CSE 574: Introduction to Machine Learning

Programming Assignment 2:
Handwritten Digit Classification

Instructor: Dr. Mingchen Gao
Date: April 18, 2018

Submitted by,
Muthuvel Palanisamy,
Person # - 50246815

Single Layer - Neural Networks

Implementation

a. preprocess ():
   - Included feature extraction method in which all the features will null values from every training samples are omitted.

b. sigmoid ():
   - This function calculates the sigmoid value of a matrix/ vector / scalar value

c. nnObjFunction ():
   - Calculates the obj_val – error and the obj_grad of the neural network
   - Called through scipy.optimize.minimize function
   - Starts by propagating input through feed forward network and the calculates the error
   - Then back propagates the error to adjust the weights and calculate gradient
   - Implemented the equations given in the pdf

d. nnPredict ():
   - Predicts the label of inputted data given hidden and output layer weights

Params.pickle – contains a list {selected_features, n_hidden, w1, w2, lambdaval}

Result of Implementation

Accuracy of Test Data on MNIST (lambda = 5, hidden nodes =50) **on testing set= 95.06%**
Training time on MNIST (lambda = 5, hidden nodes =50) = **45 seconds**
Accuracy of Test Data on CelebA (lambda = 10, hidden nodes =256 **) = 85.27%**
Training time on CelebA (lambda = 10, hidden nodes =256) **= 115 seconds**

1. **Effect of Number of Hidden Nodes on Training (MNIST)**

Table 1: Observation of Training output for hidden nodes = [0 to 50], Lambda = 5

No. of Hidden Nodes	Training Set Accuracy	Validation set accuracy	Testing set accuracy	Final Optimization Parameter	Time Taken (sec)
0	11.48%	10.0%	11.35%	3.25	3.80
4	61.44%	60.13%	61.28%	1.72	26.58
8	88.82%	88.29%	88.46%	0.79	27.32
12	91.85%	90.93%	91.81%	0.61	28.28
16	92.31%	91.39%	91.62%	0.59	30.70
20	93.63%	92.65%	93.08%	0.49	30.53
30	94.61%	94.19%	94.21%	0.43	44.02
40	94.86%	94.26%	94.58%	0.41	45.22
50	95.31%	94.86%	95.05%	0.38	53.52

