Handwritten Digits Classification Using Neural Networks

Course Number: CSE 574
Introduction to Machine Learning

Team Number: 6

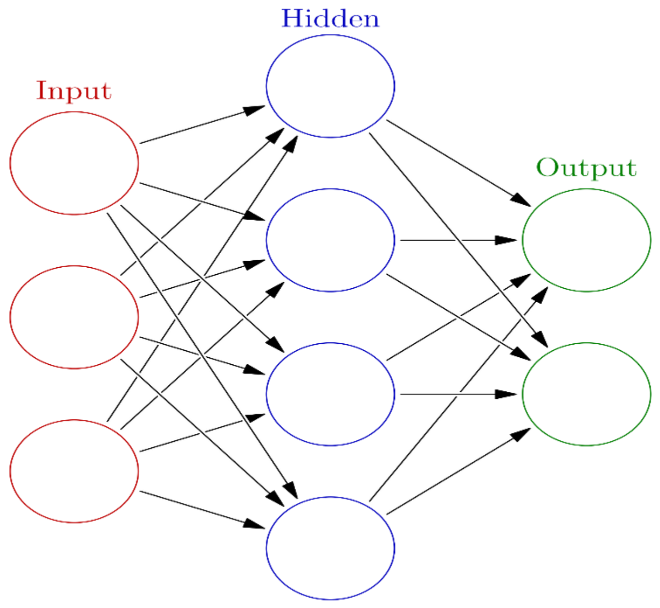
Akash Mandole

Chintan Thakker

Pranshu Pancholi

Introduction

The assignment involves implementation of a Multilayer Perceptron Neural Network and evaluation of its performance in classifying handwritten digits. A neural network can be graphically represented as following:



• The first layer comprises of (d + 1) input units, each represents a feature of image (there is one extra unit representing bias node.
• The second layer in neural network is called the hidden units. There is an additional bias node at the hidden layer as well. Hidden units can be considered as the learned features extracted from the original data set. Too many hidden units may lead to the slow training phase while too few hidden units may cause the under-fitting problem.
• The third layer is also called the output layer. The value of lth unit in the output layer represents the probability of a certain hand-written image belongs to digit l. Since we have 10 possible digits, there are 10 units in the output layer. In this document, we denote k as the number of output units in output layer.

The implementation involves two main components:

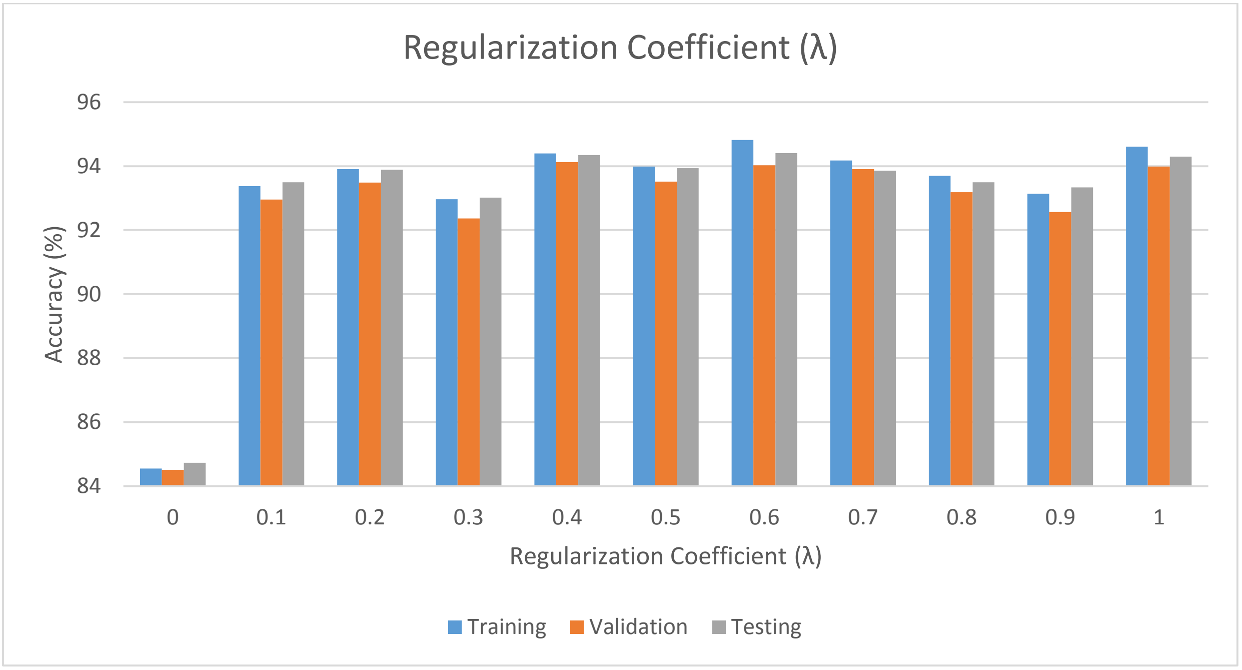
1) Feedforward Propagation
2) Error Function and Backpropagation

