CSE 574 Introduction to Machine Learning
Programming Assignment 1
Handwritten Digits Classification

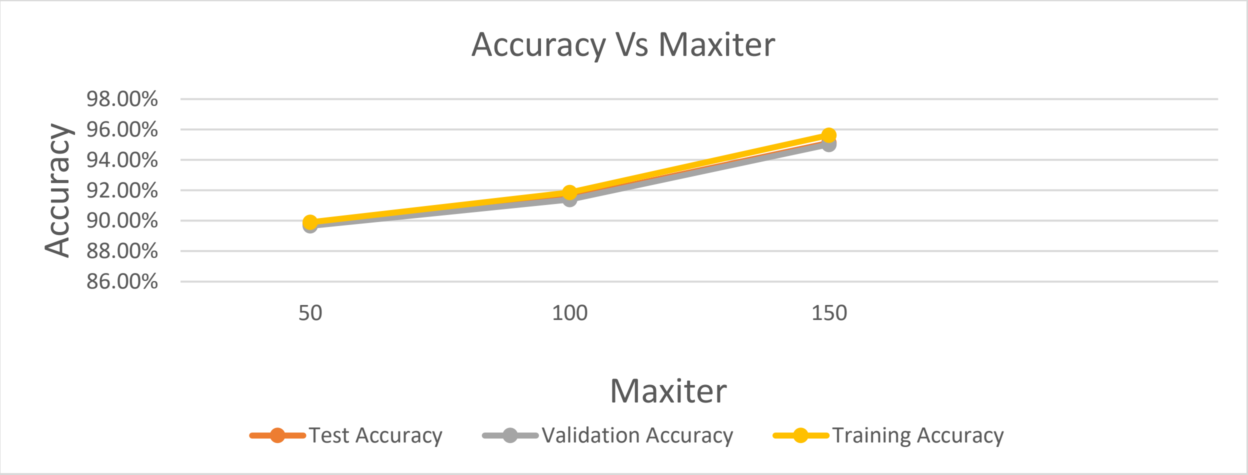
Group 4

Saurabh Bajoria	50208005
Sumedh Ambokar	50207865
Vidhi Shah	50207090

1. Abstract

- In this assignment, we have implemented a Multilayer Perceptron Neural Network to classify handwritten digits.
- Neural Network is divided into 3 layers: Input Layer, Hidden layer(s) and Output Layer. Each layer is connected to the next layer with some weights applied on the interconnections.
- We preprocessed the input data, selecting useful features from the training set. On this feature set, we implemented Forward Feed and Backpropagation, depending on the error obtained at the output layer.
- We conducted several simulations to determine how the parameters- Lambda (λ), Hidden Nodes and Maxiter influence the accuracy of the system and the training time.

2. Relationship between Maxiter and Accuracy

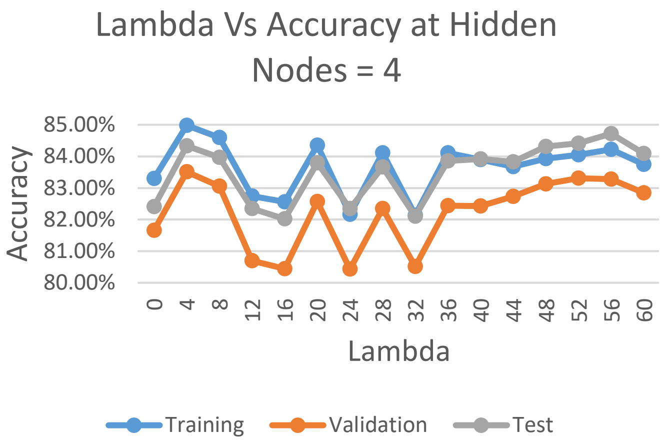


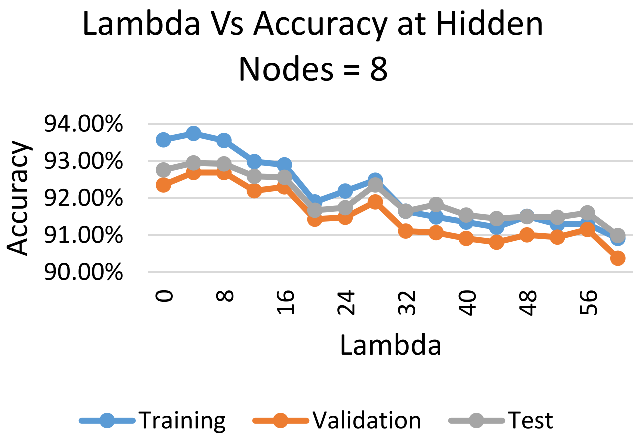
In the above graph, we have plotted the Average Accuracy of the system against different values for Maxiter.