Figure 1: Number of Hidden Nodes vs Accuracy in Single Layer NN

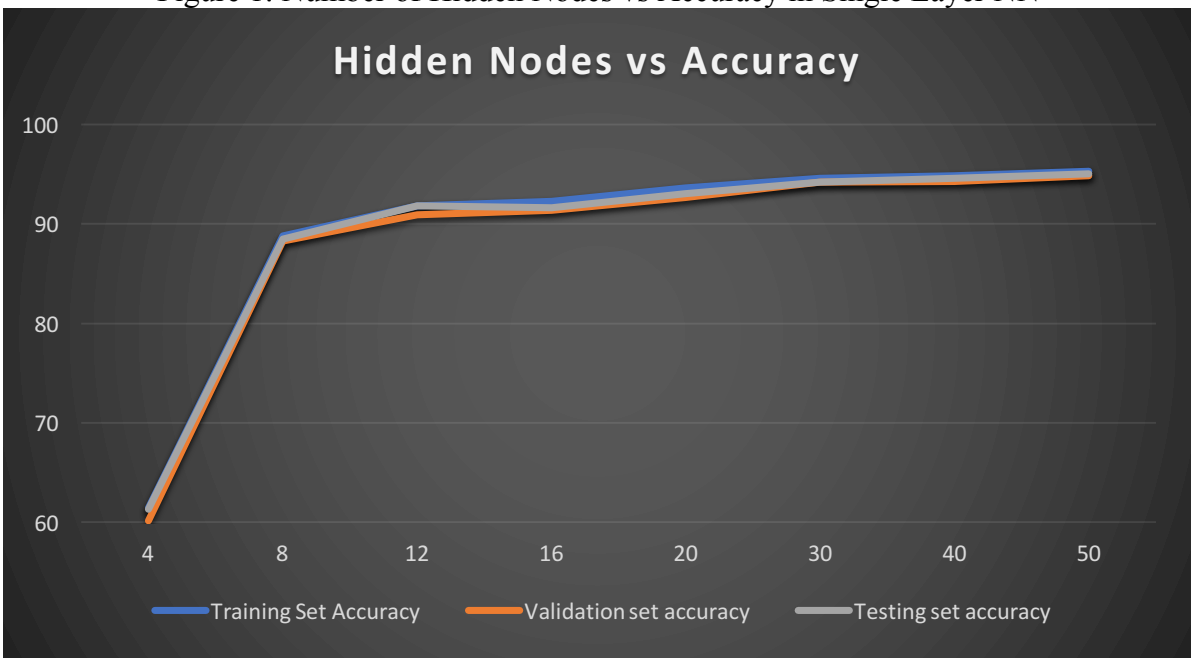


Figure 2: Number of