation came when we ran the above tests with 400 iterations. The accuracies went much higher to around 98%, but the network took much longer to train.

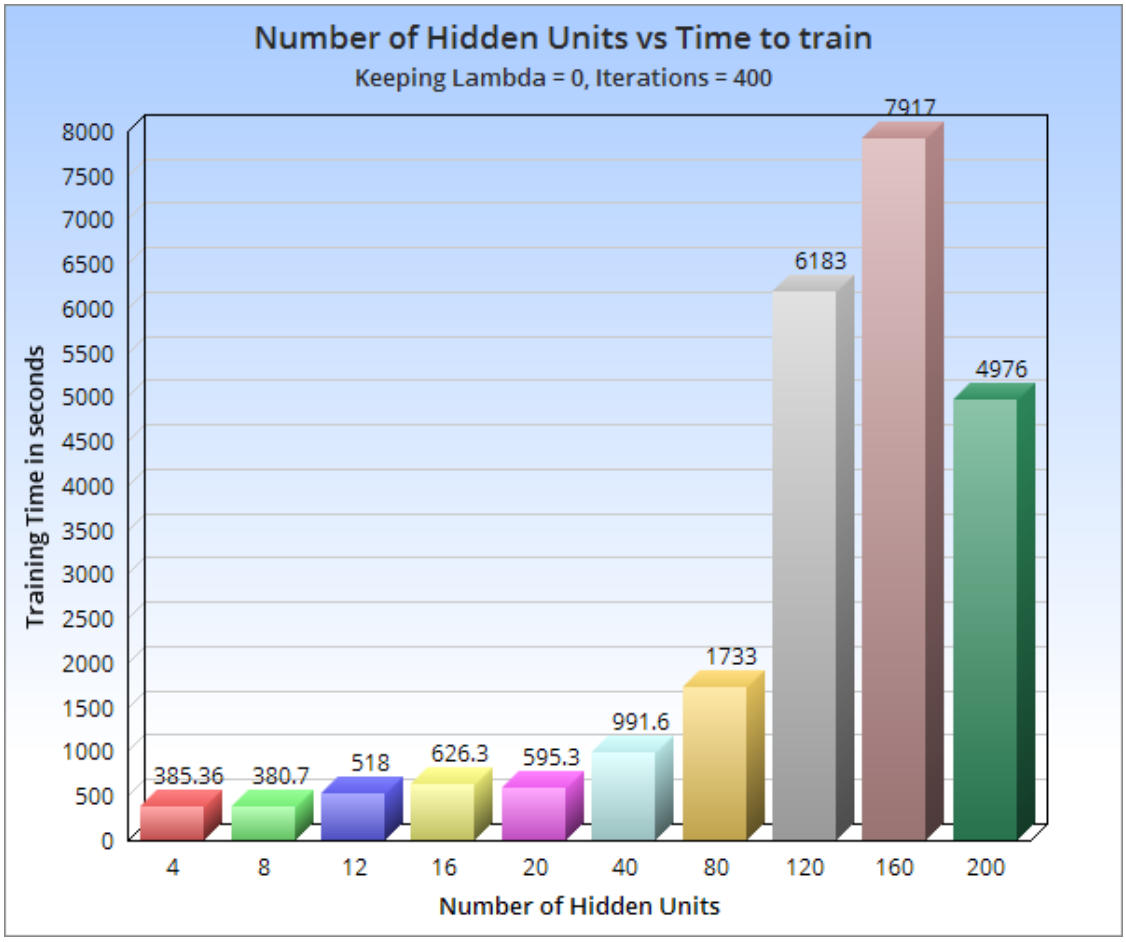


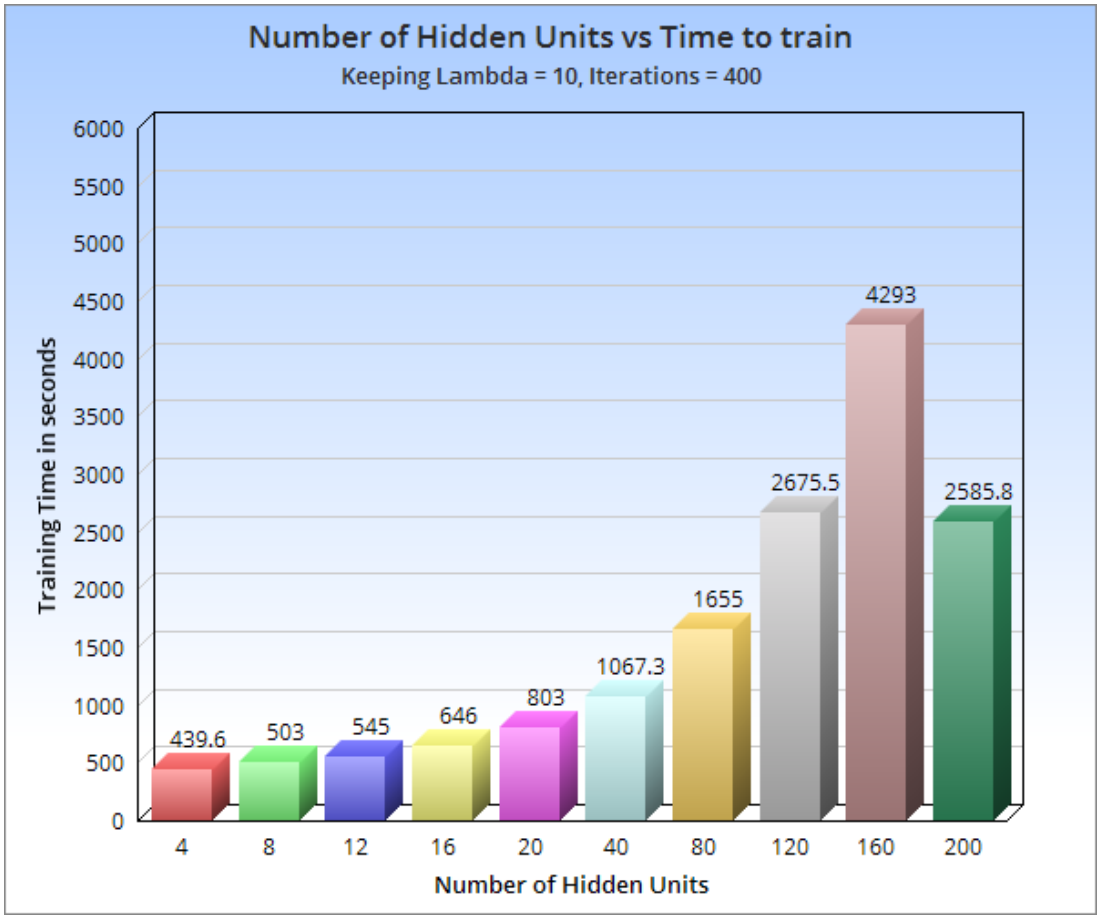


Effect of increasing hidden units on training time:

Increasing the number of units was also directly co-related to an increase in the training time of the neural network as evidenced by the results below







