CSE 574: INTRODUCTION TO MACHINE LEARNING

PROGRAMMING ASSIGNMENT – 2:

HANDWRITTEN DIGITS CLASSIFICATION

TA: Jie Zhang

Team Members (Group – 40):

50413342 - Bodhith Edara
50409091 - Teja Reddy Alla
50417093 - Venkata Krishna Sai Sakhamuri

CHOOSING THE HYPER-PARAMETERS FOR THE NEURAL NETWORK

- We can choose the hyper-parameters by producing a comparison between the accuracy obtained and the time taken by the neural network to complete its computation. Here we try to vary the values of hidden units with constant lambda values and run the neural network program.

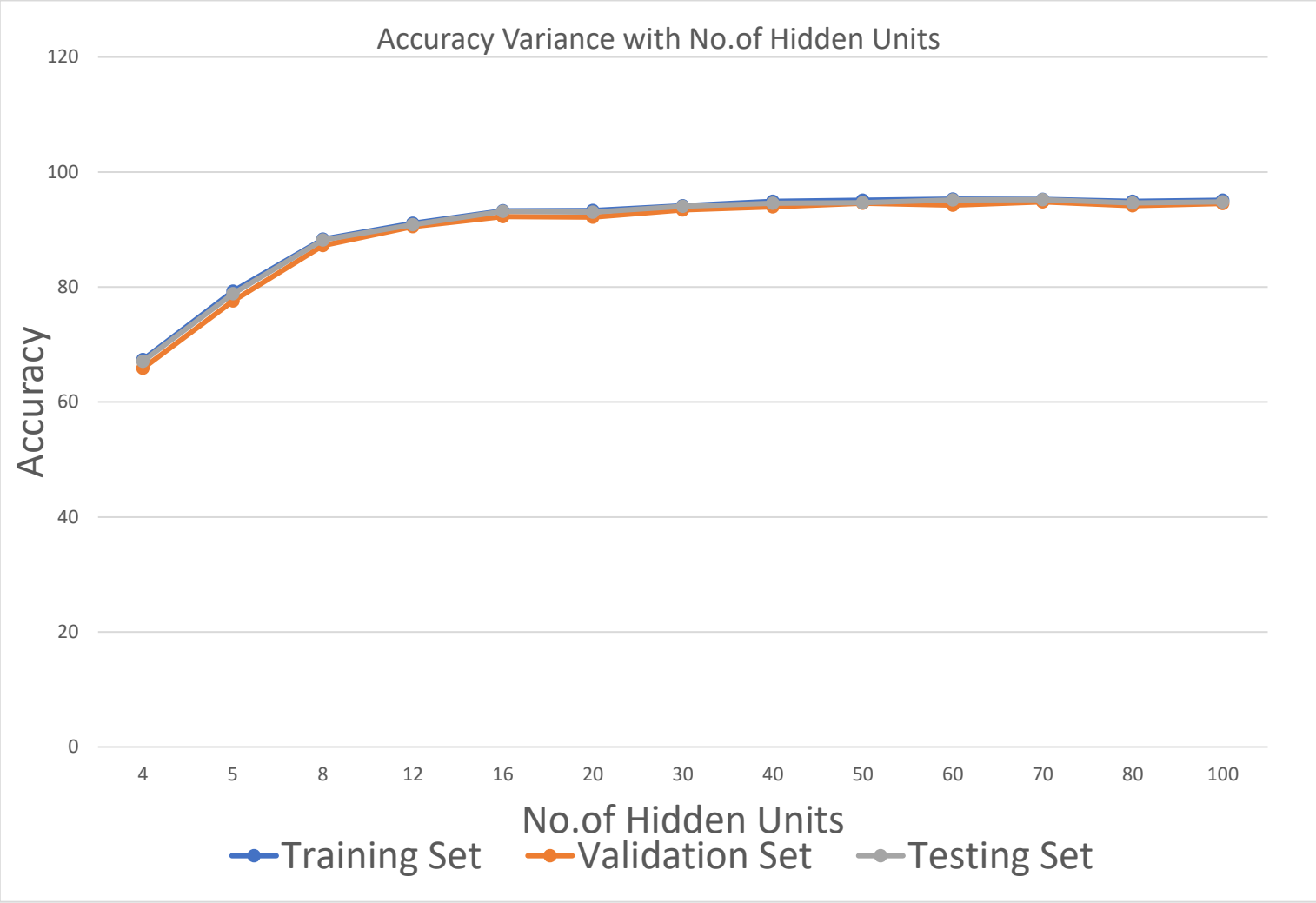
VARYING THE NUMBER OF HIDDEN UNITS WITH CONSTANT LAMBDA:

- We chose the Lambda value 20 and varied the number of hidden units.
- The nnScript is ran by using the hidden nodes values {4,5,8,12,16,20,30,40,50,60,70,80,100} while keeping the lambda(20) as a constant and the accuracies for training, validation and testing sets are recorded.
- These observations will help up in choosing the optimal values for the hyper-parameters (no. of hidden units and λ)

Comparing accuracies by varying the no. of hidden units:

Hidden units	Lambda	Training Set Accuracy	Validation Set Accuracy	Testing Set Accuracy	Time taken (sec)
4	20	67.368	65.869	67.09	33.0064
5	20	79.278	77.59	78.83	42.20534468
8	20	88.3079	87.2	88.17	39.58004
12	20	91.076	90.51	90.84	41.688
16	20	93.266	92.23	93.15	52.055
20	20	93.32	92.17	93.03	52.914
30	20	94.148	93.4	94.05	50.851
40	20	94.896	93.96	94.59	57.9867
50	20	95.102	94.55	94.67	61.862
60	20	95.286	94.24	95.13	59.3403
70	20	95.244	94.8	95.179	65.1831
80	20	94.92	94.13	94.63	63.879
100	20	95.082	94.51	94.82	72.306

Plot:



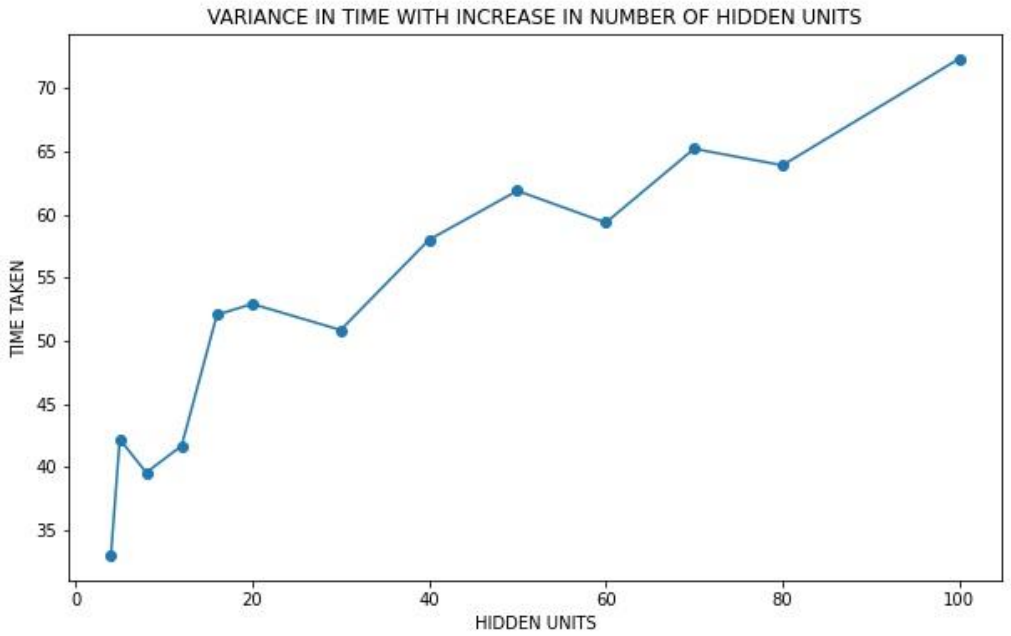
Observations:

- As the number of hidden layers increases, the accuracies (test, train and validation sets) also increased.
- There is a drastic increase when the hidden node values are change from 4 to 12 but from 12 to 100, the change in the accuracies is not drastic enough as there is convergence of the objective function.
- So, the optimal values are:

  Lambda : 15

  No. of hidden units : 60

Plot:



Observations:

- It is evident from the above graph that the time taken for different hidden units is different and we may say the overall trend is increasing with the no. of hidden units.
- During some instances, there is an increase in time taken with increase in the number of hidden nodes while some other times, the opposite is true.

