**CSE4/574 Introduction to Machine Learning**

**Programming Assignment 2**

**Members:**

Hrishagni De (hrishagn@buffalo.edu)

Sanchari Gautam (sgautam2@buffalo.edu)

Sahana Mukherjee (mukherj3@buffalo.edu )

**Neural Network – Training Phase**

* Forward Propagation: The forward propagation is implemented on 50000 training inputs with a fixed number of hidden units and layers. Each hidden layer takes the input and processes it with the activation function (here we have used Sigmoid function) and passes it to the next layer.
* Backward propagation: The backpropagation is successfully implemented to make the neural network learn the parameters by calculating the derivatives.

**Regularization**

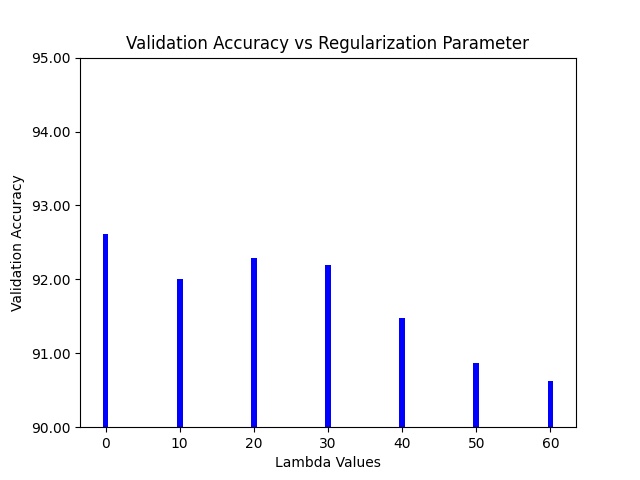
The regularization is incorporated on the neural network using the lambda values to control the overfitting problem. Here, we will see next how the accuracy on the validation set changes with the varying values of lambda.

**Analysis on Hand-Written Digits Using Single Neural Network - NN Script:**

Our neural network consists of a single layer of hidden units and we have to find the best hyper-parameters to train the neural network. We have used the accuracy on the validation set to optimize our neural network.

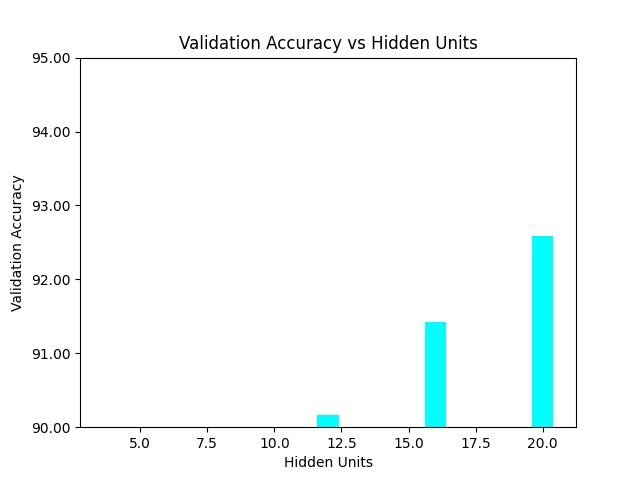
* Validation Accuracy with Varying Lambda Values with 50 iterations

The first step towards our hyper-parameter tuning is that we have used different lambda values starting from 0 to 90 and have validated against our validation set. A plot of the validation accuracies with varying values of lambda is shown below. This shows the change of accuracies with the different regularization parameters keeping all other parameters constant. We can see that **with 50 iterations, it is working best with 0 lambda value (no regularization)**.

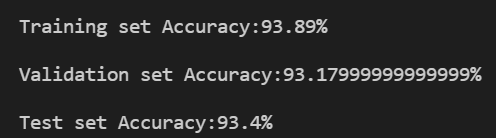


* Validation Accuracy with Varying number of Hidden units at 50 iterations

The accuracy on the validation dataset is plotted against the hidden units with values ranging from 4 to 20. It is seen that the accuracy is highest for the model with 20 hidden units.