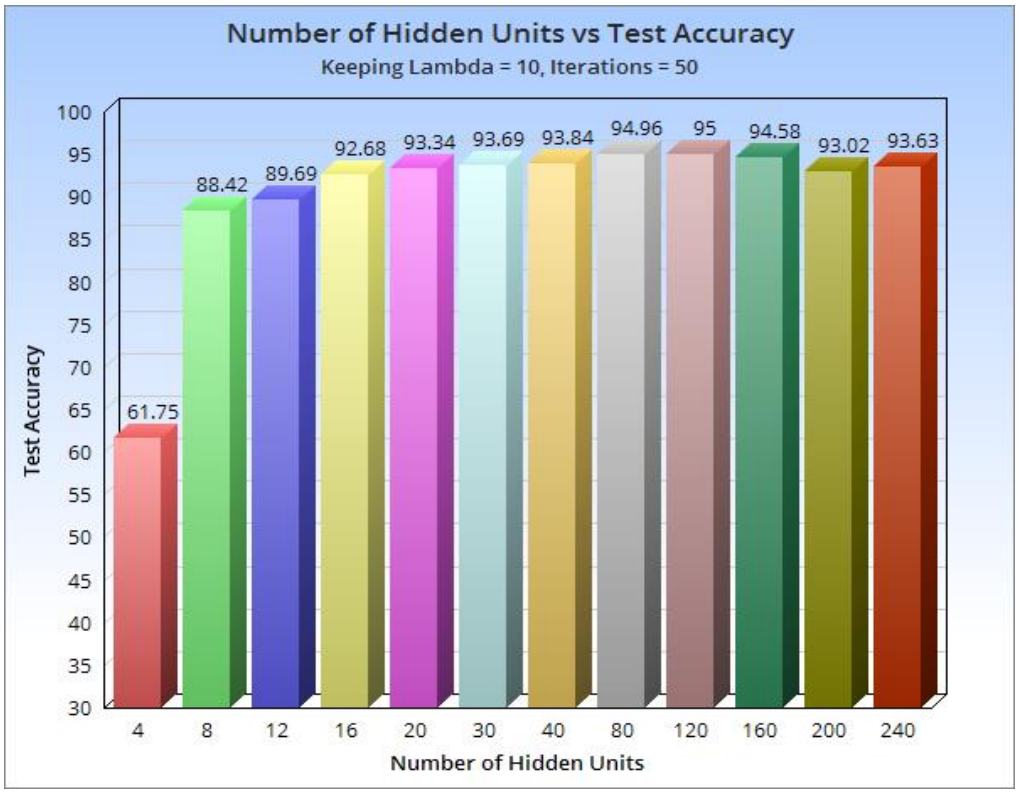
Thus we can establish a general trend that increasing the number of hidden units also increases the time taken to train the neural network.

Selection of Lambda and Number of Hidden Units:

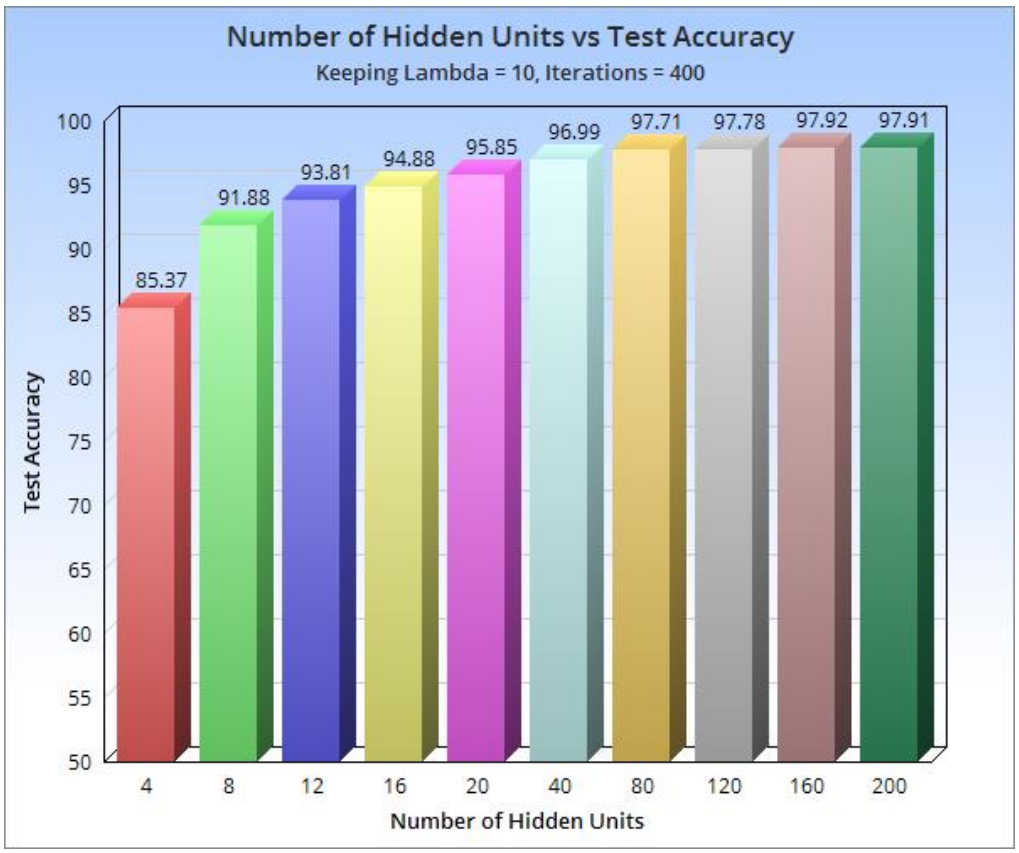
Based on the above results, we chose Lambda as 10, and the number of hidden units as 120. At 400 iterations, having even higher number of hidden units gives even better accuracy, but hampers performance as training time is increased by a lot. Hence we came to the conclusion that we should choose the optimal lambda as 10, and optimal number of hidden units as 120.