Inferences:

We can conclude that the neural network performs best when the number of hidden nodes is 60 and the accuracies obtained are the proof for the same. There is some issue when it comes to time taken by the neural network but we gave more importance to the accuracies obtained than the time consumed.

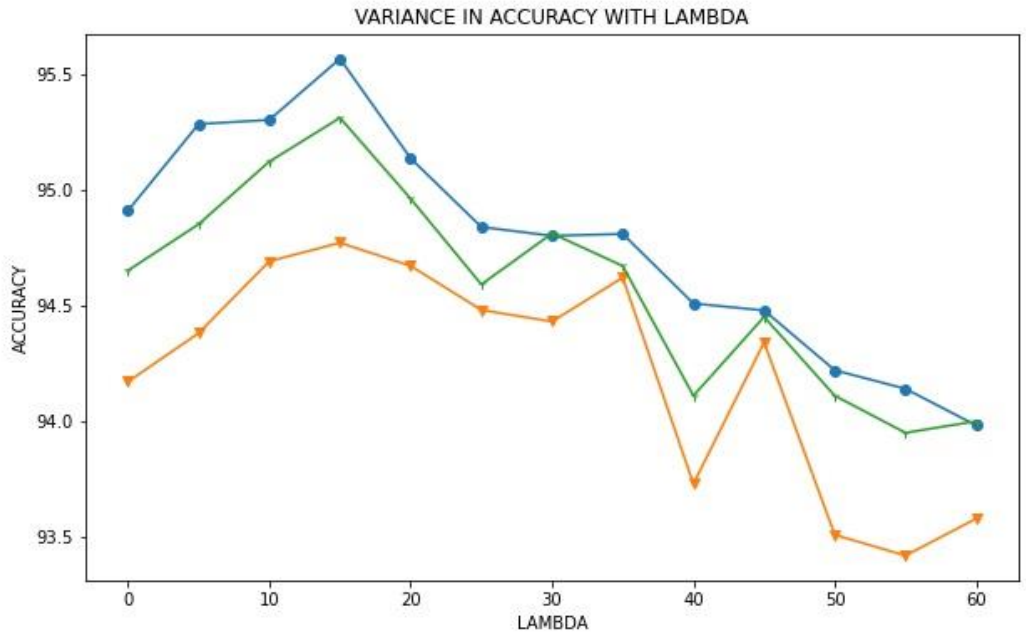
VARYING THE LAMBDA VALUES KEEPING THE HIDDEN UNITS AS CONSTANT:

Comparing accuracies by varying the lambda values

Hidden units	Lambda	Training set Accuracy	Validation set Accuracy	Testing set Accuracy	Time taken (sec)
60	0	94.908	94.17	94.65	59.946585
60	5	95.282	94.38	94.85	64.674703
60	10	95.3	94.69	95.12	60.63168
60	15	95.564	94.77	95.309	62.428
60	20	95.134	94.67	94.96	61.737
60	25	94.838	94.48	94.59	62.04474
60	30	94.8	94.43	94.81	60.841337
60	35	94.808	94.62	94.67	61.93472
60	40	94.508	93.73	94.11	60.367
60	45	94.478	94.34	94.45	62.396157
60	50	94.22	93.51	94.11	76.839755
60	55	94.14	93.42	93.95	63.559651
60	60	93.982	93.58	94.0	66.0594

- We chose the number of hidden units to be 60 and varied the lambda values
- The nnScript is ran by using the lambda values {0,5,10,15,20,25,30,35,40,45,50,55,60} while keeping the hidden units (60) as a constant and the accuracies for training, validation and testing sets are recorded.

PLOT:

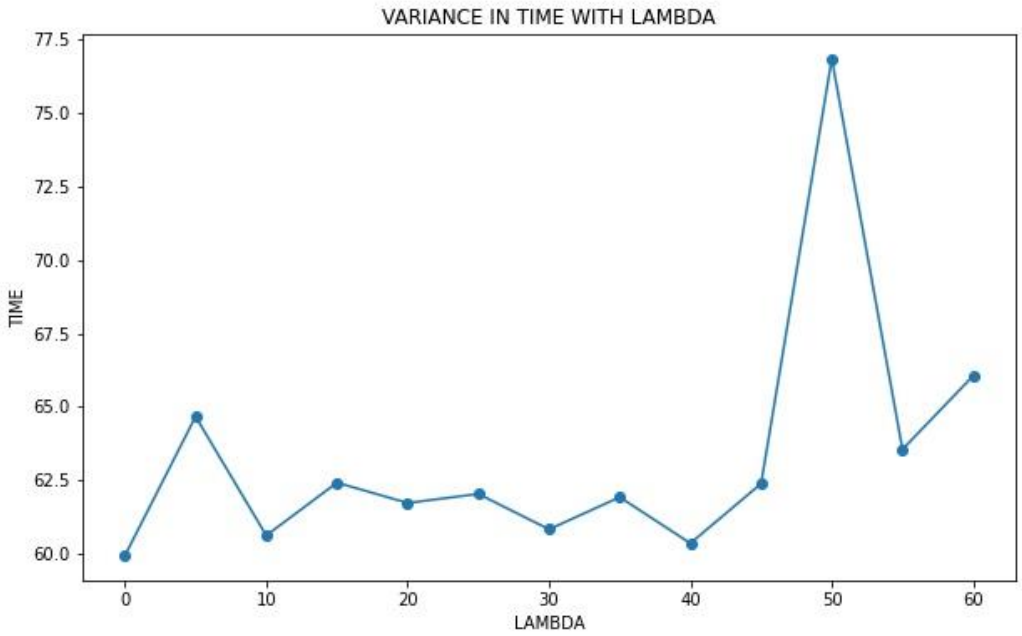


Observations:

We can observe the accuracies obtained for different values of lambda from the above graph and we need to deduce the value of lambda for which the accuracies are the largest.

In order to regularize the outcome and try not to overfit the training data, we use the lambda which is a regularization coefficient and avoid any unforeseen issues while running the neural network on the testing dataset.

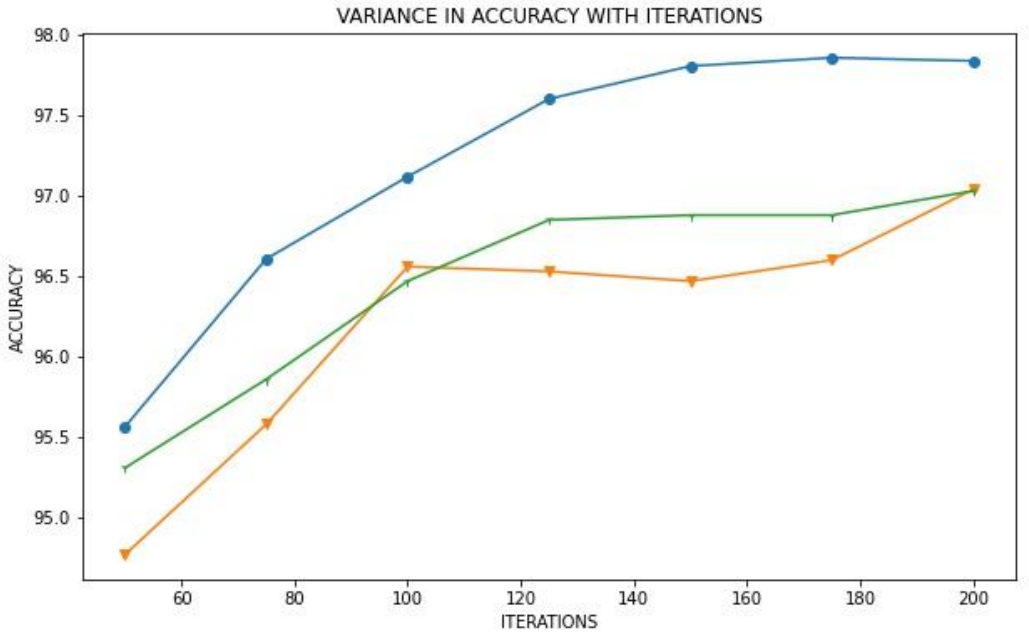
PLOT:

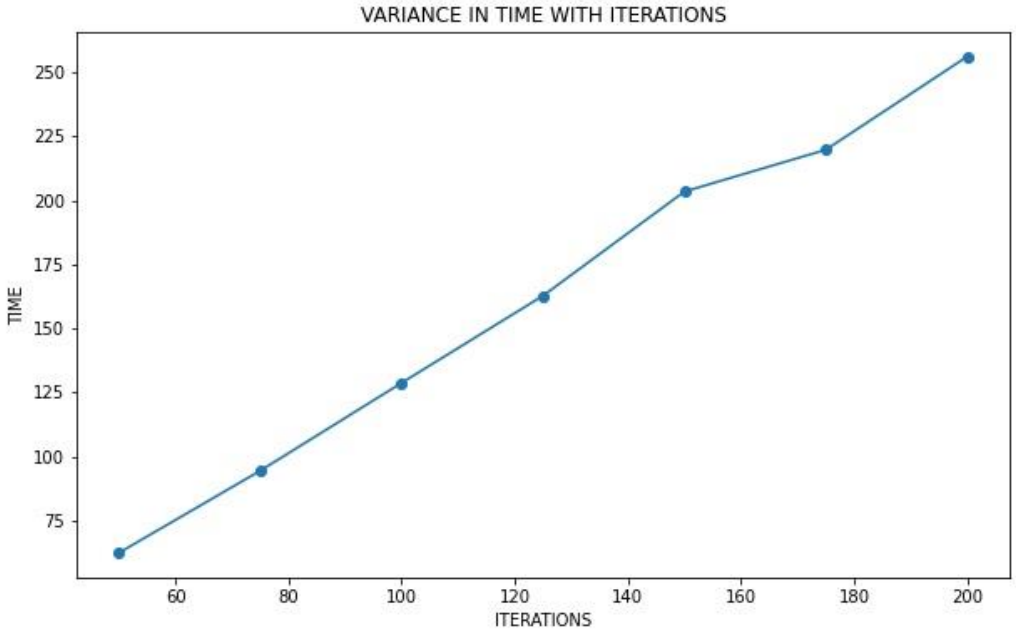


As the value of the lambda is increased initially the execution time required is almost flat initially then we could see a spike upwards in the time required.

VARIANCE IN THE ACCURACIES WITH INCREASE IN ITERATIONS:

Iterations	Training set Accuracy	Validation set Accuracy	Testing set Accuracy	Time taken (sec)
50	95.564	94.77	95.309	62.428
75	96.608	95.58	95.86	94.4884
100	97.119	96.56	96.47	128.79514
125	97.602	96.53	96.85	162.811
150	97.806	96.47	96.88	203.4821
175	97.858	96.6	96.88	219.801
200	97.838	97.04	97.03	256.19683





Observations :

- As the table below indicates we can understand that as the number of iterations is increased the accuracy of the neural networks also keeps increasing. But, after the iterations are increased over a certain point the accuracy starts to stagnate.
- As we increase the number of iterations the time taken to build the model also keeps increasing.

After trying different combinations of hyper parameters, the optimal values that gave us the highest accuracies are given below:

Hyper-Parameters	Default Values	Optimal Values
Hidden Units	50	60
Regularization Parameter(lambda)	0	15
Number of Iterations	50	200
Accuracy	Training Set - 94.30% Validation Set - 93.40% Test Set - 94.10%	Training Set -97.838% Validation Set - 97.04% Test Set - 97.03%
Time	62.699	59.3403

Conclusion:

- The optimal lambda value that gave us the highest accuracy is 15.
- When we tried to increase the value of lambda there was a reduce in accuracy because the model keeps under fitting the data.
- The optimal value for the hidden units is 60.

2: THE COMPARISION OF DEEP NEURAL NETWORKS AND NEURAL NETWORKS

Aim: To train and evaluate the neural network models with one hidden layer and multiple hidden layers on CelebA data set.