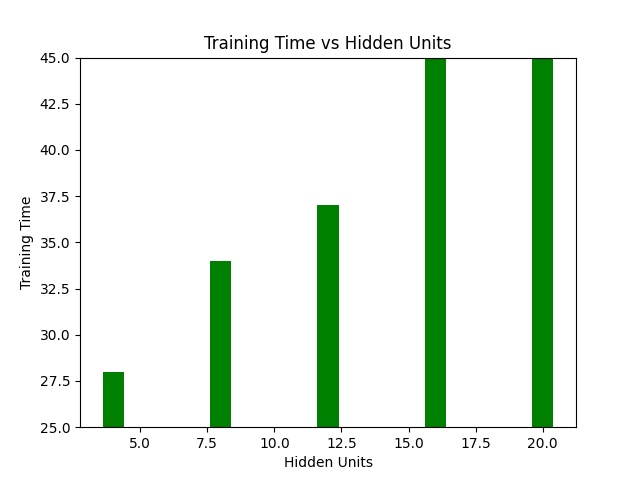


* Accuracy with 20 hidden nodes and lambda=0



* Change of Training Time with the change in hidden units

From the graph below, it is seen that the training time gradually increases with the increase in the number of hidden units, but it drops with hidden units at 20.



* Indices of the Features Used



The entire list of 719 features used for the training process is shown above. This list contains features after removing the redundant ones, which are exactly the same for all the training points. Hence, they are removed in the pre-processing phase.

* Overfitting – Under-fitting Scenario

Under-fitting is the situation where the model can neither predict well on the training dataset nor can it predict on the testing dataset. On the other hand, overfitting is the situation where it predicts well on the training dataset but fails to predict on the testing dataset. That is, it fails to generalize the dataset.

Now, we have enlisted all the training and testing accuracies by changing our lambda values. Therefore we can see that for lambda values ranging from 0 to 60, we do not have overfitting or under-fitting issues but we can see that for lambda=0, our accuracies are quite convincing.

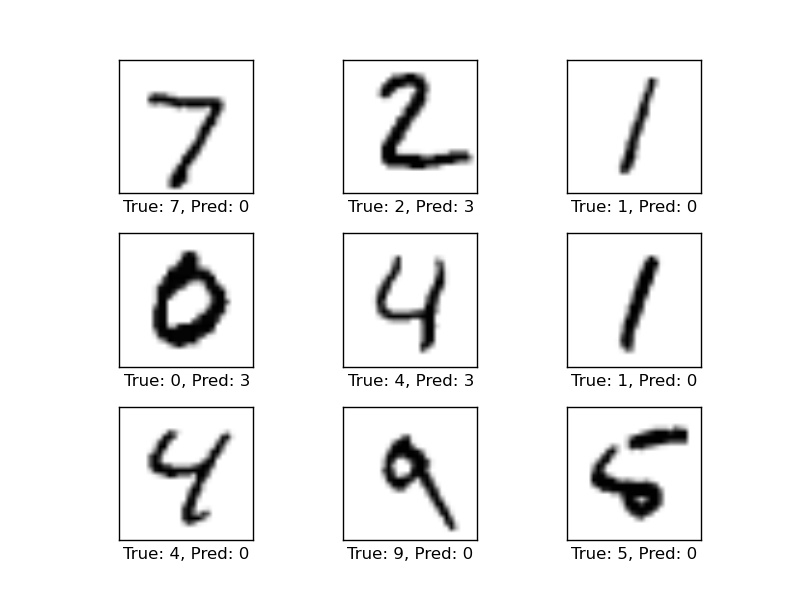
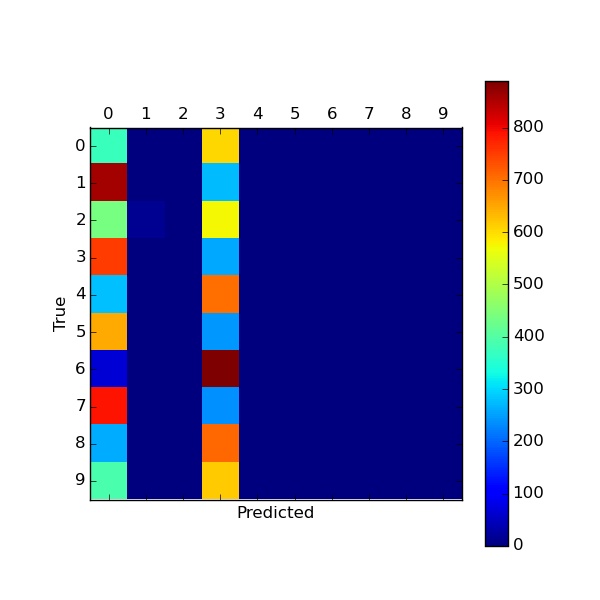
|  |  |  |
| --- | --- | --- |
| **Lambda Value** | **Training Accuracy (in %)** | **Testing Accuracy (in %)** |
| 0 | 93.348 | 93 |
| 10 | 92.44 | 92.25 |
| 20 | 92.806 | 92.62 |
| 30 | 92.842 | 92.81 |
| 40 | 92.138 | 91.199 |
| 50 | 91.566 | 91.37 |
| 60 | 91.418 | 91.08 |