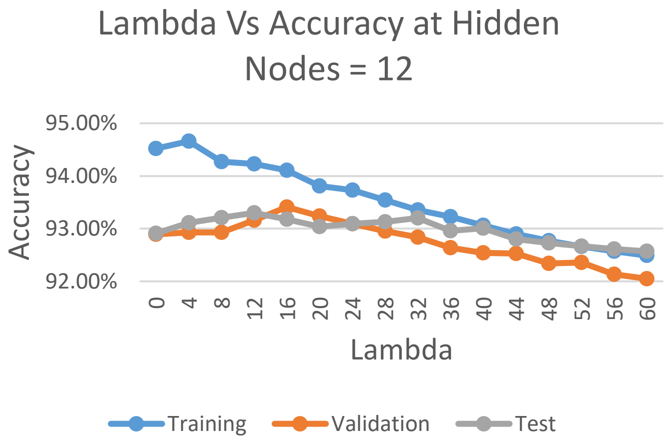
As seen, on increasing the maxiter, the accuracy of the system increases. This is observed because, maxiter value determines the number of times the objective function is called, which in turn helps to refine the weight vectors and eventually reduce the overall error.

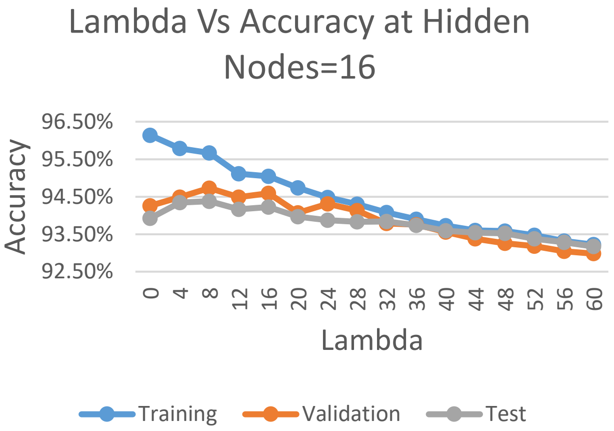
However, it was also observed that increasing maxiter value also increased the training time.

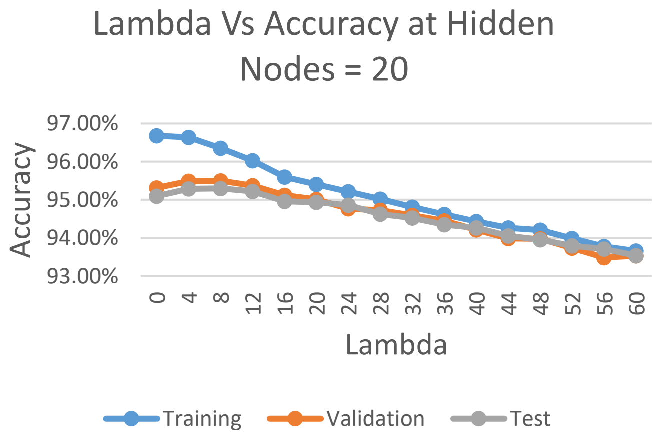
3. Relationship between Lambda and Accuracy (for different values of Hidden nodes)