Hidden Nodes vs Training Time in Single Layer NN

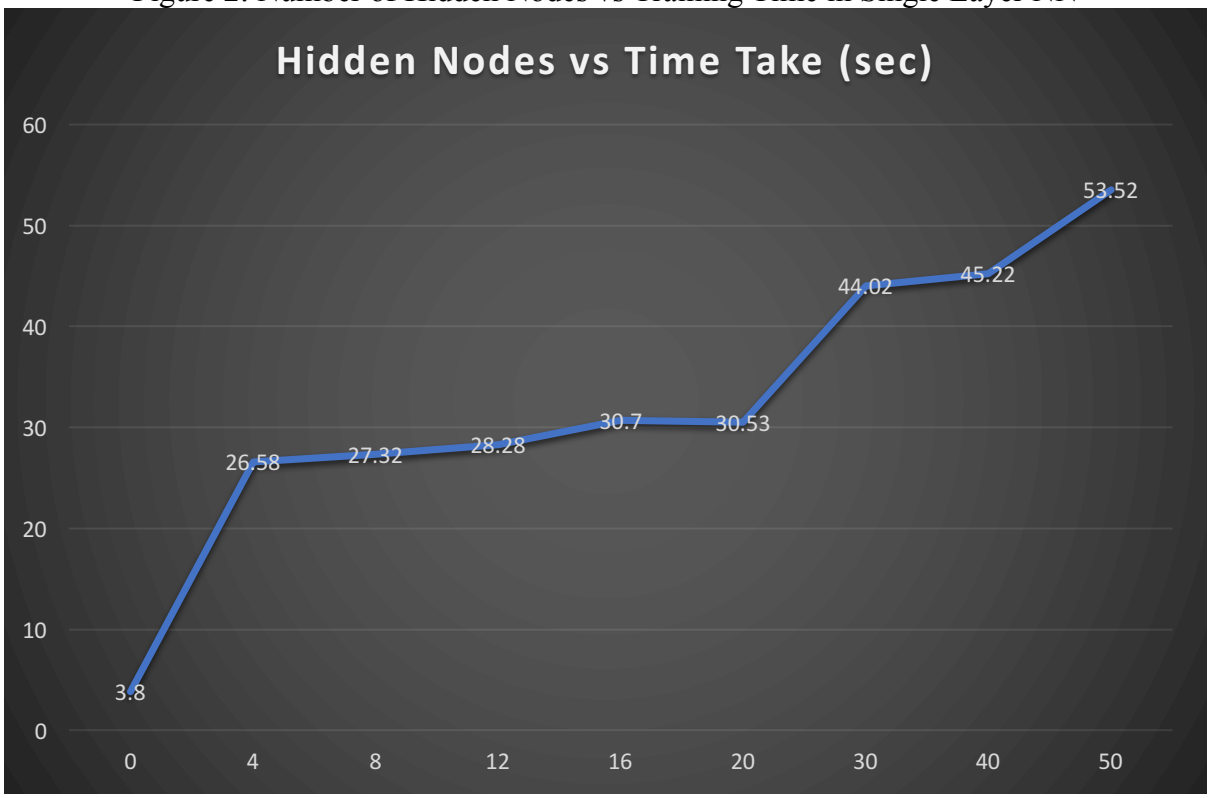
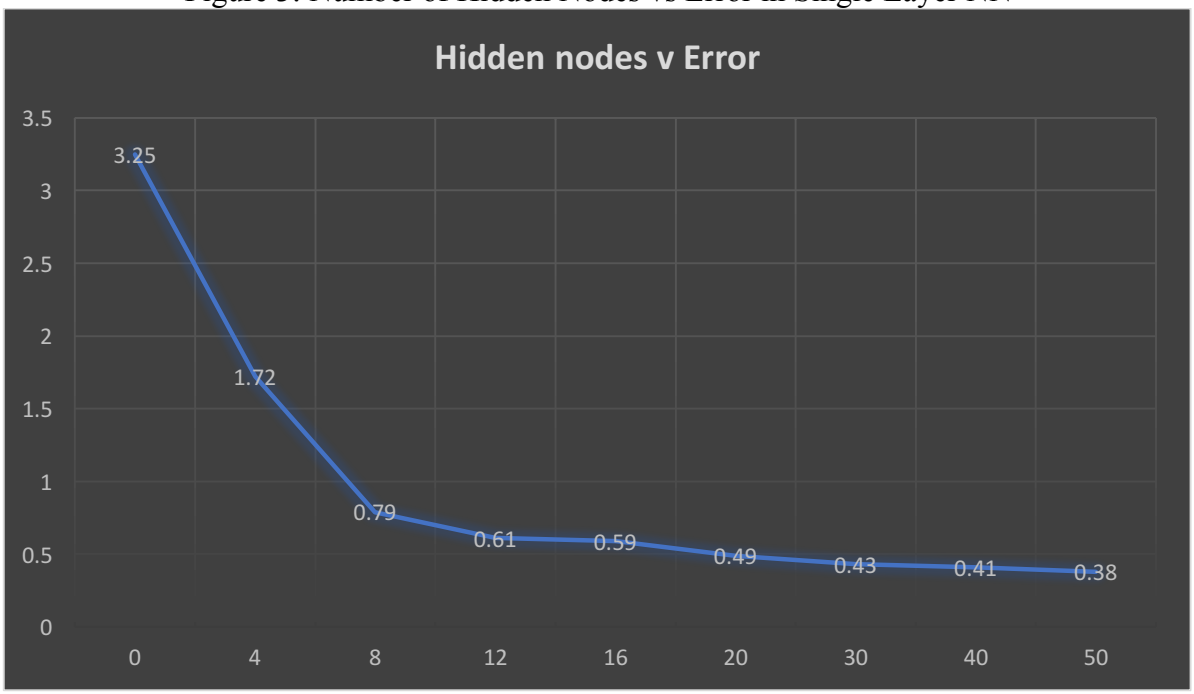


