CSE 574 Introduction to Machine Learning

Programming Assignment – 1

Handwritten Digits Classification

Date: March 08, 2017

Group 12

Venkata Saicharan Kolla
Chaitanya Vedurupaka
Anirudh Yellapragada

Classification of Digits using Single Layer Neural Network:

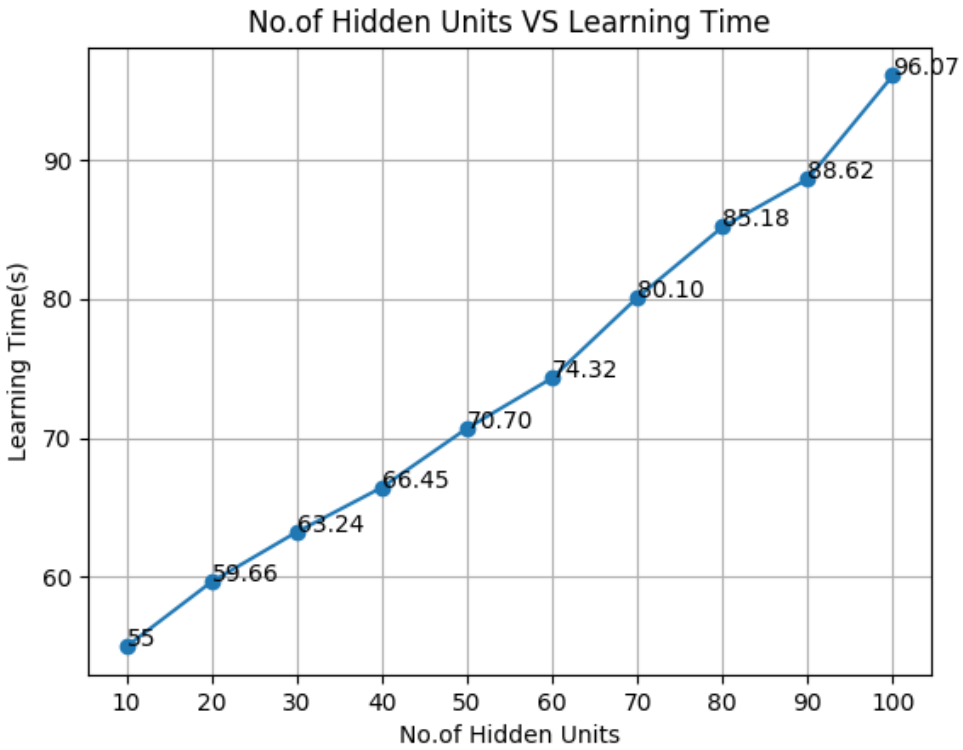
The data to train and test the neural network consists of 60,000 data points, in which 50,000 data points is used for training and 10,000 data points is used for validation to estimate hyper parameters. And a test data of 10,000 data points is used to test the trained model. The working of the model is tested on these points and is optimized verifying the accuracy of the model for different values of hyper parameters (# hidden units, lambda).

Feature Selection and pre-processing:

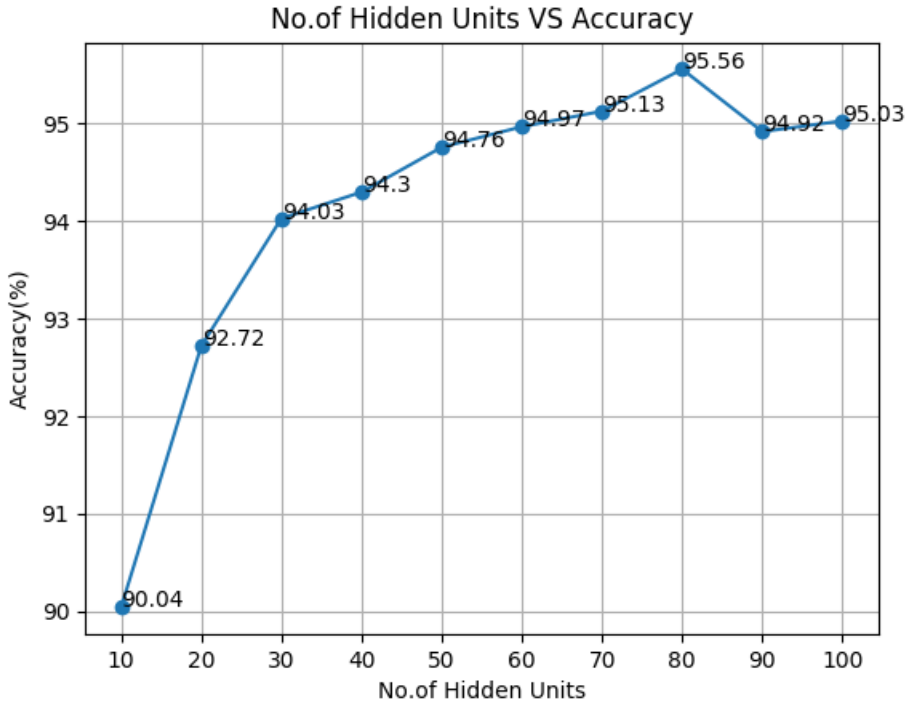
In the data images given, some of the pixels are not needed in training the model. We have a total of 784 column vectors of which we remove some of them which have the same values. These are the pixels that do not have any image data. The remaining data is used to train the model. Given large amount of data, this helps in improving the efficiency of the trained model. The number of selected features in our model is 712.

