CSE 574 INTRODUCTION TO MACHINE LEARNING
PROGRAMMING ASSIGNMENT 2
HANDWRITTEN DIGITS CLASSIFICATION

TEAM MEMBERS

KEERTHANA KANNAN (50248873)

HIREN ARAVIND MURUGUDU GOVINDARAJAN (50248817)

1 NEURAL NETWORK

The task given was to create a neural network by implementing the forward pass and the back propagation. Also feature selection and regularization of the weights (λ) was performed. There are totally 3 layers in the neural network, first layer comprises of (d+1) units, each representing a feature of image, the second layer is called the hidden units which is considered as the learned features extracted from the original data set and the third layer called the output layer is the probability of a certain handwritten image belonging to a particular digit.

Feature Selection: In the data set, there are many features which values are exactly sane for all data points in the training set and hence the classification model cannot gain any information about the difference between the data points. Hence these features are ignored in preprocess(). The number of features after reduction was observed to be 716.

Feedforward Propagation: Using the given parameters of Neural Network and feature vector x, the probability that this feature vector belongs to a particular digit is computed. The formulas implemented in the code are,

$$a_j = \sum_{p=1}^{d+1} w_{jp}^{(1)} x_p \qquad b_l = \sum_{j=1}^{m+1} w_{lj}^{(2)} z_j$$

$$z_j = \sigma(a_j) = \frac{1}{1+\exp(-a_j)} \qquad o_l = \sigma(b_l) = \frac{1}{1+\exp(-b_l)}$$

Backpropagation: In this, the error function is calculated and then it is sent backward to update the weights so that the error is reduced further.

$$J(W^{(1)}, W^{(2)}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{\kappa}(y_{il}\ln o_{il} + (1-y_{il})\ln(1-o_{il}))$$

Then the weight is updated as,

$$w^{new} = w^{old} - \gamma \nabla J(w^{old})$$

Regularization: Overfitting is when the model trains the training data too well such that the model gives poor generalization when testing with validation data. So we add a regularization coefficient to the term.

$$\widetilde{J}(W^{(1)}, W^{(2)}) = J(W^{(1)}, W^{(2)}) + \frac{\lambda}{2n}\left(\sum_{j=1}^{m}\sum_{p=1}^{d+1}(w_{jp}^{(1)})^2 + \sum_{l=1}^{k}\sum_{j=1}^{m+1}(w_{lj}^{(2)})^2\right)$$

$$\frac{\partial \widetilde{J}}{\partial w_{lj}^{(2)}} = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{\partial J_i}{\partial w_{lj}^{(2)}} + \lambda w_{lj}^{(2)}\right)$$

$$\frac{\partial \widetilde{J}}{\partial w_{jp}^{(1)}} = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{\partial J_i}{\partial w_{jp}^{(1)}} + \lambda w_{jp}^{(1)}\right)$$

2 APPROACH

We performed several simulations to determine how the network parameters (λ and number of hidden nodes) impact the accuracy and training time. In this we classify the handwritten digits using the MNIST dataset.

We started with lamda value 0 and varied lamda from 0 to 60, in increments of 10. During this the number of hidden nodes was 50 (given). The number of hidden nodes were then set to 4 and step by step incremented as 4,8,12,16,20,30,40,50 while continuing to vary lambda in same pattern. The training, validation and test data accuracy and the training time for all the cases were obtained and plotted them against different network parameters. Based on the best performance of the network the hyper parameter was decided.

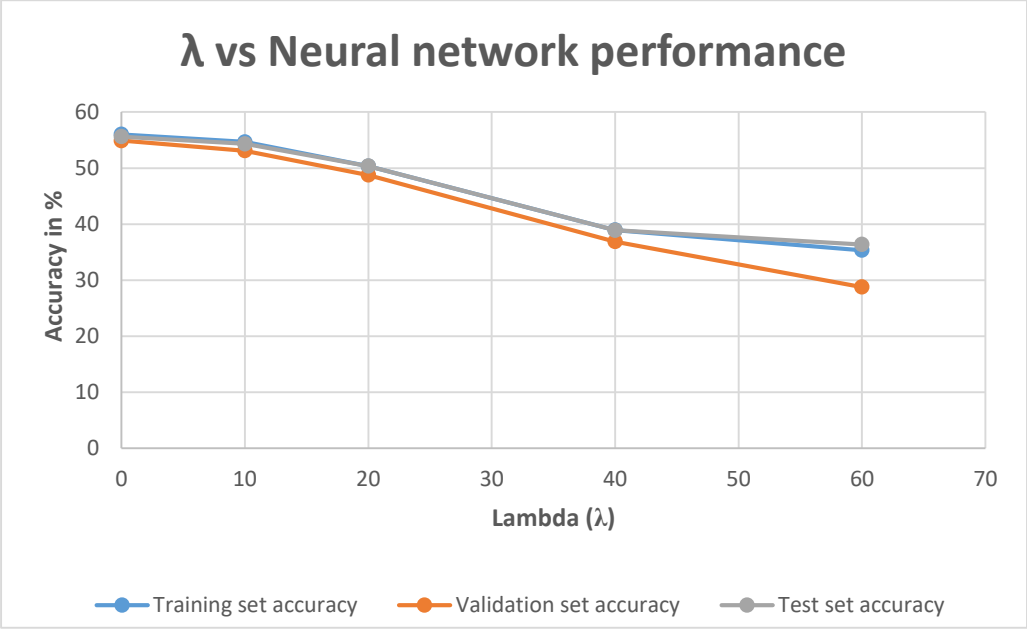
3 HYPER PARAMETERS FOR THE NEURAL NETWORK