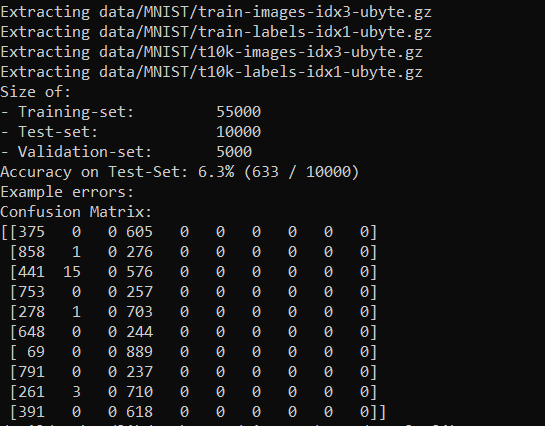
***Thus, the best parameters for training a single neural network to predict the hand-written digits is ensured to be 20 hidden nodes and regularization parameter at 0.***

**Analysis on Hand-Written Digits Using Convolutional Neural Network - CNN Script:**

The hand-written digits’ dataset is also analyzed using the Convolutional Neural Network having the same padding and 2X2 pooling technology. The stride is chosen at 1 for the simplicity of the calculations. In each layer, we have used the Relu activation function to add some nonlinearity to the function so that we can learn some complicated functions. We have also used the Adam optimizer with a learning rate of 10^-4 in order to minimize the cost. Now, we have tried to perform the classification using several iterations, each of which is described below:

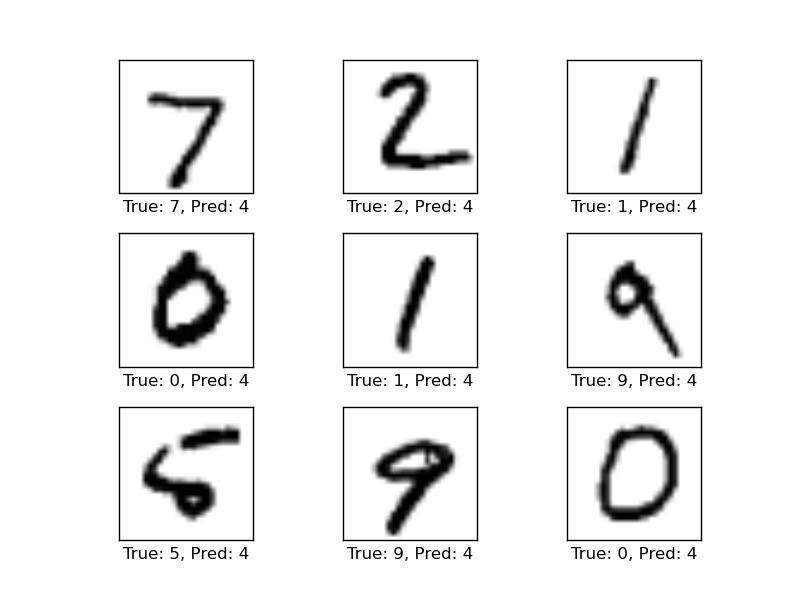
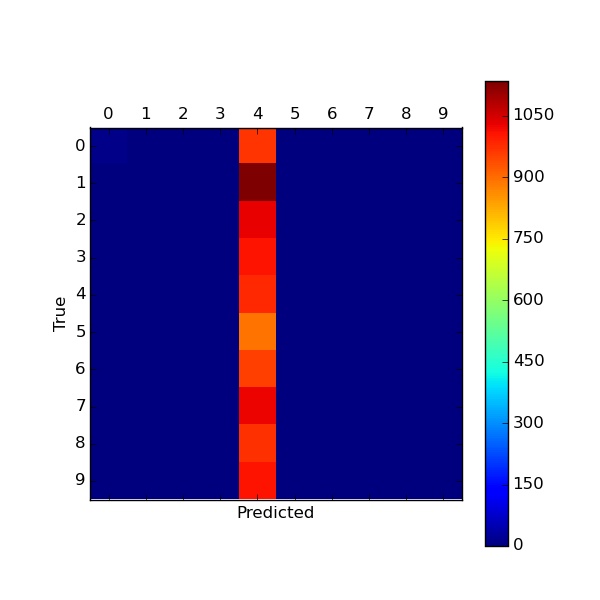
* **Number of Iterations - 1:**

