Niall Smith
ECS 171
SID :912370187

Using a feed forward neural network in order to classify the yeast data set.

I have used a feed forward neural network in order to classify the yeast data set. I have used a stochastic gradient descent with back-propagation to train our ANN. Since the gradient decent methodology requires a differentiable activation function, I have used log sigmoid function. Log sigmoid function is an shaped function having a range between -1 to1.Because of it, I have modified the classes of target data set ranging from -1 to +1 since it's a nine class classification problem. Matlab neural network tool box (nntool) is used to do the necessary classification tasks.

In this exercise we built a classifier that can find the localization site of a protein in yeast. The yeast contains of 1484 proteins that have 8 attributes(features) each. This proteins are divided in 10 different classes.

For the performing classification there was constructed a 3-layer artificial neural network(ANN) and specifically a feed-forward multilayer perceptron. The first attribute which is only an accession number has been omitted since there is no significance of that attribute in prediction. We have converted target values to numeric values by setting an array of unique class names and then assigning to each class an index of an array.

The YEAST dataset which is a 1484×9 matrix, is first processed to remove accession number and then the remaining 1484×8 matrix taken as the input data. Out of these 1484 samples, 65% sample was used for training, and 35% for testing. Under supervised learning, the target of different inputs are in a random order to ensure proper training. The network architecture taken consisted of input layer which has 8 nodes, the hidden layer which has 3 nodes and output layer with 1 node.

The results of the first part are presented below in plots:

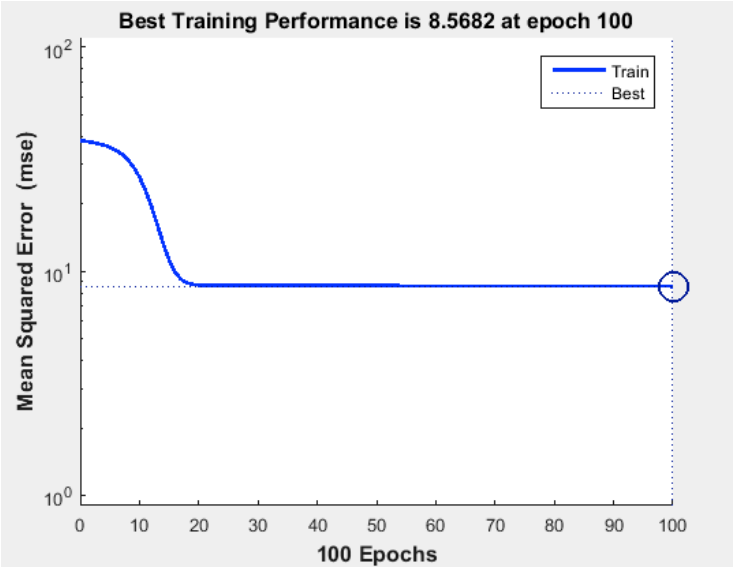


Figure 1.1.

Best performance was obtained on 100th epoch.

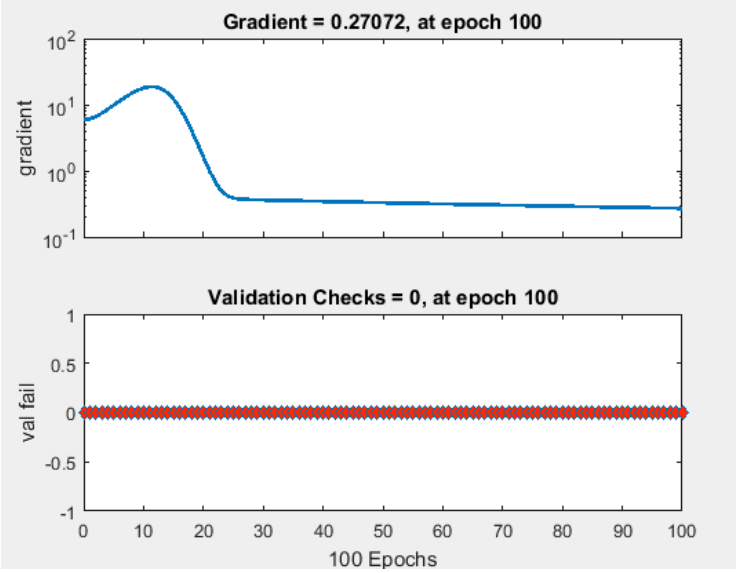


Figure 1.2.

