HANDRITTEN DIGITS CLASSIFICATION

CSE 574 – Introduction to Machine Learning
Programming Assignment – 1

Group – 63
Prakash Natarajan
Saravanan Adaikkalavan
Naveen Kumar Ramamurthy

Overview -

The objective of the assignment is to implement a Multilayer Perceptron Neural Network and evaluate its performance in classifying handwritten digits. The input MNIST dataset consists of a training set of 60000 examples and test set of 10000 examples.

Implementation –

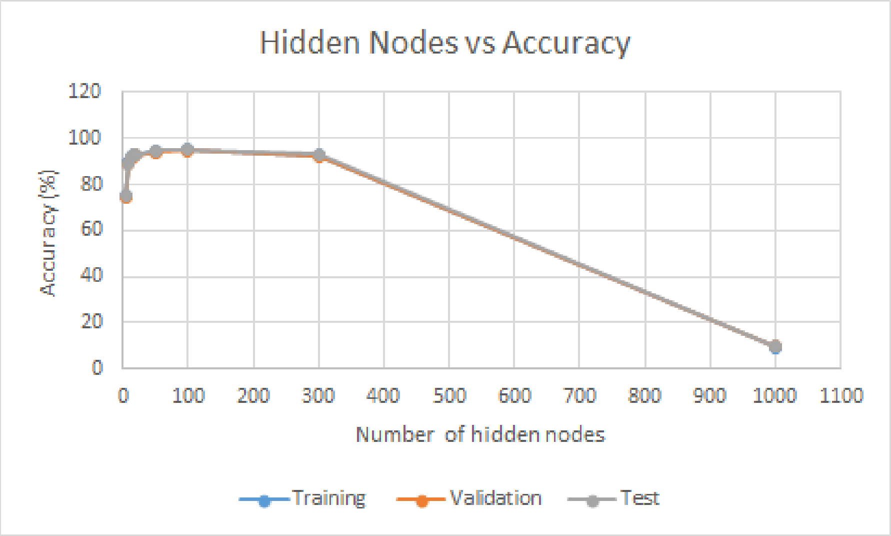
The 60000 training examples of the MNIST dataset are preprocessed and split into two sets – a training set consisting of 50000 samples and a validation set of 10000 samples. The Neural Network is implemented as explained in the assignment description and the performance evaluation results are as elucidated below.

Performance Evaluation –

Number of Hidden Nodes: 100

The hidden layers are believed to enhance the capabilities of the neural networks. The nodes present in the hidden layers, with some tolerance, influence the error that gets propagated through the network that in turn affects the accuracy of the system. In accordance to the above statement and by means of a trial and error mechanism the training accuracy, the validation accuracy and the test accuracy were plotted against a set of hidden nodes of count 4, 8, 12, 16, 20, 50, 100, 300 and 1000. It was observed that for the number of hidden nodes in the intermediate layer 100, the system attained the highest accuracy as illustrated in the graph and the table below for a fixed regularization coefficient value 0.6 as explained in the next section.

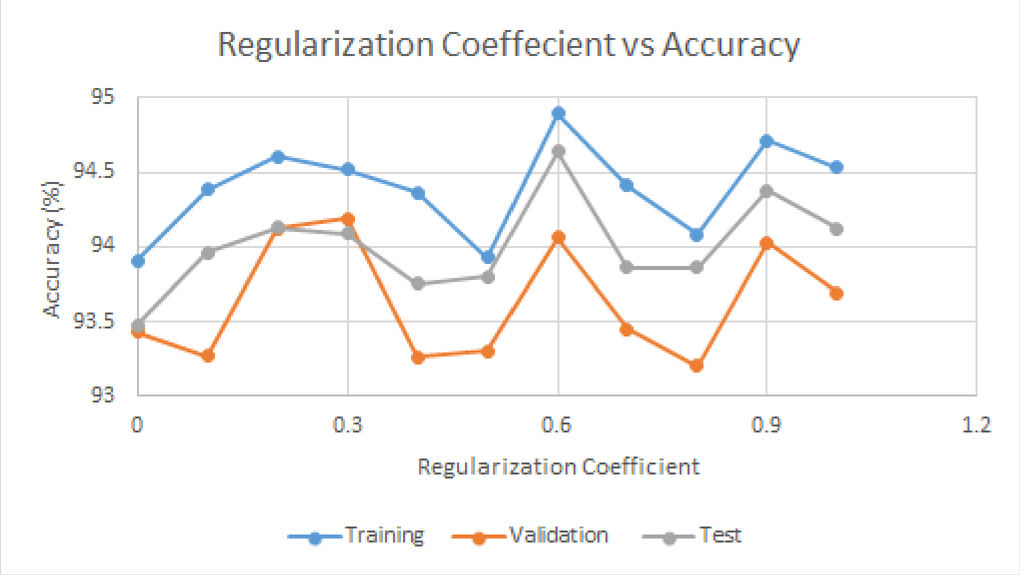
n_hidden	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
4	74.886	74.64	75.25
8	89.642	88.74	88.95
12	92.21	91.46	91.78
16	93.078	92.15	92.74
20	92.852	92.64	92.62
50	94.896	94.06	94.64
100	**95.258**	**94.75**	**95**
300	92.818	92.23	93.02
1000	9.466	9.54	9.6



Regularization Coefficient: **0.6**

The excessive number of hidden layer nodes intended to cause the "overfitting" problem as seen above i.e., the learning model becomes best fit with the training data but gives poor generalization when test with validation data. In order to prevent this problem from occurring, a regularization term is added into our error function to control the magnitude of parameters in Neural Network. The regularization coefficient is constrained to be of magnitude between 0 and 1. The accuracy of the system was obtained for every 0.1 increase in the coefficient value as shown below and the following graph was plotted for a fixed number of hidden nodes 50.

Lambda	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
0	93.908	93.43	93.48
0.1	94.382	93.27	93.96
0.2	94.606	94.12	94.13
0.3	94.516	94.19	94.09
0.4	94.362	93.26	93.75
0.5	93.928	93.3	93.8
0.6	**94.896**	**94.06**	**94.64**
0.7	94.414	93.45	93.86
0.8	94.084	93.2	93.86
0.9	94.716	94.03	94.38
1	94.53	93.69	94.12



It is observed that the system achieved the highest accuracy when the regularization coefficient was **0.6**

Result

Optimal number of hidden nodes = 100, Optimal regularization coefficient = 0.6
Training set Accuracy: 95.258% | Validation set Accuracy: 94.75% | Test set Accuracy: 95.0%