Figure 3: Number of Hidden Nodes vs Error in Single Layer NN



2. Effect of Regularization Parameter – Lambda on Training (MNIST)

Table 2: Observation of Training output for lambda = [0 to 60], Hidden Nodes = 20

Lambda	Training Set Accuracy	Validation set accuracy	Testing set accuracy	Final Optimization Parameter	Time Taken (sec)
0	93.34%	92.39%	92.84%	0.45	45.65
5	93.95%	93.08%	93.56%	0.39	39.57
10	93.77%	93.12%	93.33%	0.53	36.99
15	93.606%	92.67%	93.28%	0.60	43.52
20	93.53%	92.85%	93.24%	0.63	42.00
25	93.52%	92.75%	93.25%	0.66	40.11
30	93.07%	92.57%	92.86%	0.72	40.27
35	93.13%	92.67%	93.10%	0.73	39.26
40	92.85%	92.17%	92.60%	0.77	41.77
45	93.39%	92.88%	93.30%	0.78	43.62
50	93.07%	92.49%	93.14%	0.81	43.83
55	92.82%	92.21%	92.63%	0.88	44.67
60	93.07%	92.33%	93.11%	0.86	37.0

Figure 4: Regularization Parameter lambda vs Accuracy in Single Layer NN

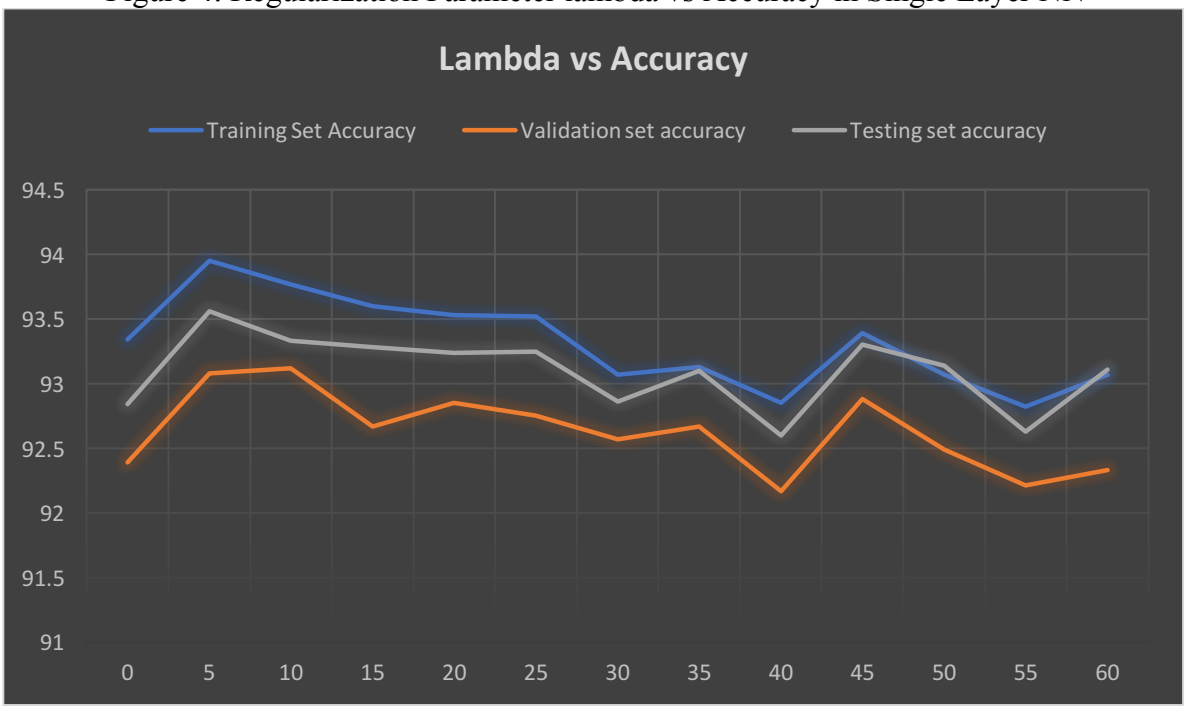


Figure 5: Regularization Parameter lambda vs Training Time in Single Layer NN

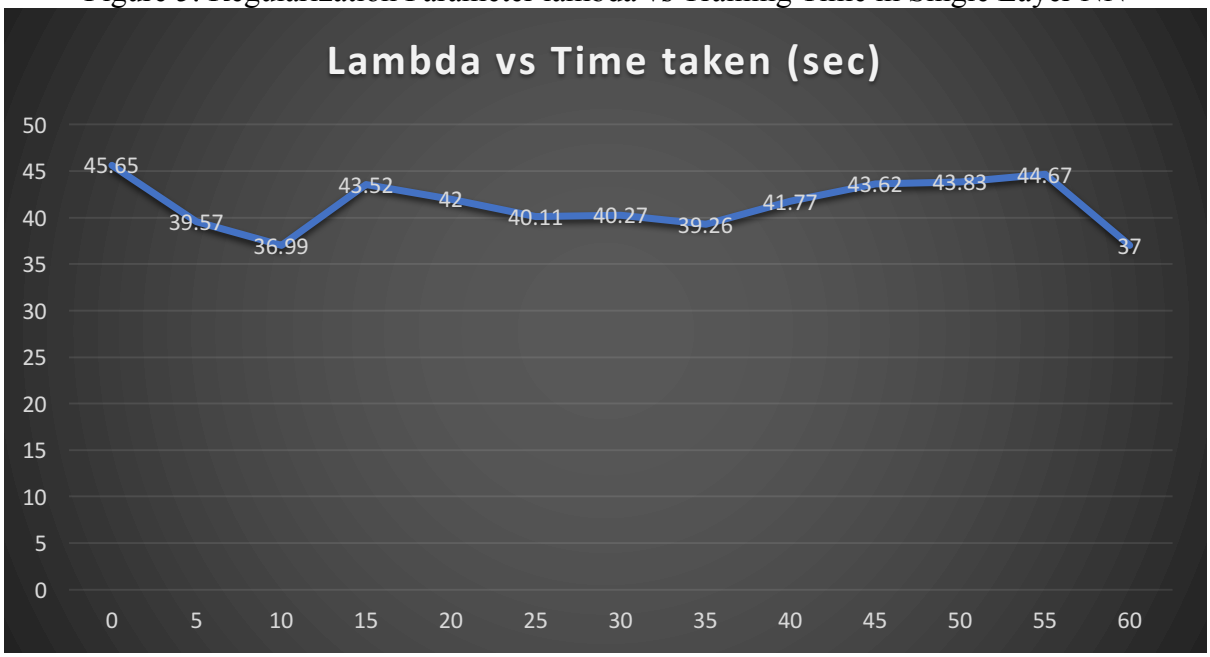
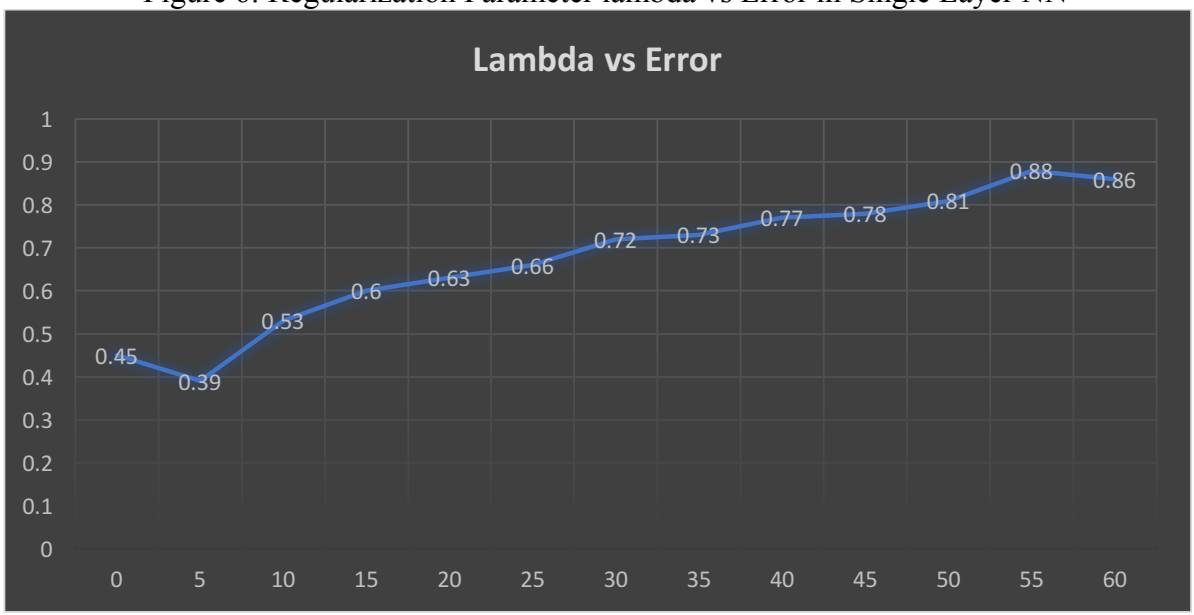


Figure 6: Regularization Parameter lambda vs Error in Single Layer NN



How to choose hyper–parameters for single layer Neural Networks

- **The ideal number of hidden nodes** would be **50 and any value above 20 is advisable** because they produce higher efficiency without sacrificing training time (evident from figures 1 to 3). They also have very low error from the objective function. The accuracy