The parameters considered here are the regularization coefficient (λ) and the number of hidden nodes. Then parameter giving higher accuracy and lesser training time is considered to be the best fit for the network.

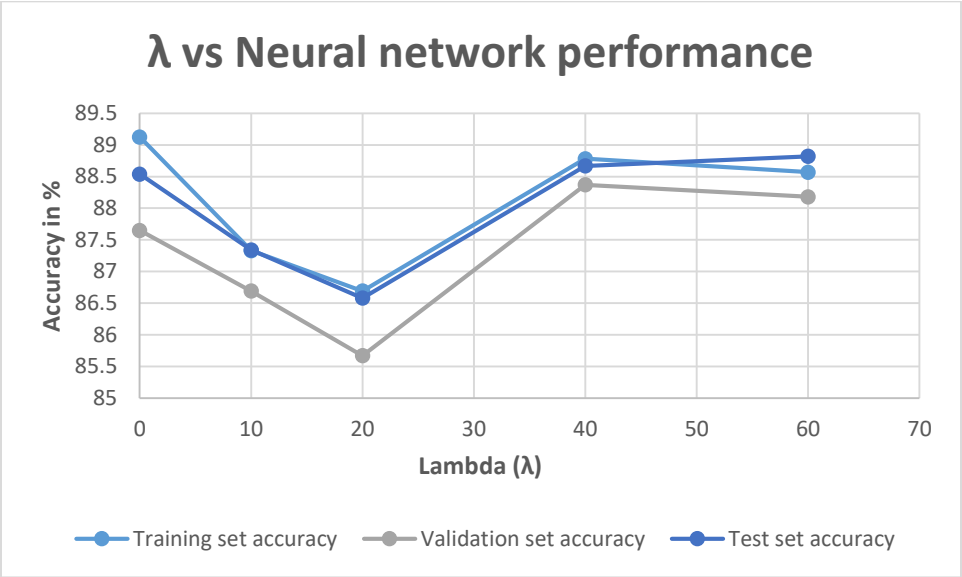
- **λ vs Network performance**

  The graphs below show the performance of the network while having the number of hidden nodes constant and varying the value of lambda. Likewise, it is done for different values of the hidden nodes. With increase in lambda values, the training set accuracy decrease and test set accuracy increase is expected but this is not the same for all the cases because of the similarity in the test and training set datas. Based the graphs it can be concluded that due to change in the lambda value (habing number of hidden nodes constant) there is no significant improvement in the performance.