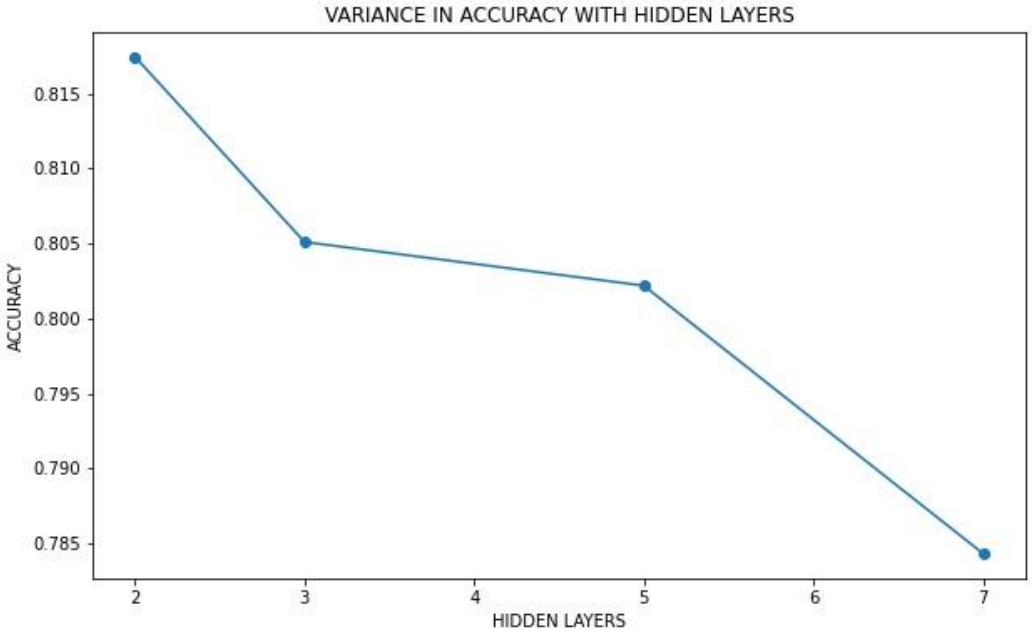
- We analyse the results after running the script on CelebA dataset using single layer neural network and deep neural network(multi-layer) and tried to understand the optimal values obtained by each model.
- The number of layers used for deepnn are {2,3,5,7} and ran the script on test data set.

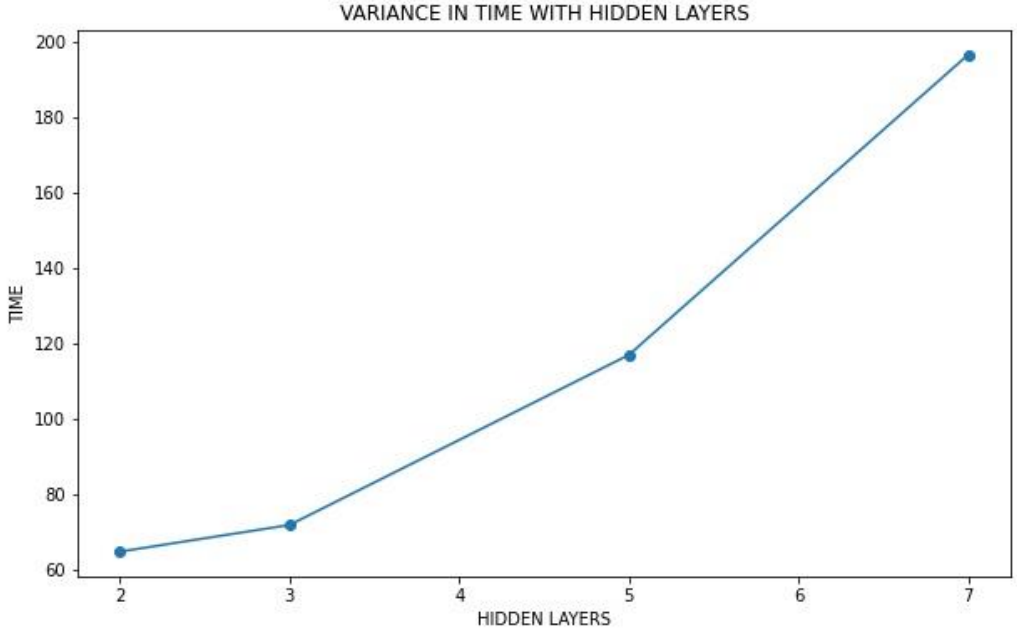
FaceNN: Results obtained by executing the faceNN script on celebA dataset are:

Hyper-Parameters	Default Values	Optimal Values
Hidden Units	256	80
Regularization Parameter	10	20
Accuracy	Training Set - 84.071% Validation Set - 83.564% Test Set - 84.352%	Training Set - 85.274% Validation Set - 84.015% Test Set - 85.93%
Time	72.699	67.134

DeepNN: Results obtained by executing the deepNN script on celebA dataset are:

No. of Hidden Layers	Accuracy	Time
2	0.817468	64.65
3	0.805108	81.73
5	0.8021931	116.86
7	0.784312	196.54


VARIANCE IN ACCURACY WITH HIDDEN LAYERS
ACCURACY
0.815
0.810
0.805
0.800
0.795
0.790
0.785
2
3
4
5
6
7
HIDDEN LAYERS


VARIANCE IN TIME WITH HIDDEN LAYERS
TIME
200
180
160
140
120
100
80
60
2
3
4
5
6
7
HIDDEN LAYERS

Inference:

- The accuracy of the neural network with a single layer is better than that of the deep neural network.
- As evident from the graphs and observed data, the time taken for the script to execute increases with increase in number of hidden layers. This is because of increase in complexity of the model used. As a result, the execution time increases with increase in complexity of the model.
- And as the number of hidden layers increases, the accuracy of the model decreased. This trend is due to concept of over-fitting and as a result we obtain an increasing trend when observing validation and training accuracies but it is opposite when it comes to testing accuracies.

3. CONVOLUTIONAL NEURAL NETWORK

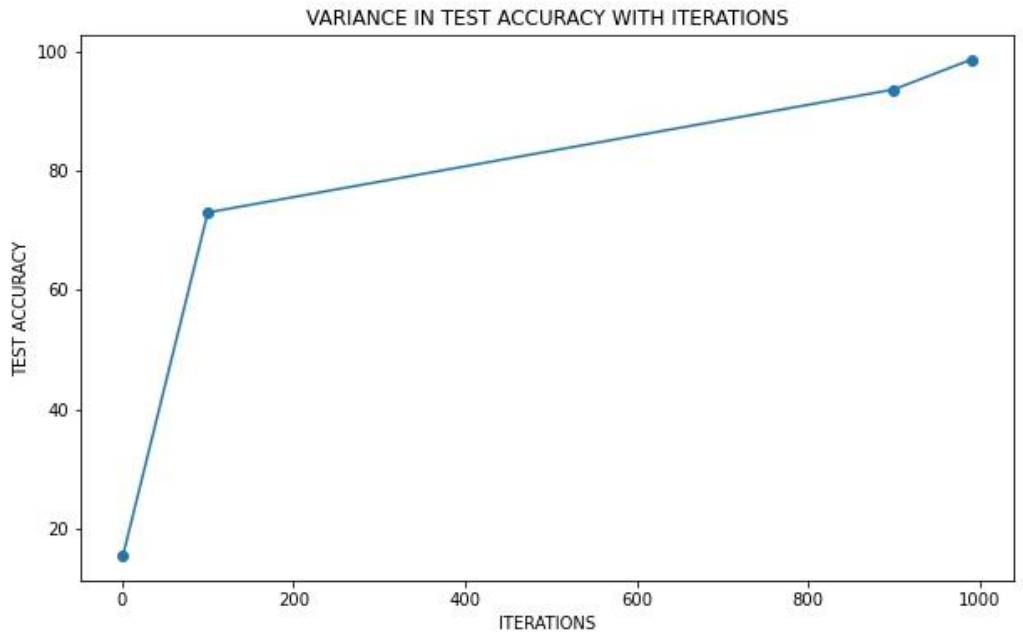
We try to gauze the variance in accuracies and time with change in no. of iterations and plotted the same using the confusion matrix.