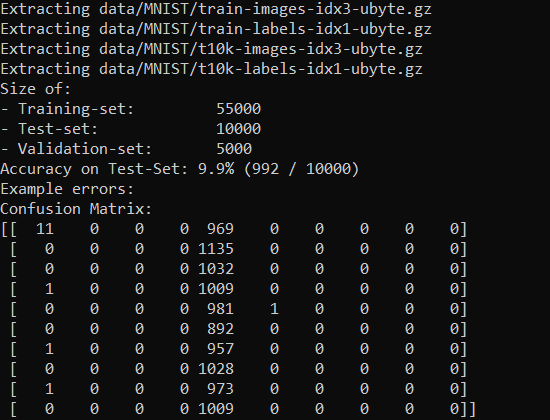
****

****

From the above pictures, it is clear that the prediction labels do not match the true classes of the digits. The testing accuracy is found out to be 6.3%, that is 633 digits out of 10000 digits have been correctly identified. To improve the accuracy, let us try to increase the number of iterations.

* **Number of Iterations - 99**

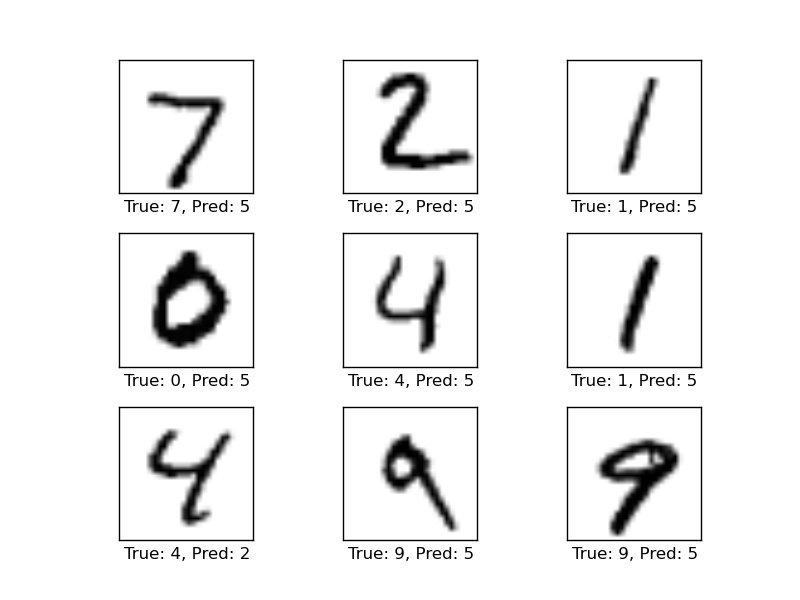
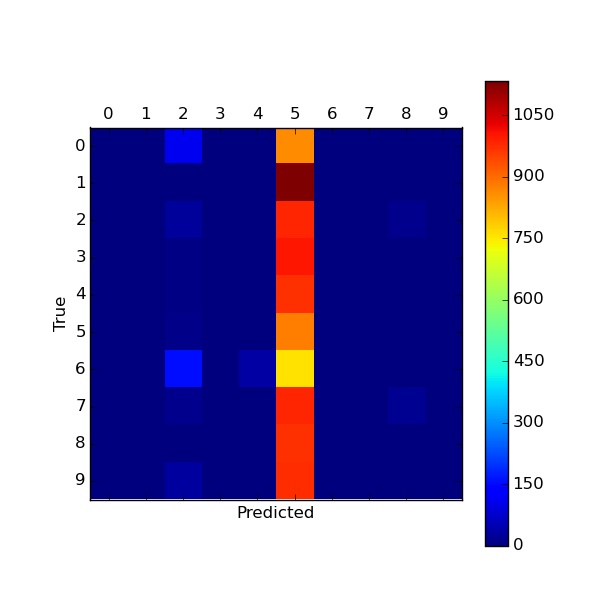
****