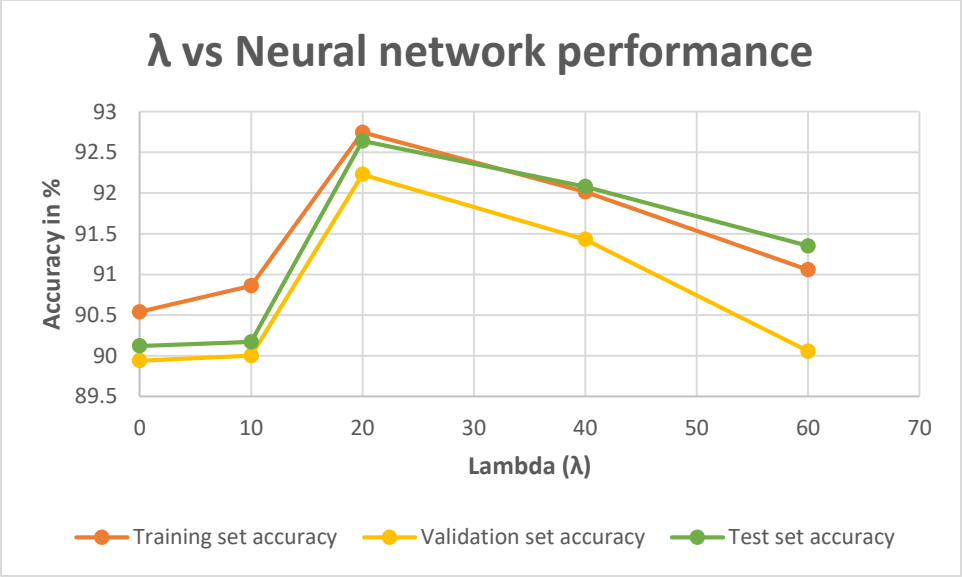
  For hidden node = 4,



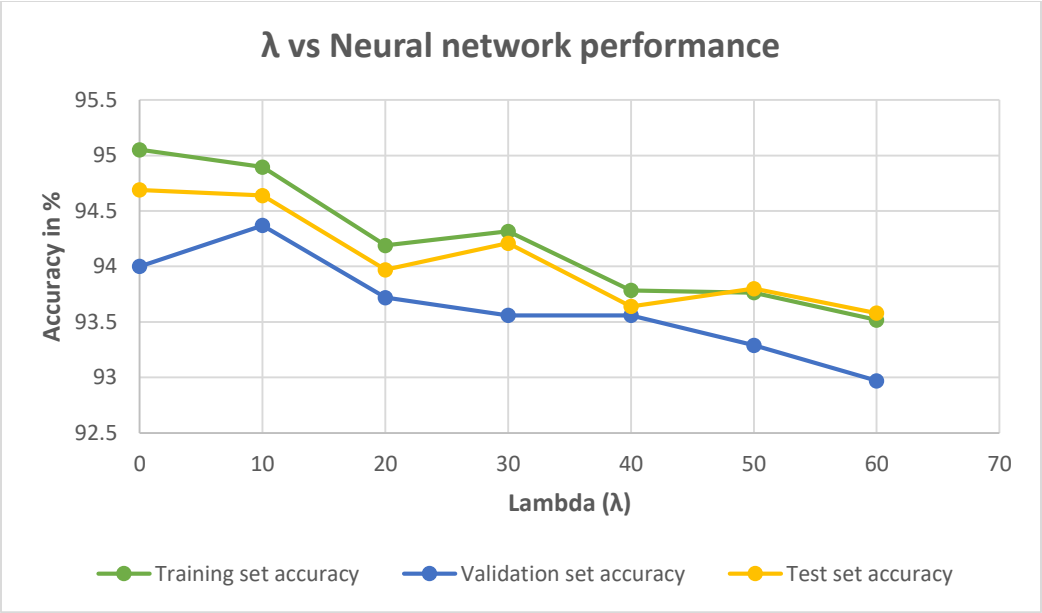
For hidden node = 12,



For hidden node = 20,

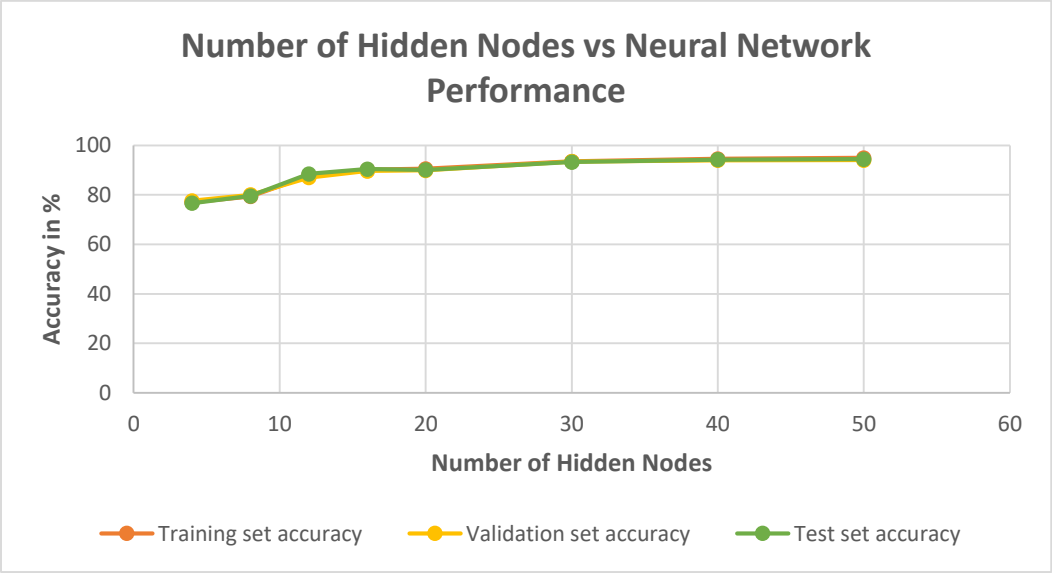


For hidden node = 50,



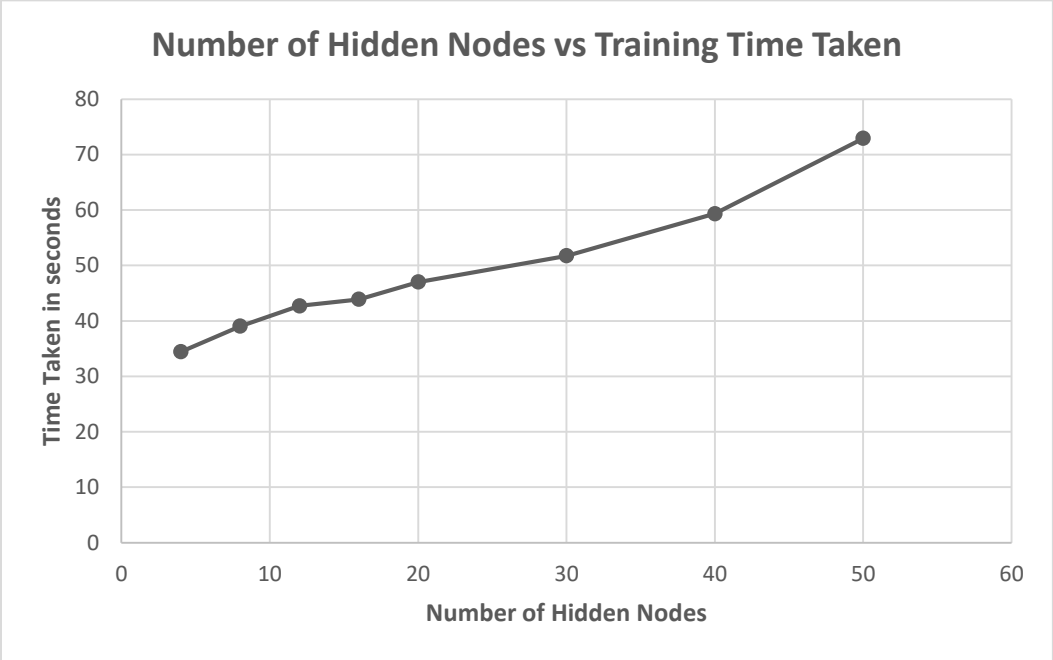
- **Number of hidden nodes vs Network performance**

We observe that as the number of nodes increase, the accuracy of the network also increase to a certain extent and for higher value values of the hidden nodes there is no much change in the accuracy of the network and hence it is observed like a constant line after 20. The lambda value is set to be constantly 10.



- **Number of hidden nodes vs Training time**

  It can be noticed that as the hidden nodes is increased the training time is also increased. This is because when there are more hidden nodes, then there are more weights and gradients to be computed. Hence increasing the complexity of the network and also takes more time to bring a solution.



- **Hyper- Parameters**

  From the graphs shown above, different combinations of hidden nodes and λ were tried to determine the optimal values. Hence the following conclusions can be drawn,

  If training time is not considered, the optimal number of hidden nodes is 20 because even if the value of hidden nodes are increased beyond that, there is no significant change in the accuracy.

  If training time is considered, 12 to 16 hidden nodes seem to look good giving lesser training time and satisfactory amount of accuracy.