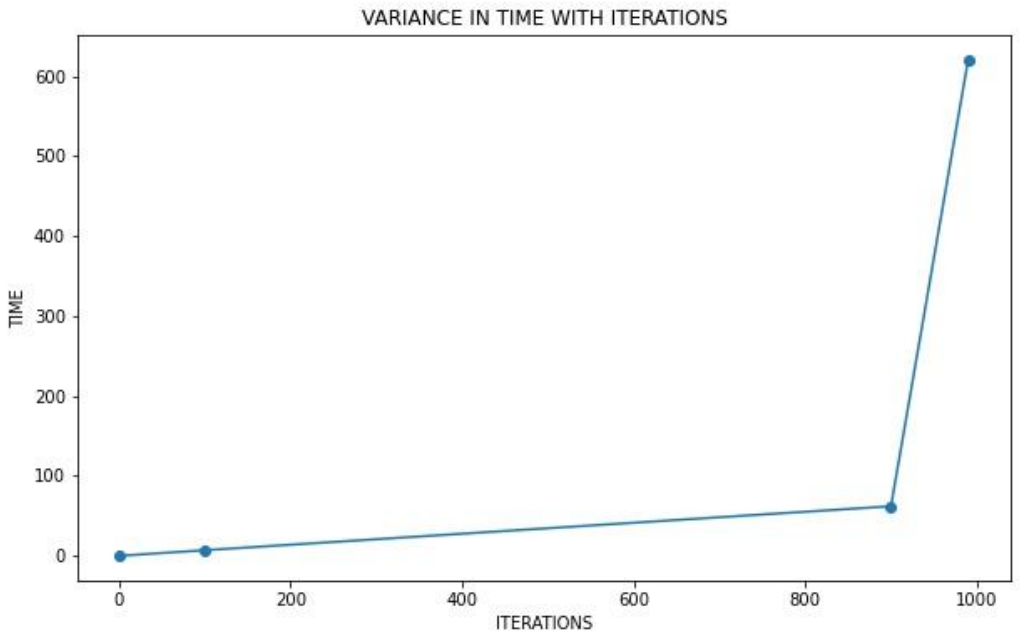
The results obtained and the confusion matrix obtained are as shown below:

Table:

Iterations	Test set Accuracy (%)	Time (in secs)
1	15.5	0
100	73	7
900	93.6	62
9900	98.6	621

Plots:



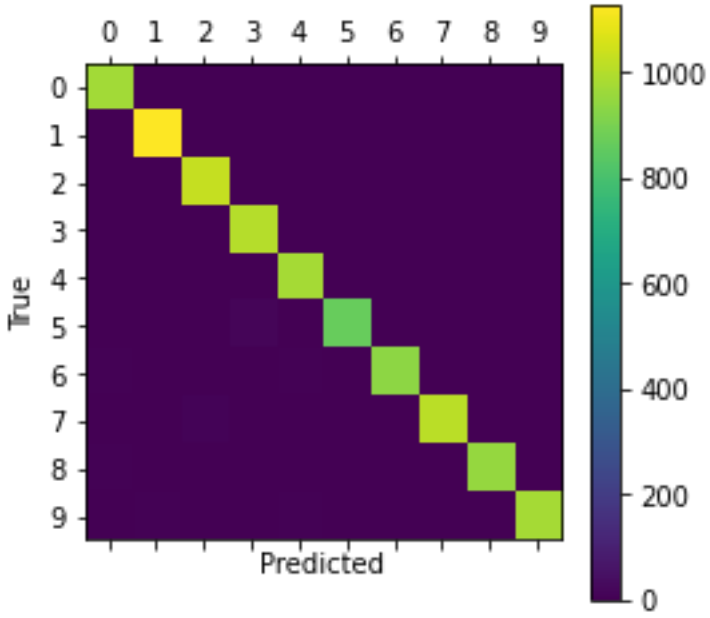


- Confusion matrix compares the target values with the values predicted by the machine learning model.

The confusion matrix is as follows:

Confusion Matrix:

```
[[ 975    0    1    0    0    1    1    1    1    0]
 [   0 1130    3    0    0    1    1    0    0    0]
 [   0    0 1029    0    0    0    0    3    0    0]
 [   1    0    1 1003    0    0    0    1    2    2]
 [   0    0    3    0  978    0    0    1    0    0]
 [   2    1    0   14    0  868    3    1    1    2]
 [   6    3    1    1    5    4  938    0    0    0]
 [   0    1   10    3    0    0    0 1011    1    2]
 [   6    1    4    4    3    1    0    2  951    2]
 [   4    5    2    4    8    4    0    3    2  977]]
```



Observations:

- As we increase the number of iterations, there is:
  - An increase in time taken,
  - Increase in the accuracy obtained,
  - Decrease in errors, and
  - Increase in correctness of the prediction
- The **accuracy obtained is 98.6 %** and
- The **time** taken is **10 minutes and 21 seconds**
- Accuracy is very high due to the presence of filter matrix that input image can be divided into parts and combine with the same dimension's filter so that there is flexibility in changing the image at each iteration in the computation.

Inference:

CNN is far superior in terms of accuracy obtained when compared to single layer neural network and deep neural network as hyper-parameters are tuned based on the image and prior knowledge obtained by feature selection.