Introduction to Intelligent Systems: Project 2

Lee Avital and Mayur Sanghavi

May 2, 2014

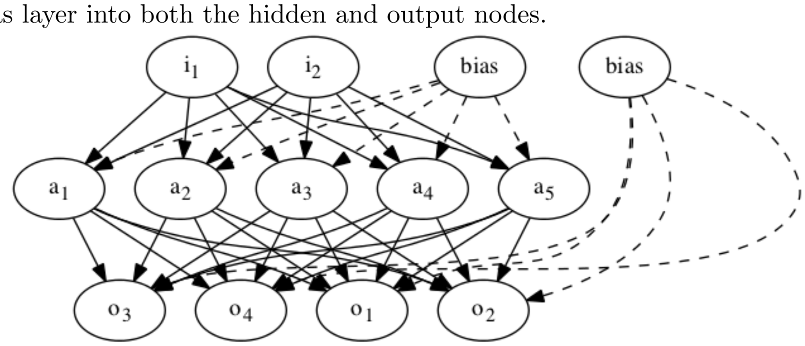
1 Classifier Designs

1.1 Random Forest Classifier

We used a random forest classifier that generated a variable number of decision trees that all went down three nodes deep. Since the input variables were continuous, we needed to generate random splits to emulate the decision boundaries necessary for using decision trees.

1.2 Multi-Layer Perceptron Classifier

We also used a 2-5-4 multi-layer perceptron (illustrated in full below) feeding a bias layer into both the hidden and output nodes.



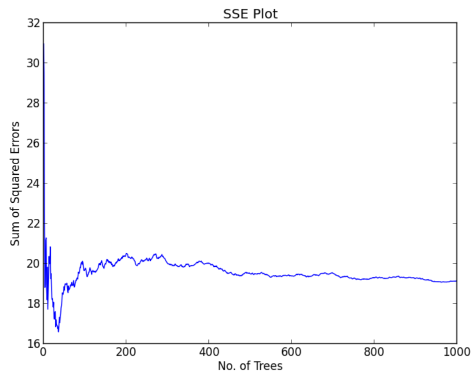
1.3 Hypothesis

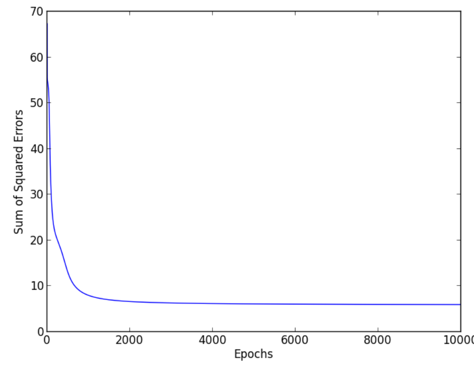
When running both classifiers for a short time (i.e. 100 trees for the random forest and 100 epochs for the multi-layer perceptron) we expected the multi-layer perceptron to outperform the random forest. This was because the multi-layer perceptron, by its nature, worked on continuous inputs while the random forest needed to extrapolate discrete decision classes.

We expected the random forest to outperform the multi-layer perceptron after running a long time (i.e. 10,000 trees or 10,000 epochs.) This was because the random forest was less likely to overfit than the multi-layer perceptron.

2 Data Sets

Below are the sum of squared errors for the random forest solution and the multi-layer perceptron solution. The random forest solution starts droping off at around 20% while the multi-layer perceptron drops off around 10%.





3 Results

Below are decision matrices

4 Discussion