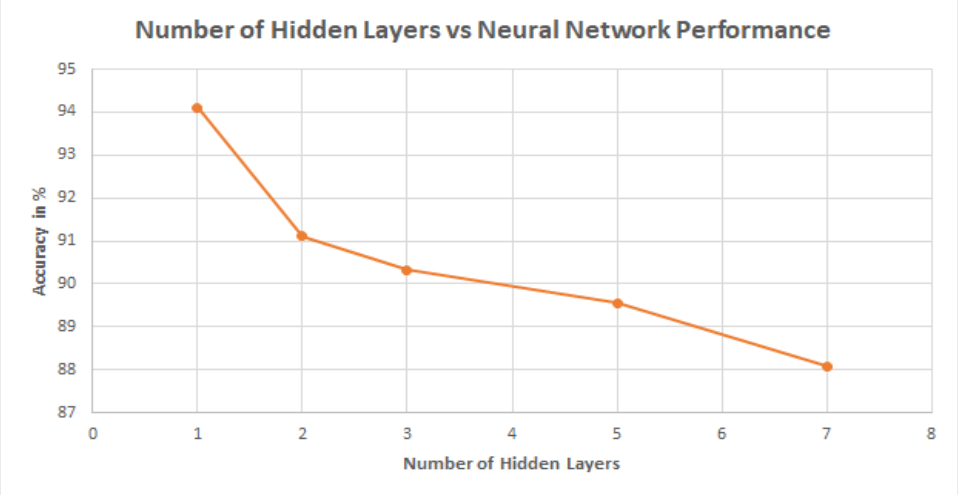
  The optimal value for lambda changes for each set of hidden nodes and the progress of it is also uncertain. Hence by analysing the network with 50 hidden nodes, the good value of lambda is between 0 to 5 and the optimal value being 5.

4 DEEP NEURAL NETWORK

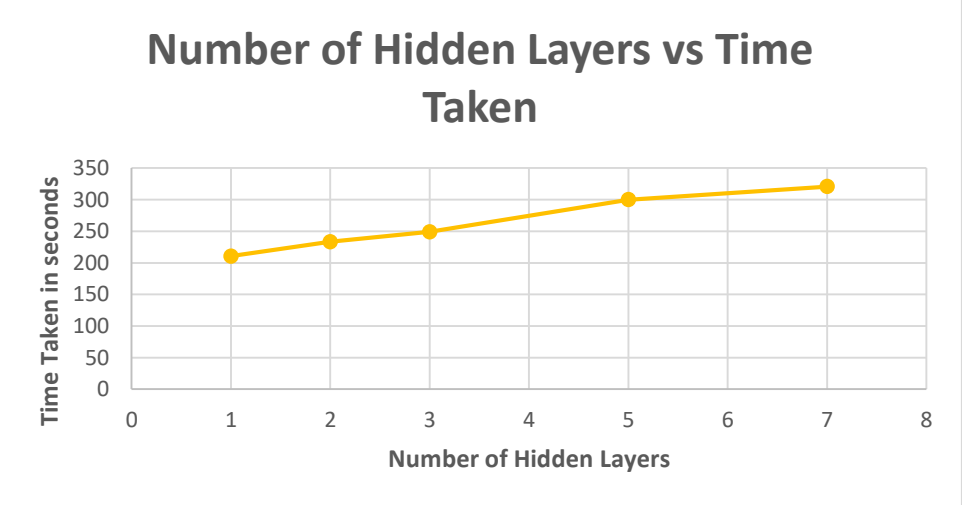
Deep Neural Network is the neural network with multiple hidden layers between the input and the output layers. Deep neural network is compared with the single layer neural network based on the accuracy and the training time. In this we classify whether the person in the image is wearing glasses or not by using Celeb A dataset.

➢ Single Layer vs Deep Neural Network

- Number of hidden layers vs Accuracy
  It is expected that the accuracy of the deep neural network should be greater than the single layer network. But here the accuracy of the neural network is observed to be decreasing as the number of hidden layers are increased. The maximum accuracy is observed in single layer neural network because of the regularization of the parameters and hence avoiding the problem of overfitting. But for deep neural network there is problem of overfitting as we keep adding layers and hence the decrease in accuracy.



- **Number of hidden layers vs Training Time**
  As the number of hidden layer is increased the training time taken is also increased because more time is taken in calculating the weight, bias and error for each layer.