Accuracy of neural network on handwritten digits:

The accuracy we achieved with the optimal parameters was close to 95% on the test data set (for 50 iterations), and 97.78% when running for 400 iterations.



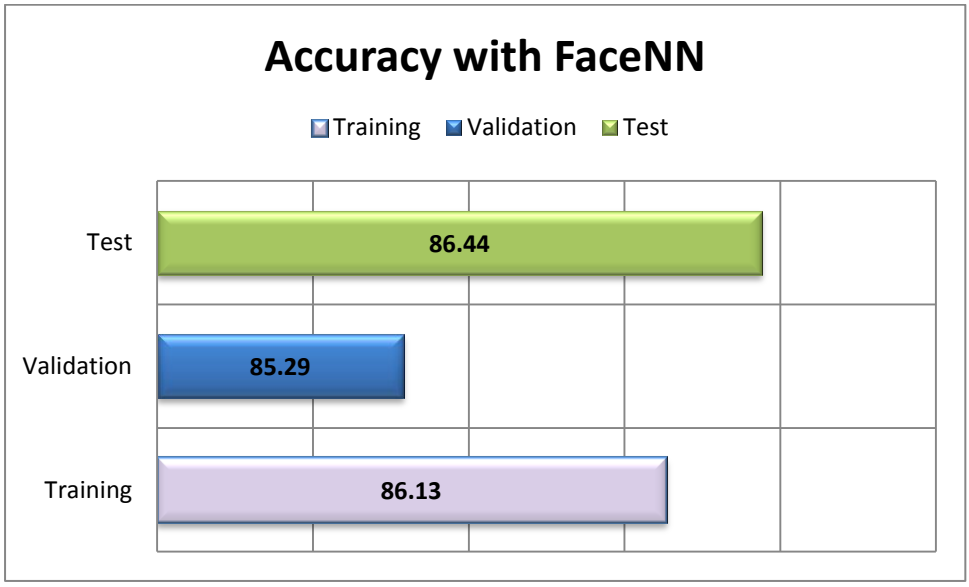
We can see that accuracy reaches 95% when running with 120 hidden units, keeping Lambda = 10.



For 400 iterations, we see that accuracy reaches around 97.8% with 120 hidden units, keeping Lambda = 10.