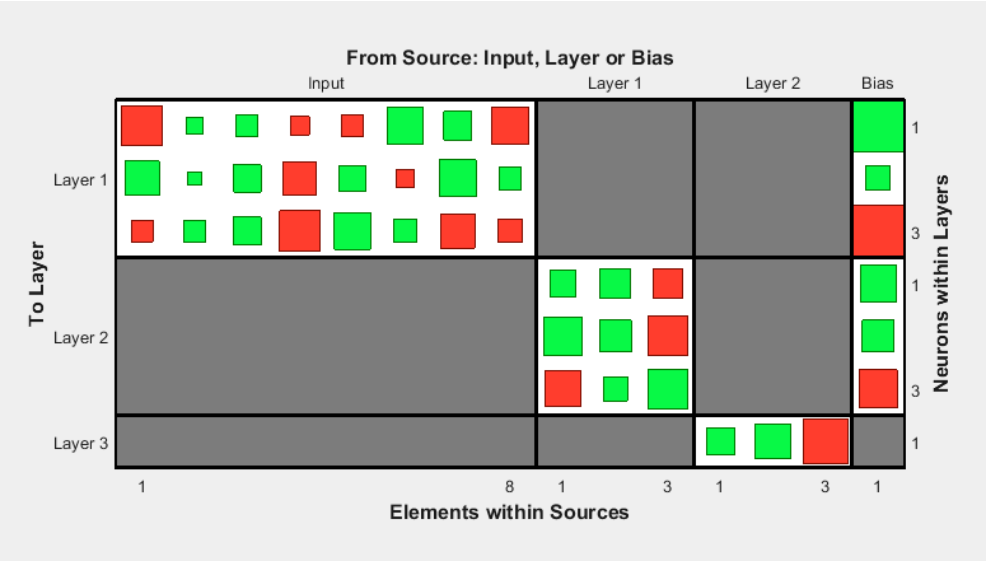


Figure 1.3.

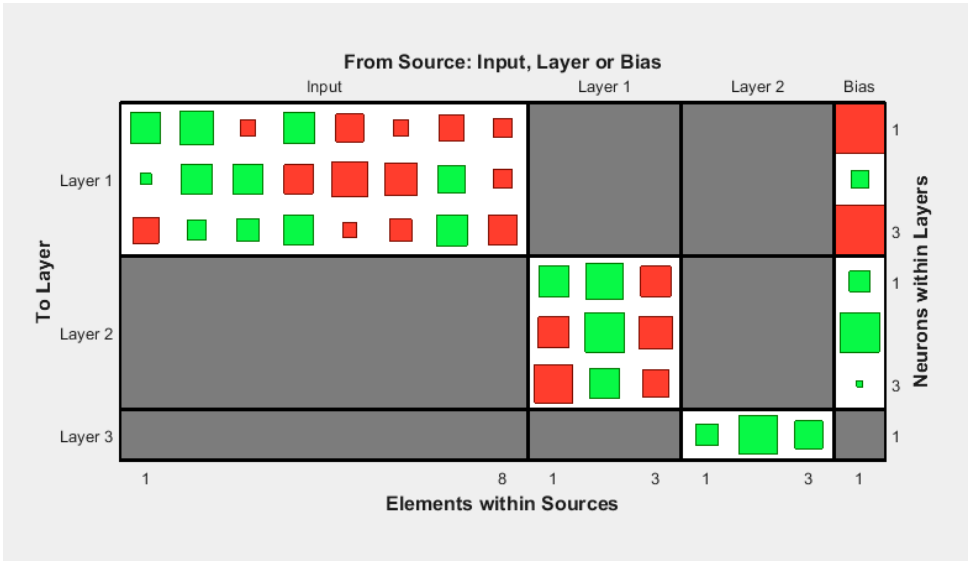


Figure 2.1.

In the second part of the classification we use all of our data instead 65%. We have provided a final activation function equations after training and the training error, which is 9.3063.

In the fourth part of this research we have incrising the number of hidden layers from 1 to 2 and then to 3. Then we increased the number of hidden nodes per layer from 3 to6, then to 9 and finally to 12.