of MNIST dataset will be above 93% and the accuracy of CelebA dataset will be above 84% if you follow the above mentioned parameter. . Values below 20 will cause underfitting

- Please note that increasing the hidden notes does not necessarily increase the efficiency significantly after a point. Go for higher number of features if you can afford another 10 to 20 seconds of training time for a few percentage increases in accuracy. Also, very low number of hidden nodes will give a poorly training network
- **The ideal value for regularization parameter lambda** would **5.** From figures 4 to 6 and Table 2, we could observe that the network gives the best result when lambda ranges from 5 to 10. The network maintains the accuracy above 93% while keeping the training time below 40 seconds. Testing on CelebA, the accuracy also sticks to above 84% for this lambda value.
- The objective function increases directly with increase in lambda (figure 6)
- Always have a regularization factor. No regularization means very less accuracy because of overfitting while higher regularization may result in underfitting

Deep Neural Networks

Table 3: Observation of Time and Accuracy with increase in No of hidden layers

Number of Hidden Layers	Number of Features per layer	Accuracy (%)	Time Taken (sec)
2	32	81.18	58.60
2	64	81.49	69.29
2	256	81.45	153.34
3	32	80.46	64.32
3	64	80.77	66.23
3	256	80.88	174.35
5	32	68.96	69.24
5	64	72.18	67.15
5	256	73.58	197.12
7	32	69.22	63.75
7	64	69.03	78.13
7	256	74.79	230.36