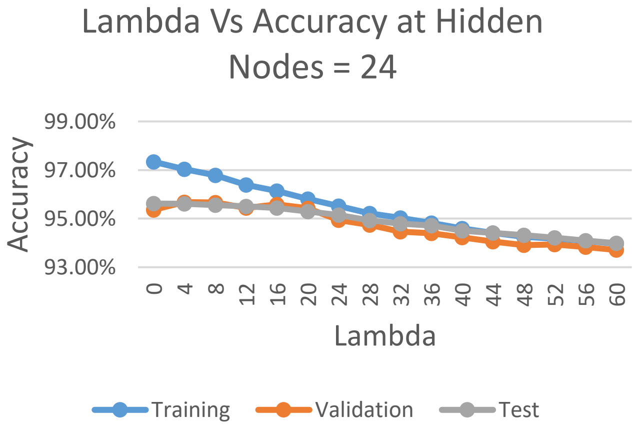












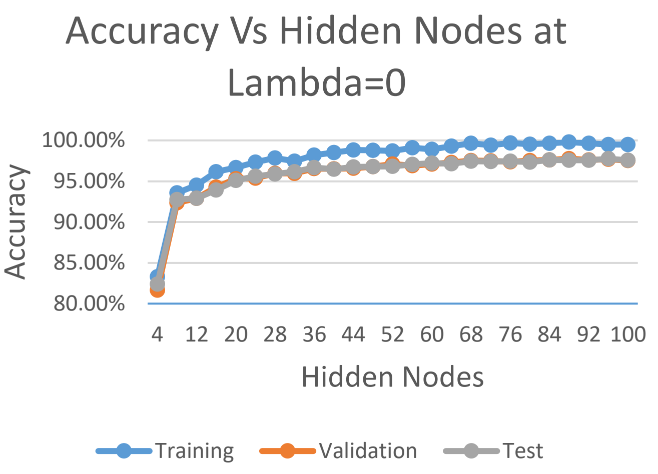
The above graphs show the relation between lambda and accuracy of the system, for different number of hidden nodes.

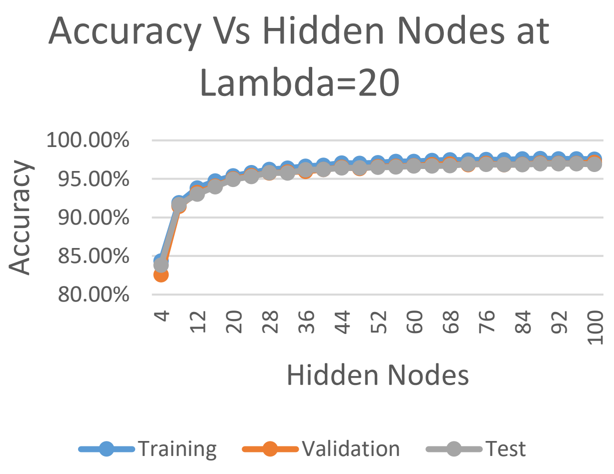
Lambda helps in regularizing the error function to control the magnitude of the change in weight parameters and avoid overfitting.

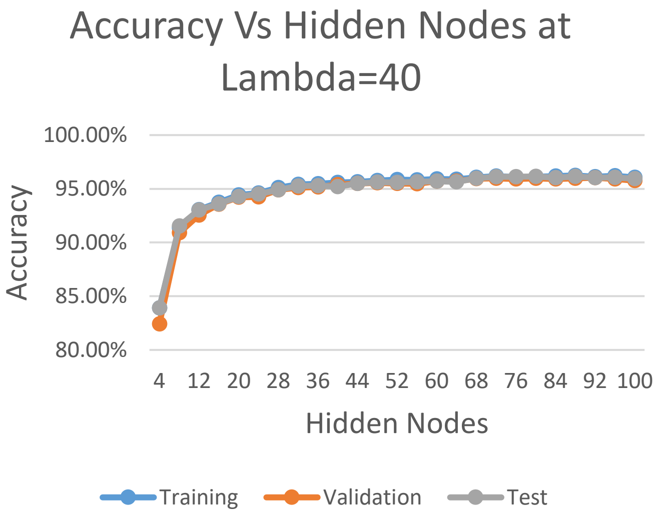
Ideally, the accuracy of the testing data should increase with increase in the value of lambda, whereas the accuracy of the training data should decrease. But instead, we found that the accuracy was better for lower values of lambda. This could be because of the similarity of the test data to the training data and randomized initial weights.

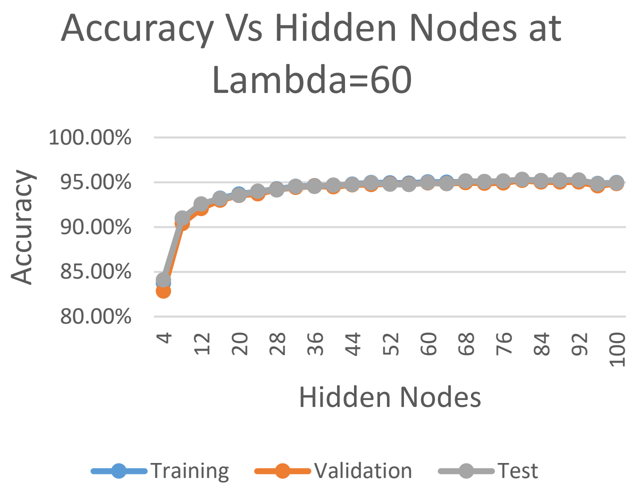
Also, the training time of the system is higher for values of lambda close to 0 because non-regularized runs often takes large number of iterations when compared to regularized runs.