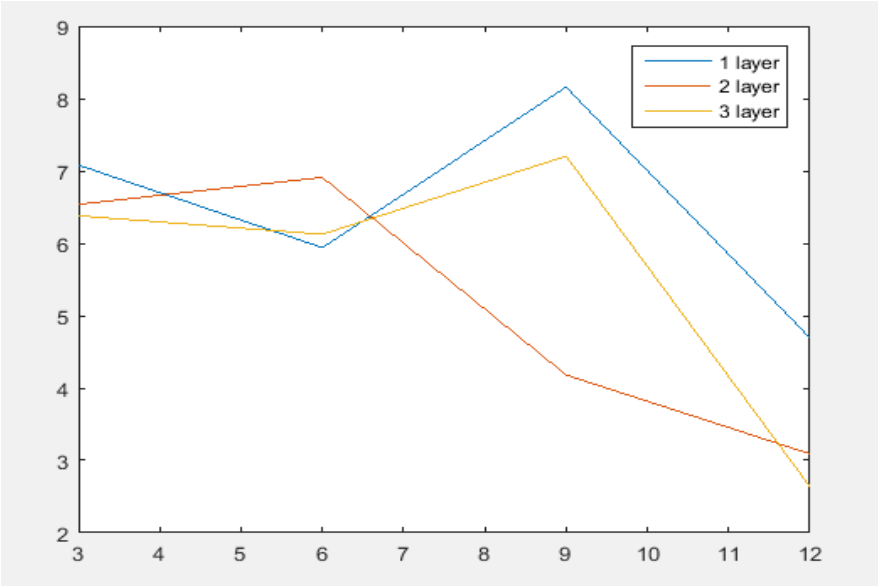
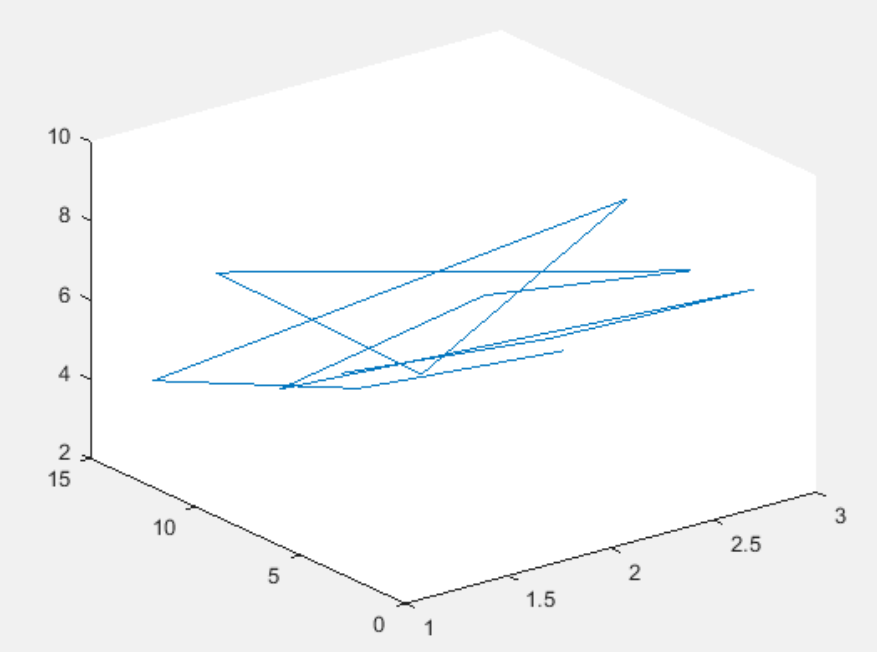


Figure 4.1.

Figure 4.2



In the fifth part we have an Unknown Sample 0.50 0.49 0.52 0.20 0.55 0.03 0.50 0.39.

As a result I have got that this sample belongs to 'ME3' class.

In addition I had to come up with a quantitative measure of uncertainty for each classification. Measure of uncertainty is a function of the number of equally probable outcomes. To help to quantitatively measure the interpretation uncertainty and the mis-classification risk are needed classification and estimation methods based on computational statistical techniques such as non-parametric Bayesian classification, bootstrap, and neural networks. I need quantifying uncertainty to assess risk, integrate data from different sources and estimate value of additional data. We can use quantifying tools of uncertainty like pdf estimate, that must from prior knowledge or available training data and the training data need to be extended or enhanced; or using rock physics models: methods, parametric approach, nonparametric approach, histogram.

So for our Unknown Sample I can count uncertainty with an entropy, which is a measure of the uncertainty of an experiment in which there are random events, and equal to the average of all the possible uncertainty of the outcome and it is $log_2 n = log_2 8 = 3$. Thus, the uncertainty introduced by each of the equally probable outcomes, is: $-\frac{1}{8} * log_2 (\frac{1}{8}) = -\frac{1}{8} * (-3) = 0.375$.