Figure 7: Number of Hidden Layers Vs Accuracy in Deep Neural Network
(No. of features per layer = 256)

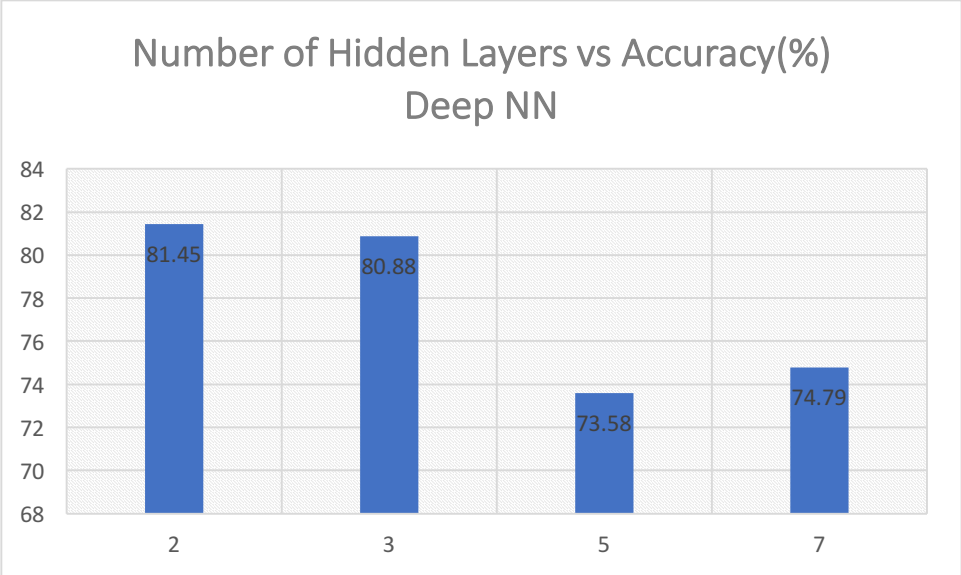
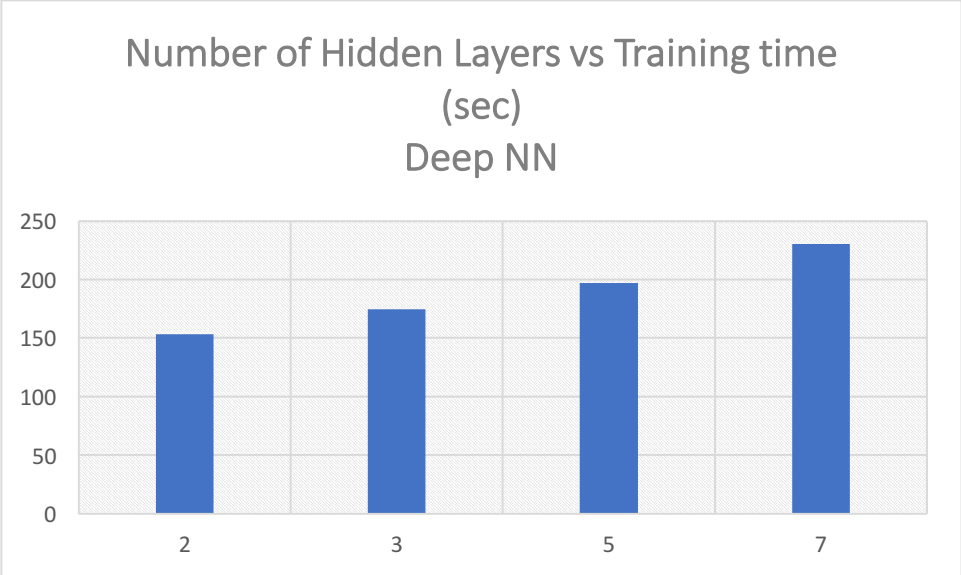


Figure 8: Number of Hidden Layers Vs Training Time in Deep Neural Network
(No. of features per layer = 256)