4. Relationship between Hidden Nodes and Accuracy







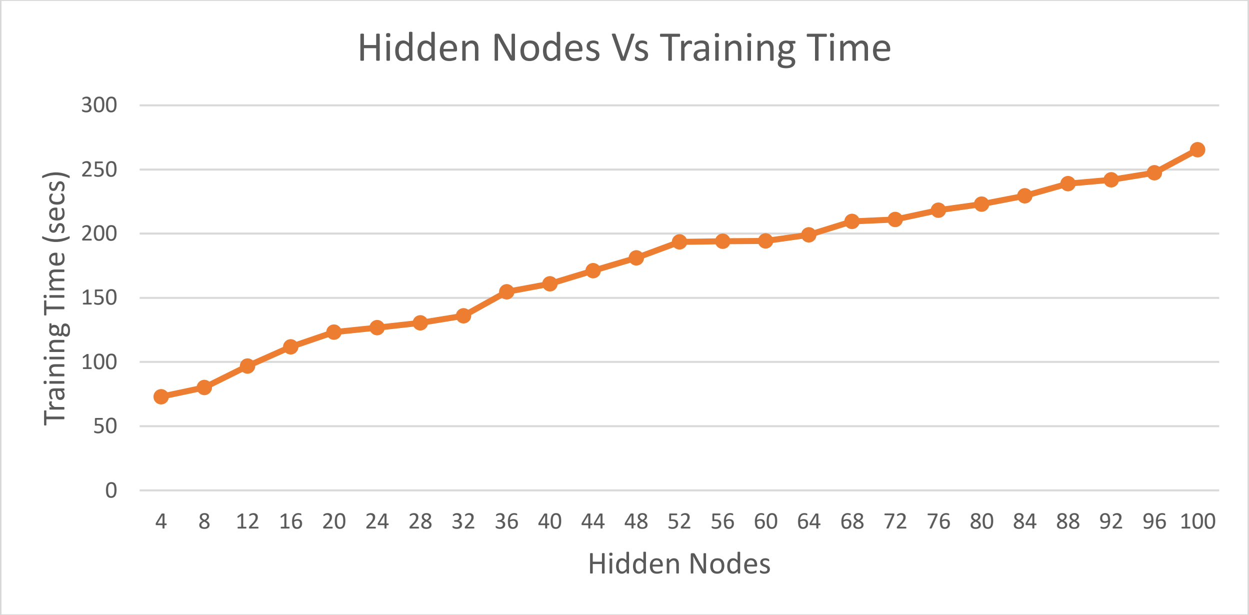


As seen above, for lower values of Hidden Nodes, we receive low accuracies for any value of Lambda. The accuracy of the system increase with increase in the Hidden Nodes. This is an expected behaviour as a larger number of hidden nodes provides a higher precision.

We learn from the graphs that the accuracy increases exponentially during the initial increase in the value of hidden nodes but has a gradual increase after a certain value. In this case this change is observed after value of hidden nodes is increased to 12.

Also, the gap between the testing and training accuracy was larger when the value for lambda was low and the gap reduced significantly for higher values of lambda. Thus, value for lambda and hidden nodes should be selected such that we get maximum consistency in prediction result.