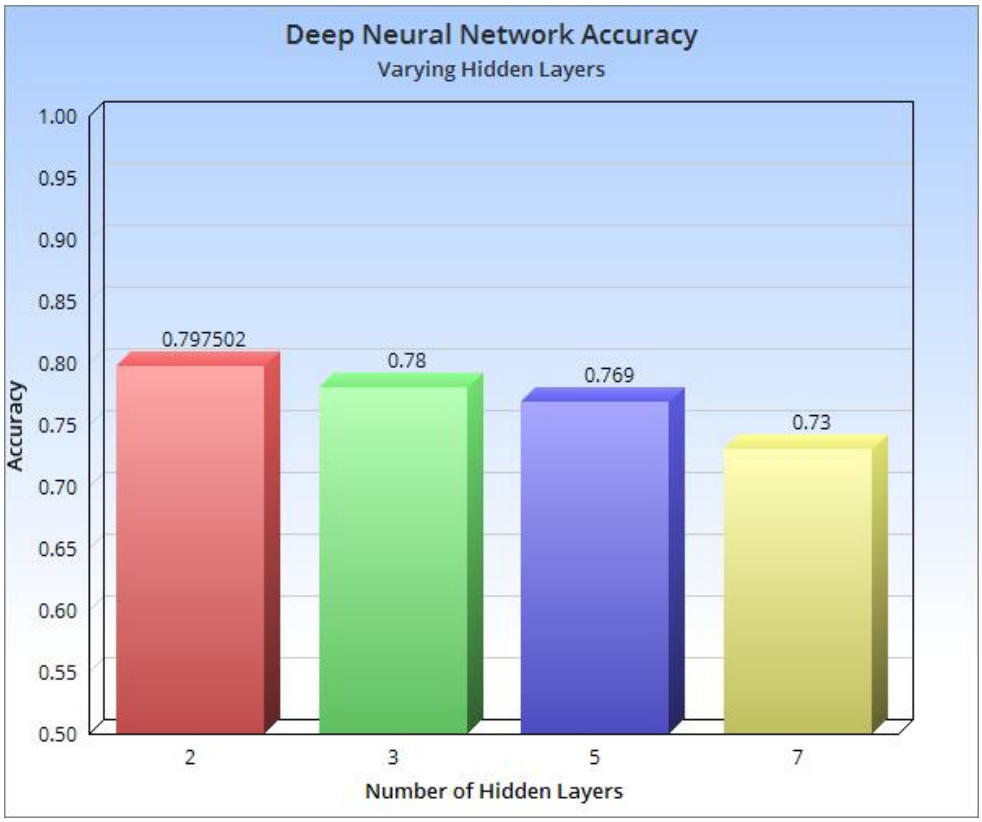
Accuracy of classification method on the CelebA data set:

For the CelebA data set, we ran the facennScript python script on Jupyter on our personal machine. We achieved an accuracy of **86.44%** with the test set.