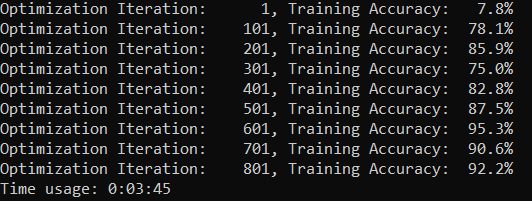
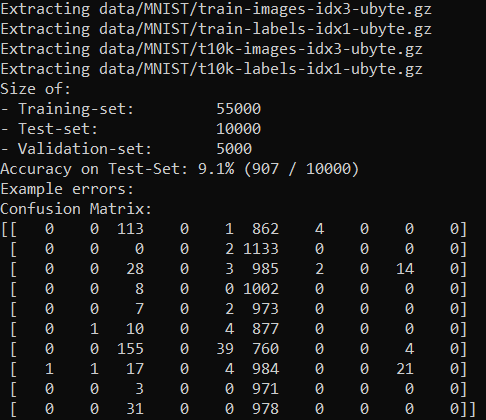
****

****

Now it can be seen that the testing accuracy has increased a bit to 9.9% with 99 iterations from 6.3% with 1 iteration. Next, we will try to increase the iterations more to improve the accuracy.

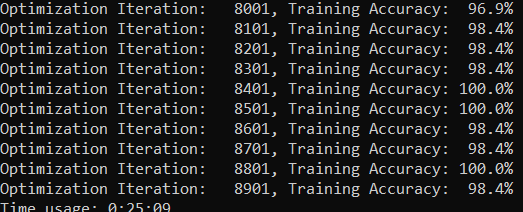
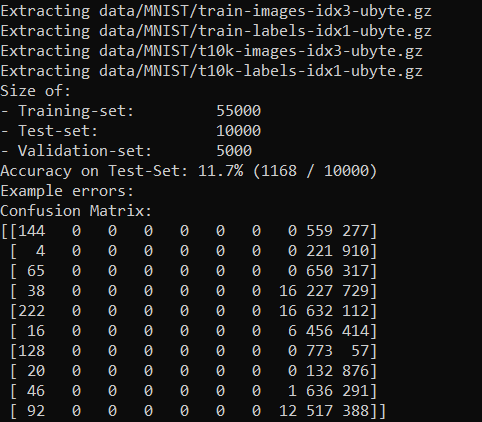
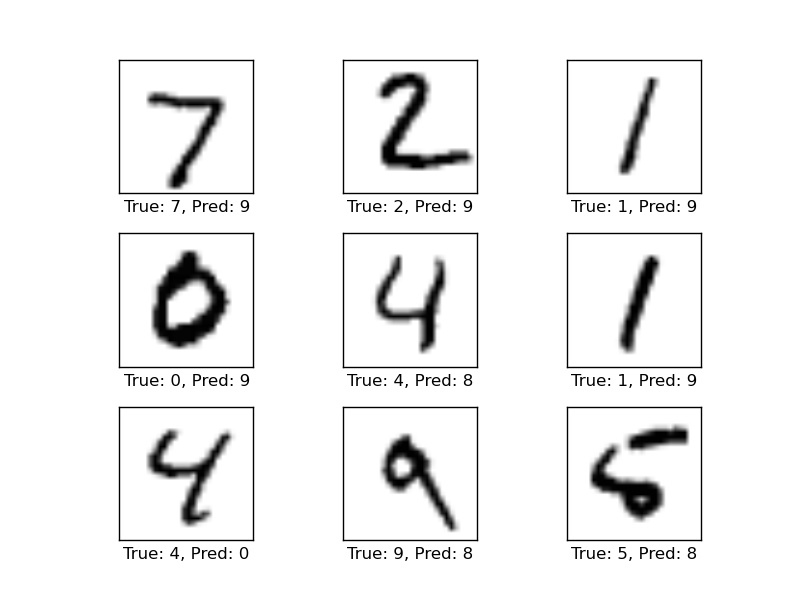
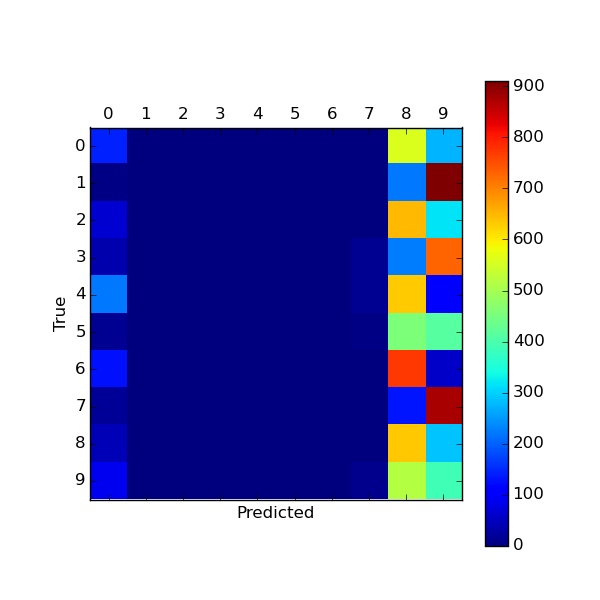
* **Number of Iterations - 900**