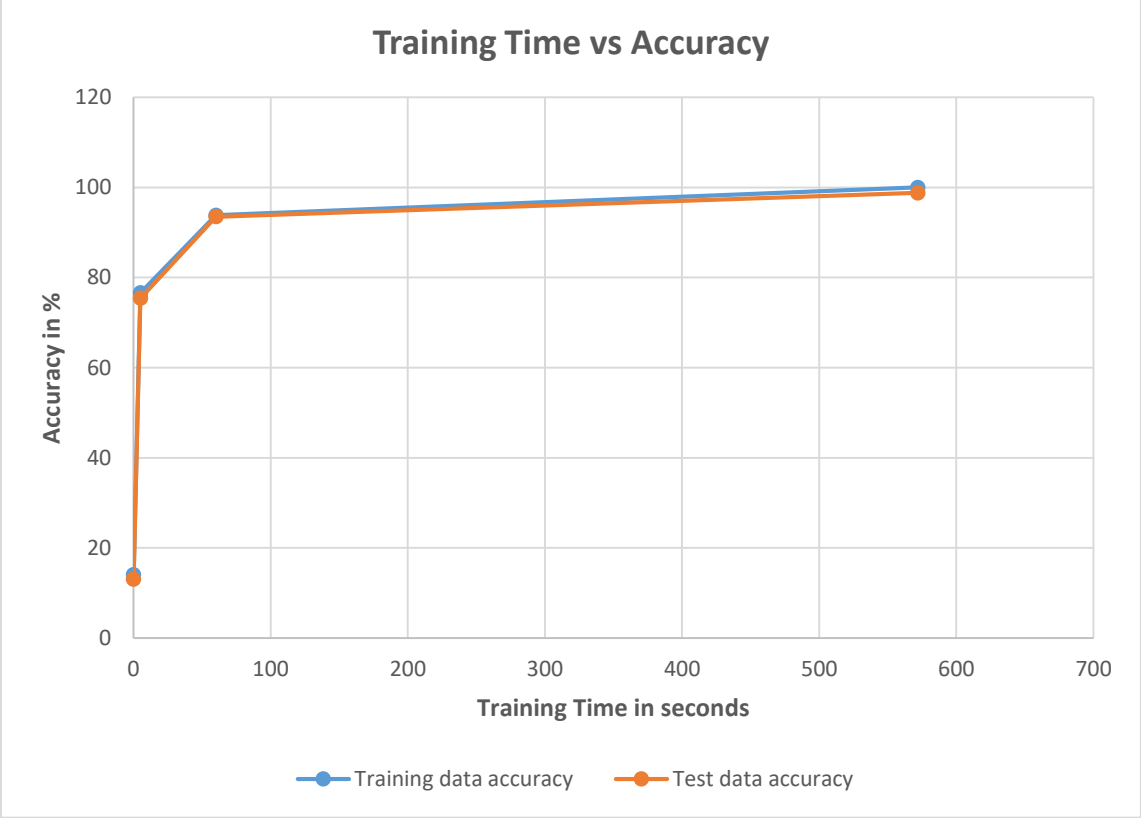
5 CONVOLUTIONAL NEURAL NETWORK

In the convoluted neural network, we are using ReLU neurons and initialized them with a positive value to avoid “dead neurons” issue. Our convolutions use a stride of one and are zero padded so that the output is the same size as the input. We are also using max pooling over 2 x 2 blocks. Our first convolutional layer will compute 32 features for each 5x5 patch. We reshape x to a 4d tensor, with the second and third dimensions corresponding to image width and height, and the final dimension corresponding to the number of color channels. In order to build a deep network, we stack several layers of this type. The second layer will have 64 features for each 5x5 patch.

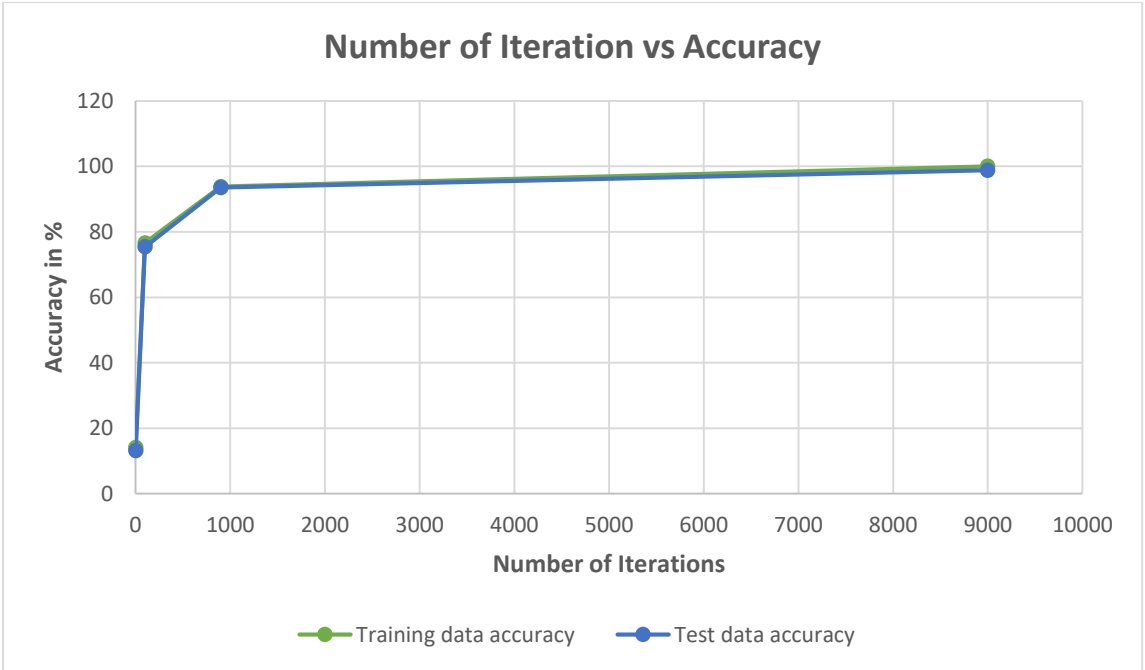
Here we have plotted the accuracy of the model at different training times and we can see the accuracy has increased rapidly for the first 50 seconds and then gradually maintain to a certain level after that.