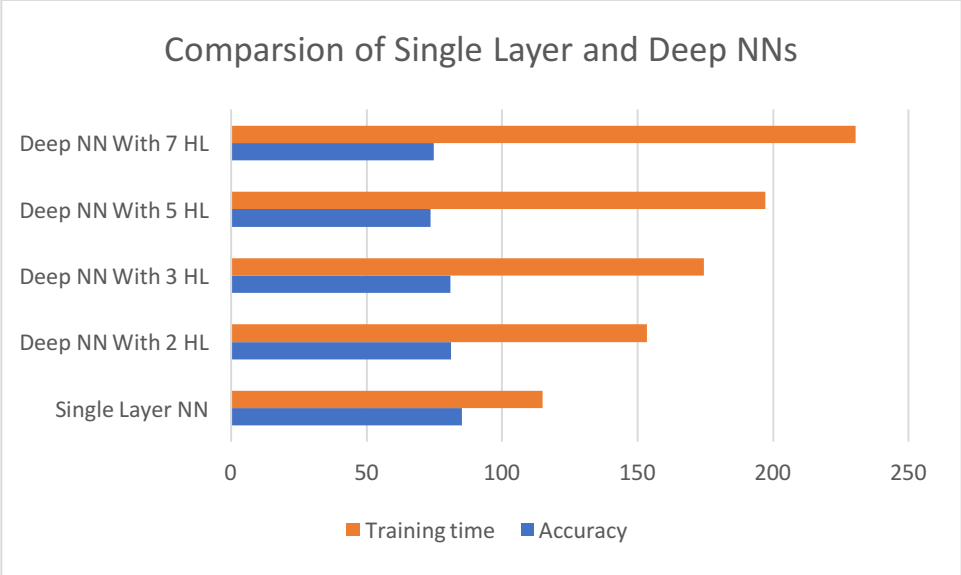
Deep Neural Network Analysis

- Number of hidden layers above 3 begin to cause overfitting (from figure 7)
- Number of hidden layers causes the training time to increase significantly
- Number of features in every hidden layer does not provide a very significant change compared to changing the number of hidden layers itself (Table 3).

Comparison of Single Layer Neural Network vs Deep Neural Network using Tensorflow

Single Layer NN – CelebA Dataset	**Deep NN - CelebA**
Highly efficient for CelebA and MNIST dataset	Cause overfitting if number of hidden layers increase or deeper the network gets
Number of hidden features affects the accuracy greatly	Number of hidden features does affects the accuracy significantly in comparison with increasing the hidden layers (proof: table3)
Training time is below 45 seconds	Training time is above 150 seconds
Best Accuracy is **85.27%** for the CelebA Dataset	Best Accuracy is **81.18%** when number of hidden layers = 2, for CelebA dataset
Works better for the smaller datasets	Will be a better option for larger datasets

Comparison Charts



Convolutional Neural Networks using Tensroflow

Number of convolutional layers = 2
Training Time: 630.67 seconds
Accuracy: 98.7% (9871 / 10000)

Final Confusion Matrix of CNN on MNIST:

```
[[ 975    0    1    0    0    0    1    1    2    0]
 [   1 1128    1    1    0    0    1    2    1    0]
 [   2    2 1012    5    1    0    0    4    6    0]
 [   0    0    0 1004    0    2    0    2    2    0]
 [   1    0    2    0  966    0    1    4    1    7]
 [   1    0    0    7    0  878    3    0    0    3]
 [   8    2    0    1    1    3  941    0    2    0]
 [   0    1    4    3    0    0    0 1015    1    4]
 [   3    0    2    6    0    2    0    4  951    6]
 [   0    3    0    3    1    2    0    3    1  996]]
```

- CNN proves to be very effective with an accuracy above **98.7%,** significantly better than DeepNN and Single Layer NNs.

APPENDIX:
Contains all results and observations for different combinations of Lambda and Hidden nodes