****

****

Although the testing accuracy has not been affected much, the training accuracy has increased drastically from 15% on average to 85% on average. The time required to complete the training period is 3 minutes. We will now try to optimize the network for 9000 iterations.

* **Iteration 9000**

****

Now with the same architecture of the convolutional neural networks, with 9000 iterations, we can see that the test accuracy has increased to 12% approximately with an average training accuracy of 98%. The total training time is 25 minutes.

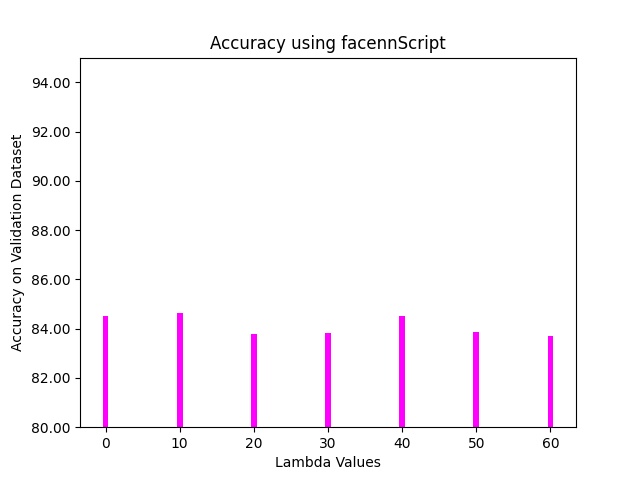
* **Convolutional Neural Network in terms of Accuracy and Training time**

|  |  |  |
| --- | --- | --- |
| **Iteration #** | **Testing Accuracy** | **Time Usage** |
| 1 | 6.3% | 0 secs |
| 99 | 9.9% | 8 secs |
| 900 | 9.1% | 3 minutes 45 secs |
| 9000 | 11.7% | 25 minutes |

**Analysis on Celebrity Faces (CelebA Dataset) Using Single Neural Network - FaceNN Script:**

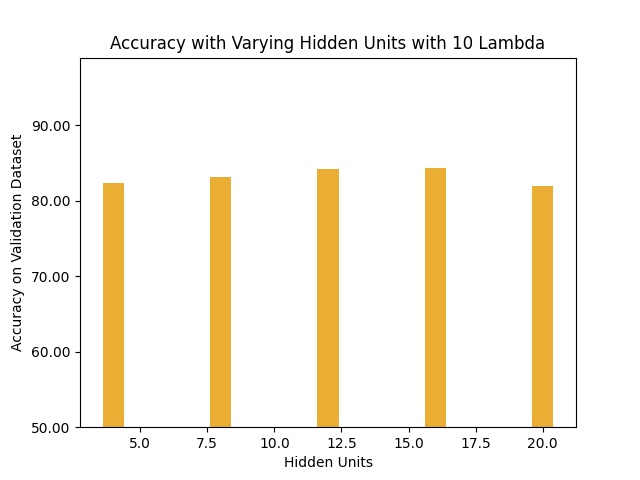
* Accuracy vs Lambda

For the FaceNN script to classify the CelebA dataset, we can see from the graph below that lambda=10 works best because the validation accuracy is highest for this value. Hence we will move forward with this value.