Comparison of CelebA results with TensorFlow Neural Network

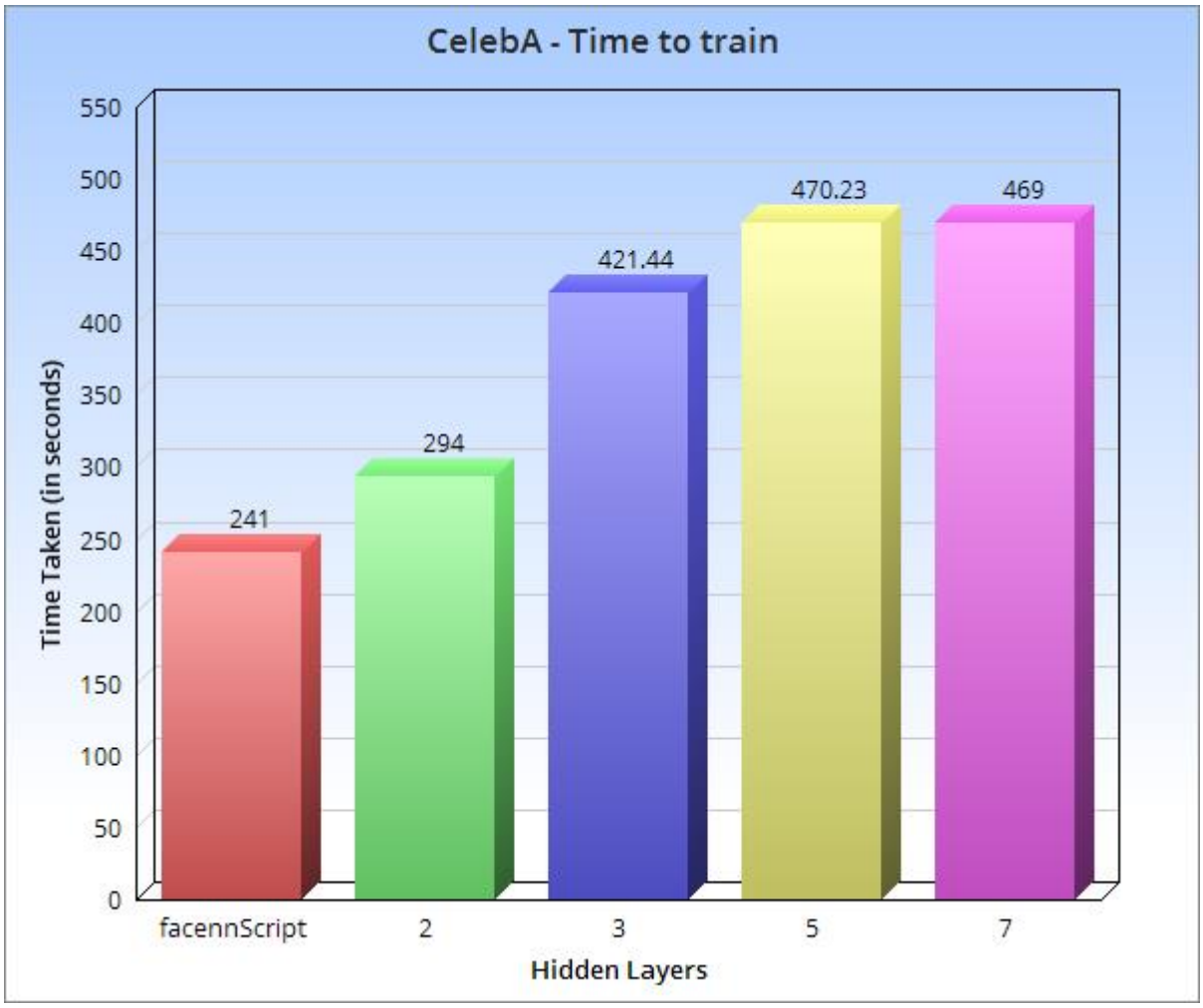
Running the Deep Neural Network with 3, 5 and 7 hidden layers, we got the following results:



We see that increasing the number of hidden layers doesn't necessarily increase the accuracy of the deep neural network, in fact, we see a decrease when we test it with 7 hidden layers.

Also, comparing the results between the Single Layer Neural Network and the TensorFlow Deep Neural Network, it is surprising to see that the accuracy of the single layer neural network comes out to be higher than that of a deep Neural Network. Perhaps this may change depending on other factors such as learning rate, number of iterations and number of hidden nodes used.

Comparison of time taken to train Single layer network vs Deep Neural Network:



The scripts facennScript.py and deepnnScript.py were run on the UB CSE servers Metallica and Springsteen, respectively. We tried 2, 3, 5 and 7 hidden layers for the deep neural network. A general observation was that the facennScript ran faster than the deep neural network script. Also, increasing the number of hidden layers increased the training time (though not by much, in case of 50 iterations). For 500 iterations, the difference in training time was much higher. Note that training time also depends on the available resources on the server at the time the script was being run, so in case a lot of users were running their respective jobs at the same time, this might have had some effect on the training time, but it would be safe to say that increasing the number of hidden layers would increase the training time.