Choosing hyper- parameters for Neural Network

One of the problems that occur during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations.

Following graph shows the variation of accuracy by changing different values of regularization parameter (λ):

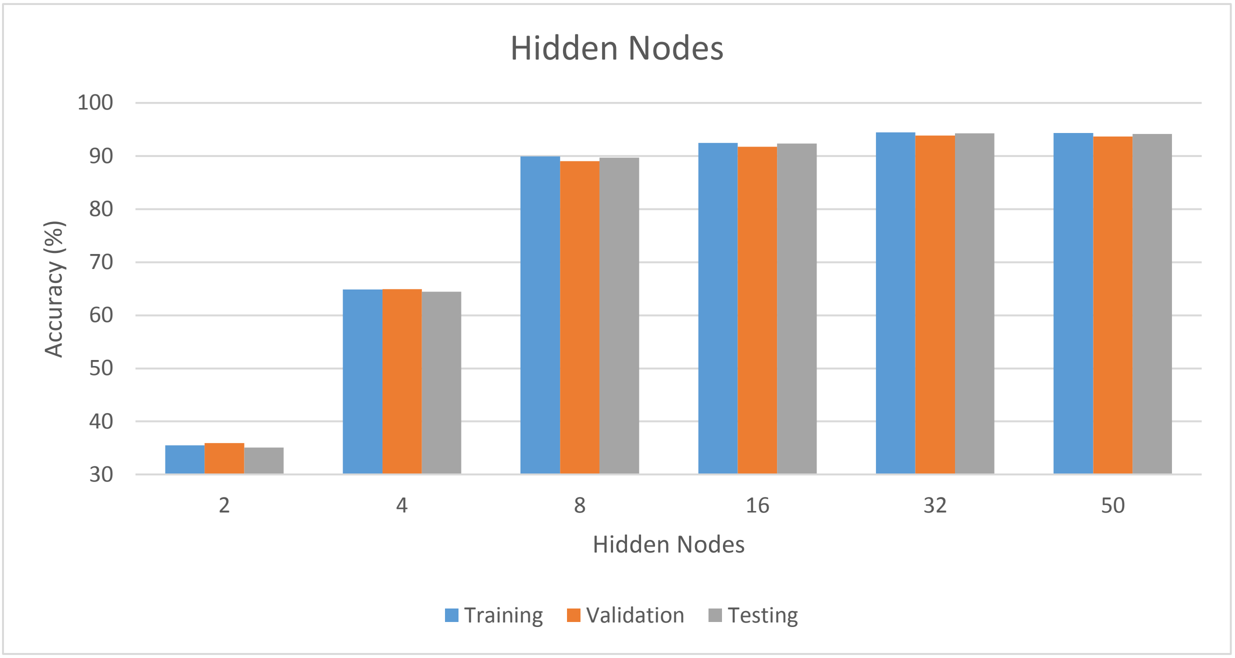
*NOTE: Number of hidden nodes are 50.



Conclusion: Value of regularization coefficient can be varied to improve the classification accuracy and reduce the over-fitting problem. As it is evident from the graph with regularization coefficient as ‘0’ the accuracy of the system was considerably low. Hence non regularized runs performed least efficiently and the algorithm was not able to converge to local minima making it harder for the system to learn weights. The **best performance of 94.41%** was achieved by setting the value of **Regularization Coefficient (λ) as 0.6**.

Following graph shows the variation of accuracy by changing number of hidden nodes:

*NOTE: The Regularization Coefficient is set to 0.1.



Conclusion: It is clearly evident from the graph that increasing the number of nodes significantly increases the accuracy of the system. However, the computation time also increases significantly with the increase in number of nodes. For a system where the accuracy is more desirable the number of nodes must be increased, here we have chosen the **hidden nodes as 50** to achieve the highest accuracy.