****

* Accuracy vs Hidden Units at lambda=10

Now, we have plotted the validation accuracy with various hidden units in a range of 4 to 20. It is clear from the graph that for hidden units at 16, the validation accuracy is highest for the celebrity face classification.

****

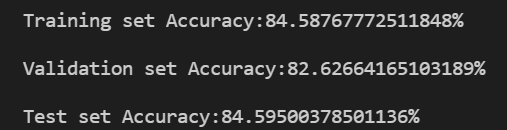
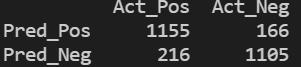
* Training Time vs Hidden Units

The training time to classify the celebrity faces using a normal neural network is plotted below against the number of hidden units. It is seen that the training time is highest for the hidden units at 16.



* Accuracy of single hidden layer Neural Network on CelebA data set

The accuracy of single hidden layer Neural Network is computed on CelebA data set (test data only), to distinguish between two classes - wearing glasses and not wearing glasses. The screenshot below shows the training, validation and testing accuracy along with the confusion matrix.

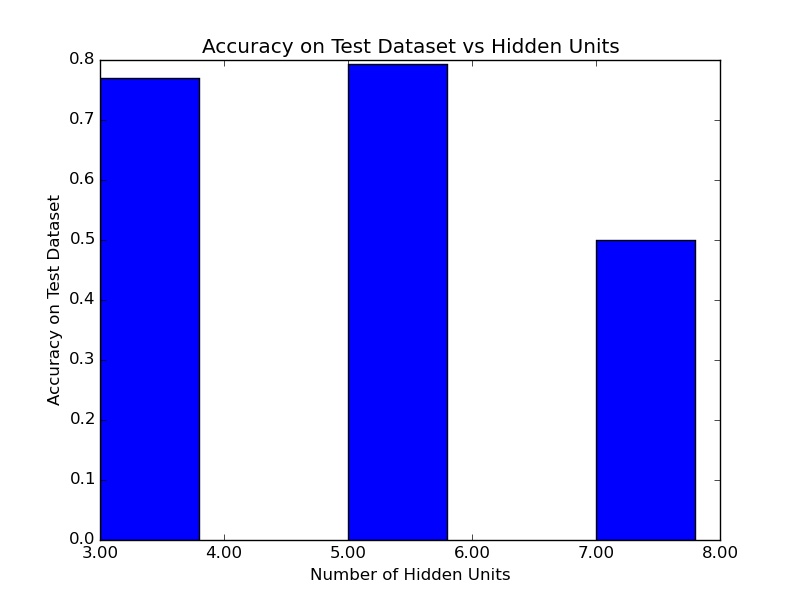
 

Thus, it is seen that using a single hidden layer neural network on the CelebA dataset, the validation set accuracy is 82.6% and that on test dataset, the accuracy is 84.6%. Hence, we can conclude that the classification has been done satisfactorily.