Also the accuracy keeps increasing after each iteration as the system trains the data for each iteration and the training time obviously increases for the set of iterations.

Finally we have shown the example errors and the confusion matrix after 1,99,900,9000 iterations. This aids to get a clear view of the accuracy of the model.



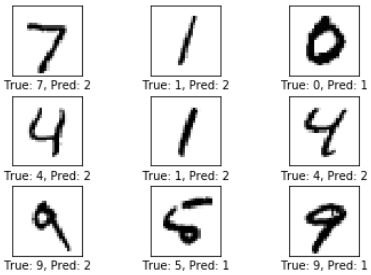

Number of Iterations vs Training Time
Training Time in seconds
700
600
500
400
300
200
100
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Number of Iterations


Number of Iteration vs Accuracy
Accuracy in %
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Number of Iterations
Training data accuracy
Test data accuracy

- **Confusion Matrix and Example Errors of the test data**

After iteration 1,

Example errors:



Confusion Matrix:

[[0 694 284 0 0 2 0 0 0 0]

[0 362 770 0 0 1 0 0 0 2]

[9 76 946 0 0 1 0 0 0 0]

[6 75 904 0 0 0 0 0 0 25]

[0 93 886 0 0 3 0 0 0 0]

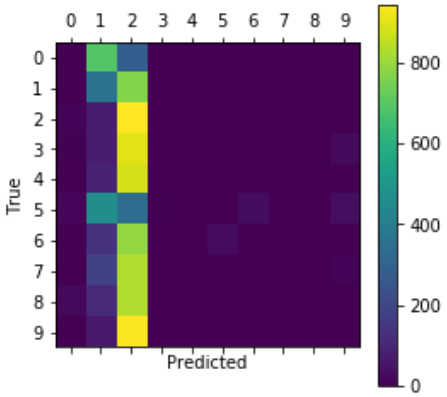
[13 467 337 6 2 0 32 0 0 35]

[0 135 792 0 0 29 2 0 0 0]

[0 185 832 0 0 0 0 0 2 9]

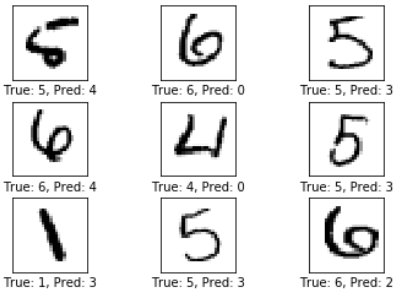
[21 117 832 0 0 3 1 0 0 0]

[0 69 939 0 0 0 1 0 0 0]]



After iteration 99,

Example errors:



Confusion Matrix:

[[955 6 0 7 2 1 2 1 2 4]

[0 1107 0 8 0 0 3 0 0 17]

[58 46 826 35 34 0 11 14 7 1]

[44 40 17 842 4 2 1 18 10 32]

[2 16 2 0 787 0 13 2 0 160]

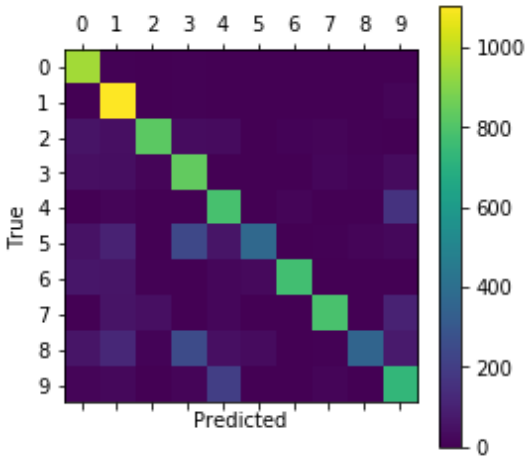
[53 106 3 242 63 376 2 7 16 24]