Lambda	No. of Hidden Nodes	Training Set Accuracy	Validation set accuracy	Testing set accuracy	Final Optimization Parameter	Time Taken
0	4	32.108%	30.44%	31.51%	2.3477346986617	44.28
0	8	90.42%	89.98%	90.25%	0.6584262333545	37.85
0	12	91.242%	90.81%	91.2%	0.5907573502290	43.08
0	16	92.916%	92.43%	92.27%	0.4759312832412	48.95
0	20	93.702%	93.05%	93.28%	0.4279432621826	50.31
5	4	63.856%	62.43%	64.12%	1.6753423309480	33.12
5	8	89.05%	88.49%	88.92%	0.8035987000887	38.70
5	12	90.914%	90.47%	89.98%	0.6744545417759	41.00
5	16	93.36%	93.14%	93.21%	0.5113768604655	44.70
5	20	92.99%	62.74%	92.66%	0.5346798266305	50.60
10	4	64.188%	92.46%	63.45%	1.6186787153553	36.80
10	8	89.886%	89.3%	89.59%	0.7979866690212	36.85
10	12	91.828%	91.23%	91.63%	0.6728574649961	40.92
10	16	93.048%	92.28%	92.74%	0.5735220919152	43.60
10	20	93.764%	93.41%	93.14%	0.5354447614383	48.99
15	4	80.186%	79.48%	79.59%	1.2701117346039	39.80
15	8	89.054%	87.97%	89.04%	0.9189362907847	37.02
15	12	91.096%	90.69%	90.87%	0.7867954657656	40.02
15	16	93.078%	92.81%	92.8%	0.6329159014616	46.02
15	20	93.452%	93.03%	93.37%	0.6100967749411	48.18
20	4	69.37%	67.67%	70.1%	1.6841130407761	33.61
20	8	89.776%	88.84%	89.57%	0.9049212859481	39.69
20	12	92.198%	91.58%	91.82%	0.7545112440863	41.92
20	16	92.802%	92.44%	92.4%	0.6866018935851	50.20
20	20	92.838%	92.2%	92.41%	0.6995152865317	64.06
25	4	76.186%	74.92%	76.44%	1.6138850635929	42.15
25	8	90.234%	89.56%	90.04%	0.9706980752338	38.44
25	12	92.028%	91.48%	92.01%	0.7788432994284	40.88
25	16	92.38%	92.09%	92.09%	0.7613645045125	43.11
25	20	92.97%	92.66%	92.98%	0.7068599817226	48.57
30	4	67.434%	66.98%	67.77%	1.8482993652082	33.63
30	8	88.23%	87.4%	87.71%	1.0626775002944	37,32
30	12	91.646%	91.18%	91.48%	0.8473667718838	42.36
30	16	92.806%	92.4%	92.69%	0.7580002204763	46.01
30	20	93.362%	92.9%	93.24%	0.7250093505607	46.18
35	4	49.276%	47.85%	48.24%	2.3330053230228	32.95
35	8	89.482%	89.08%	89.12%	1.0478179375263	37.53
35	12	92.016%	91.93%	91.63%	0.8335093769959	41.19
35	16	93.178%	92.77%	92.72%	0.7820609219785	43.50
35	20	93.344%	92.95%	93.0%	0.7387273611253	47.45
40	4	74.676%	73.67%	74.79%	1.7036624290192	31.93
40	8	89.514%	88.67%	89.95%	1.0713278422308	36.11

40	12	91.61%	91.27%	91.44%	0.9156486257094	41.26
40	16	92.696%	92.26%	92.55%	0.8123838791856	44.33
40	20	93.03%	92.84%	93.0%	0.7940607512691	49.21
45	4	55.668%	54.14%	55.56%	2.2750081806838	36.11
45	8	89.314%	88.86%	89.01%	1.0961955678581	35.06
45	12	91.158%	90.98%	91.31%	0.9612539288719	38.43
45	16	91.844%	91.54%	92.01%	0.8838786762879	45.54
45	20	92.758%	92.63%	92.7%	0.8235047057082	48.85
50	4	62.124%	61.38%	62.31%	1.9074033209988	34.87
50	8	89.424%	89.01%	89.36%	1.0945586595373	37.62
50	12	91.492%	91.38%	91.33%	0.9425001713562	41.14
50	16	92.348%	92.19%	92.41%	0.8813811195920	44.81
50	20	92.626%	92.42%	92.51%	0.8650937653819	47.73
55	4	74.584%	74.0%	73.44%	1.7459059067174	32.58
55	8	89.276%	89.24%	89.44%	1.1900759745216	37.10
55	12	92.23%	91.92%	92.14%	0.9352062711769	41.60
55	16	92.076%	91.79%	92.06%	0.9367257219716	45.29
55	20	92.53%	92.52%	92.5%	0.8874585117592	49.25
60	4	67.728%	65.96%	67.62%	1.8748121272003	35.66
60	8	89.064%	88.63%	88.96%	1.2057961310909	37.17
60	12	91.734%	91.4%	91.67%	1.0119829915763	41.55
60	16	92.5%	92.18%	92.56%	0.9143257063503	45.33
60	20	92.574%	92.48%	92.57%	0.9028820317320	48.18