Hyper Parameters & Accuracy for the nnScript:

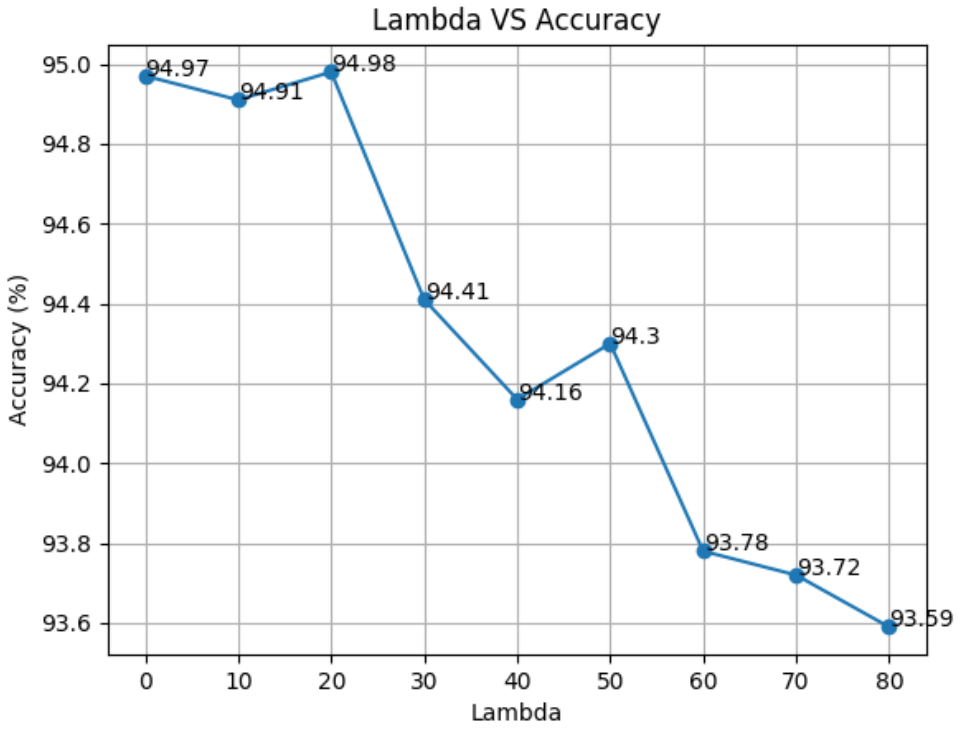
For the handwritten digits, when the number of units in hidden layer is increased, the learning time for the training of the model increases and the accuracy also increases.



The learning times (in sec) are plotted for lambda value, $\lambda = 10$



The highest accuracy is obtained at 80 hidden units and for lambda of 10. The accuracy increases as the number of hidden units increase and then decreases after 80 units.



We have plotted the graph for various values of lambda and for the default value of 50 hidden units. The highest accuracy is observed at $\lambda = 20$. All the plots are acquired evaluating the test data.

Explanation behind choosing hyper-parameter for Neural Network:

We can observe from the graphs that as we increase the number of nodes hidden nodes the prediction accuracy is increased. Also the learning time increases as we increase the number of nodes. The accuracy increased significantly by increasing the hidden nodes (64.46% at 4 nodes and 95.56% at 80 nodes), but the downside is that the learning time also increases reasonably. This is because as we make the network more complex with more number of nodes, the computations increase and hence more time. Since we are focused on accuracy and the increase in accuracy is noteworthy, the increase in learning time can be disregarded. That is how we arrived at the optimal hidden nodes value of 80.

In addition, we can observe from the graphs that there is decrease after reaching highest peak in case of the accuracy. This is because as we make the network more complex with more nodes, the model doesn't perform well on new data. This is called *overfitting*. This is more observed in deep learning as the neural network has more layers containing many neurons. Regularization is used to reduce over-fitting and increase classification accuracies. The regularization factor is mainly used to prevent the overfitting observed in more complex models. It is used in modifying the objective function to manage the weights in the neural network. So, for choosing the lambda regularization factor, we need to do some heuristics to find the optimal lambda value. We have experimented with different values of regularization parameter lambda and found that non-regularized runs take more time to converge compared to regularized runs. In the non-regularized runs, we observed that weight vector was growing by a large margin compared to regularized runs. We tested with different values of lambda and hidden number of nodes and tried to find that value of lambda which gives smaller difference between Training accuracy and testing accuracy. For smaller hidden values, the difference between Training accuracy and testing accuracy was large compared to higher hidden values. After testing with different lambda values (0-80) and hidden nodes (0-100), we observed the maximum accuracy at lambda = 10 and hidden nodes = 80. The accuracy observed for the chosen parameters are