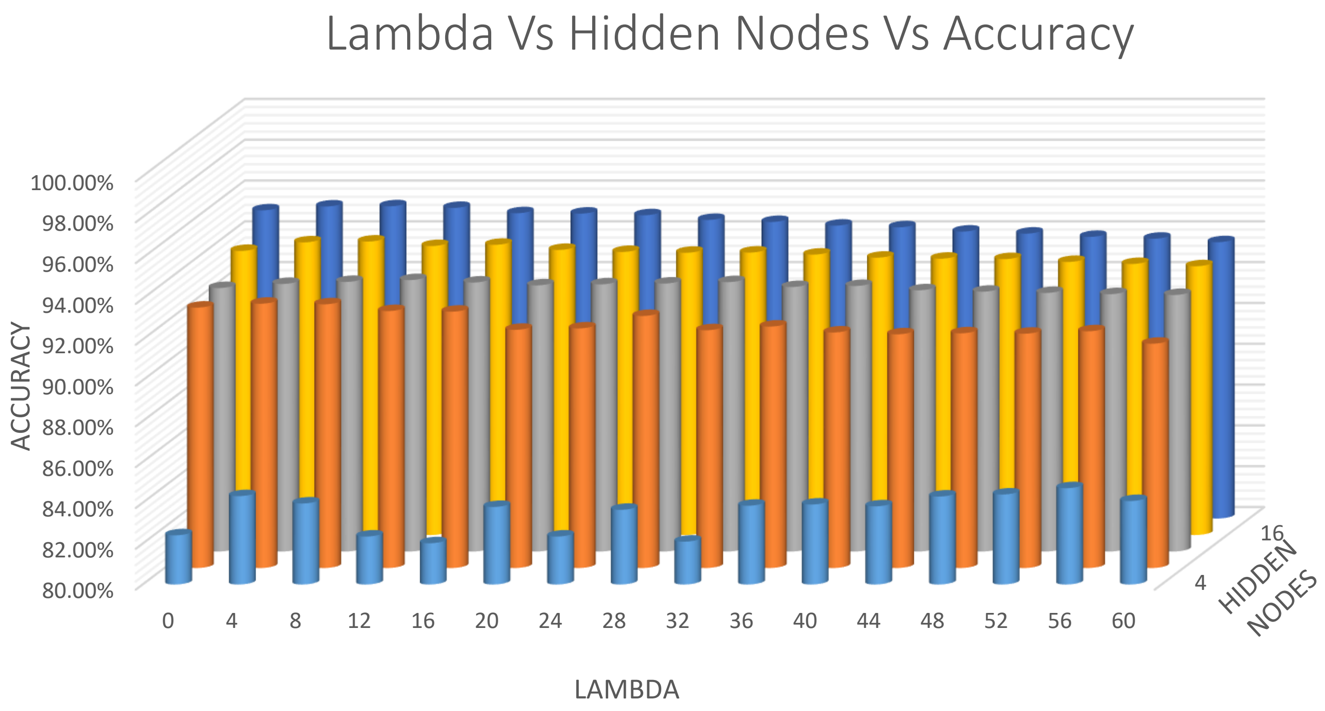
5. Relationship between Hidden Nodes and Training time



The above figure depicts the training time taken against different number of hidden nodes.

As expected, the system takes more training time when the number of hidden nodes is increased. This is because, increase in the number of the hidden nodes would result into computation of more weights and gradients, thus increasing the overall complexity of the system.

6. Relationship between Lambda, Hidden nodes and Accuracy



The above figure plots Accuracy against Lambda and Hidden nodes.

As per our observations, if training time is not an issue, increase in the number of hidden nodes increases the overall accuracy of the system. But as observed in earlier graphs the increase in accuracy of the system is minimal after certain value of hidden nodes. Thus, if time is an constraint then a hidden node value of 12 or more can be set.

While considering the best combination for the values of Lambda, Hidden Nodes and Maxiter we have taken into account Accuracy percentage as well as Execution time. Below is the best combination as per our output values.

- Test Accuracy Percentage: 97.53%
- Lambda(λ): 8
- Hidden Nodes: 96
- Maxiter: 150
- Training time: 185.36 seccs

A slightly better value of Accuracy was achieved when lambda was set to 0 but the execution time for that combination was much higher than the one provided above.

7. Single Layer Neural Network Vs Multi-Layer Neural Network (facenn Vs deepnn)

Facenn script is a Single Layer Neural Network implementation and deepnn script is Multi-Layer Neural Network.

We observed that Facenn script though having a lesser complexity model provided better accuracy and took less execution time. Usually based on the concepts of the neural network a complex architecture should provide a better result in reference to the accuracy of the testing data. But for the data provided we observed contrary of this theory. This can be accounted to the size of the training data and efficiency of the regularization function applied.

Also, we observed that the accuracy of the deepnn script reduced with increased number of layers.

Thus, for this specific example more complexity has reduced the accuracy of the system.

Accuracy values obtained for facenn and deepnn:

- facenn
  Test set Accuracy: 86.9795609387%
- deepnn
  Layer 2: Test set Accuracy: 80.7721%
  Layer 3: Test set Accuracy: 78.8796%
  Layer 5: Test set Accuracy: 77.1764%
  Layer 7: Test set Accuracy: 73.732%
  Layer 9: Test set Accuracy: 71.0825%

8. References:

- http://www.numpy.org/
- https://www.tensorflow.org/
- https://docs.python.org/2/library/pickle.html
- https://www.scipy.org/