**Analysis on Celebrity Faces (CelebA Dataset) Using Deep Neural Network - DeepNN Script:**

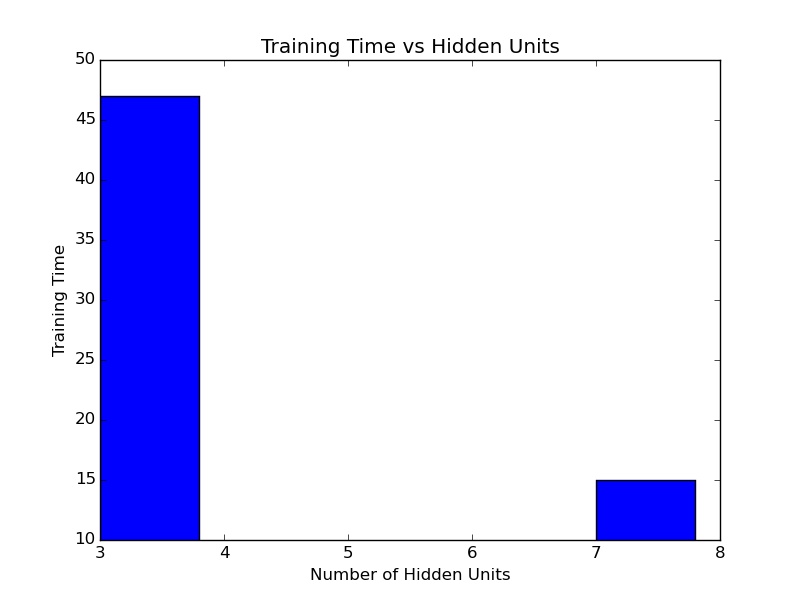
* Accuracy on test dataset with hidden units at 3, 5 and 7

The accuracy on the test dataset is plotted against the hidden units at 3, 5 and 7 and it is seen that the accuracy is highest at 5 hidden units.

****

* Training Time vs hidden units at 3, 5 and 7

The training time is plotted against the number of hidden units. It is seen that for hidden units at 3, the training time is the highest, that is 47 seconds but for hidden units at 5 and 7, the training time is significantly low (below 15 seconds).

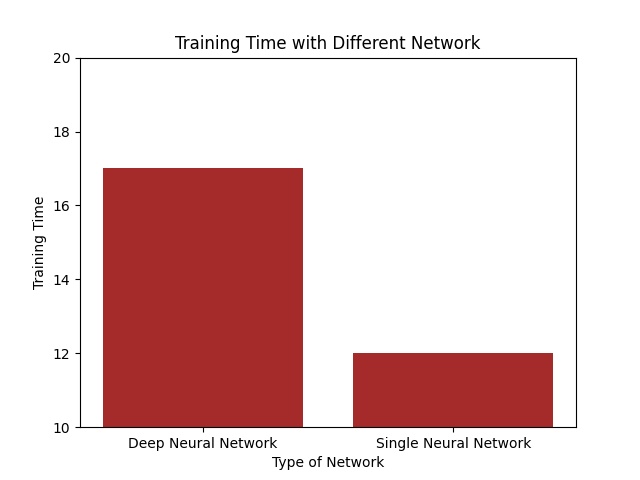
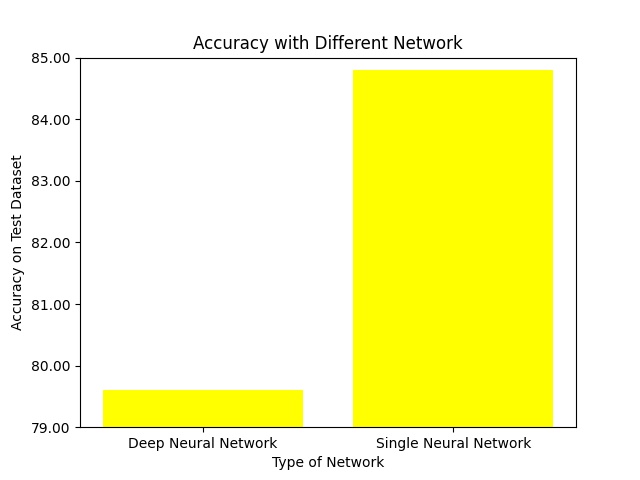


* Performance of Single vs. Deep Neural Networks in terms of accuracy on test data and learning time

We have classified the CelebA dataset using both a single neural network as described in the FaceNN script and a deep neural network using the DeepNN script. Both of the networks have successfully classified our positive and negative classes.

*Deep NN Single NN*

**

From the graph above, it can be seen that using Deep Neural Network, we could reach a testing accuracy of only 79% but with a single neural network, we were able to classify with an accuracy of 84.21% on the testing dataset. Hence, the single neural network happens to work more accurately in this scenario.

Also, the deep neural network takes a lot of training time in comparison to a single neural network. Thus, it can be concluded that our single neural network has worked more efficiently in terms of accuracy and training time on the CelebA classification than the deep neural network.