Training data: 95.69 Validation data: 95.14 Testing data: 95.56

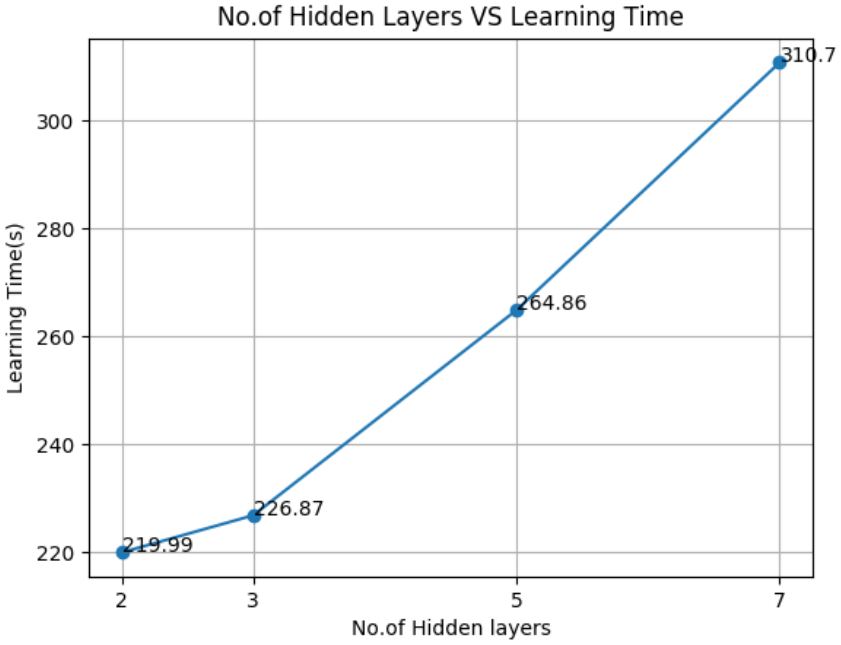
Comparison between facenn and deepnn models:

Using the neural network designed, we ran the model on a photographic image data (celebA data) and test the accuracy of our neural network. For default values given, the accuracy values for corresponding data are

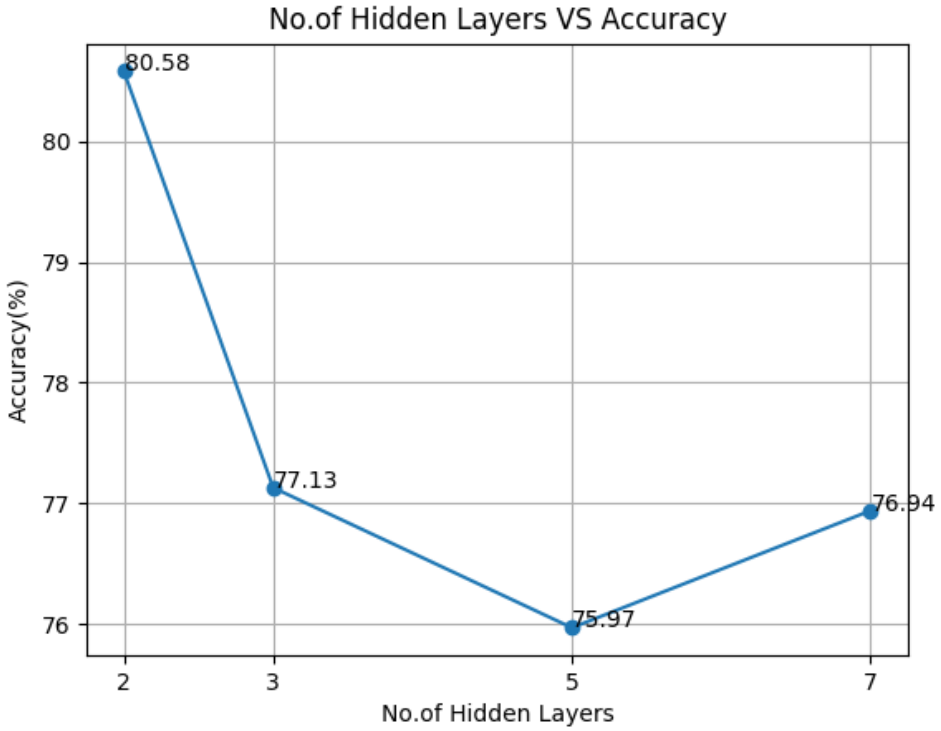
Training data: 86.64% Validation data: 86.34 Testing data: 86.48

Deep Neural Network:

The celebA test data is now evaluated on a deep learning network to learn the accuracy and the learning time, when number of layers are increased. This is done using the Google's TensorFlow Library.



We can observe that as the layers are increased (2 - 7), the learning time is increased. This is because as more layers are added, the calculations become more complex and more optimization of values is needed.



References:

[1] https://www.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html

[2] http://www.kdnuggets.com/2015/04/preventing-overfitting-neural-networks.html