[66 63 6 2 18 29 772 0 0 2]

[3 60 46 4 19 0 0 788 2 106]

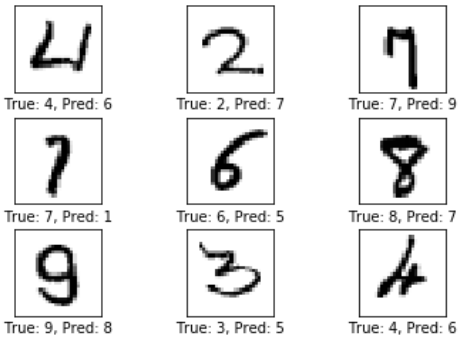
[64 123 11 251 46 32 2 6 358 81]

[14 25 3 15 205 1 0 14 1 731]]



After iteration 900,

Example errors:



Confusion Matrix:

[[963 0 1 0 0 3 6 1 6 0]

[0 1115 1 3 0 1 4 0 11 0]

[14 6 916 20 13 0 11 19 30 3]

[3 2 5 934 0 26 1 12 19 8]

[1 5 3 1 904 1 12 1 4 50]

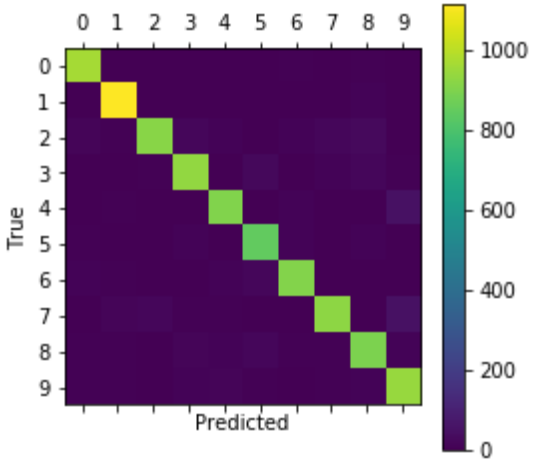
[7 3 0 11 1 847 9 1 10 3]

[12 5 2 1 11 18 906 0 3 0]

[1 14 21 6 5 2 0 925 2 52]

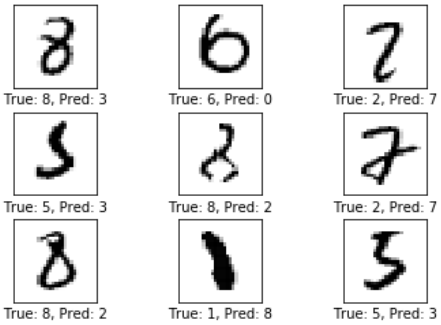
[7 6 2 14 10 19 5 7 893 11]

[8 7 2 9 15 8 0 5 10 945]]



After iteration 9000,

Example errors:



Confusion Matrix:

[[973 0 0 0 0 2 2 1 2 0]

[0 1127 3 0 1 0 1 2 1 0]

[3 1 1020 0 1 0 0 5 2 0]

[0 0 1 1004 0 0 0 2 2 1]

[0 0 0 0 981 0 1 0 0 0]

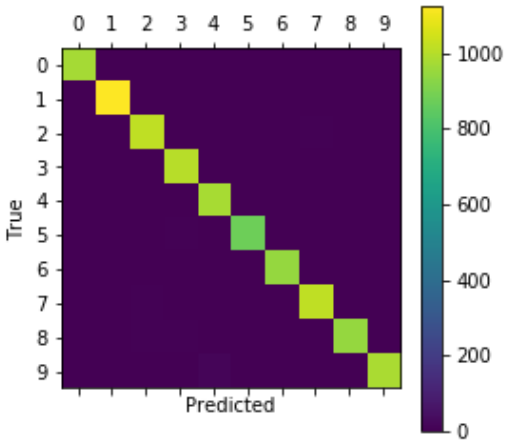
[2 0 0 7 0 878 3 0 0 2]

[4 2 0 0 3 2 947 0 0 0]

[0 1 5 1 0 0 0 1021 0 0]

[3 1 6 5 2 0 2 3 949 3]

[0 4 0 1 14 3 0 4 0 983]]



6 CONCLUSION

Thus the Neural network was implemented and tested with MNIST and CelebA datasets. The result was plotted and compared with the deep neural network. The hyper parameters were also obtained. The accuracy of classification method on the handwritten digits' test data obtained is 95%. The accuracy of classification method on CelebA data set is obtained as 94%. Finally, the convolutional neural network was studied and the output was plotted and explained.