Solution

In order to get good classification I implemented multiple algorithms.

Firstly divided train data in train and test data with 3999 and 500 elements respectively.

Accuracy was checked by close match is predicted for test data.

Single layer Neural Net could give around 61% , Multiple layer Neural Net should have worked better since I assumed that data might not be linearly separable but it didn't work predictions were either 0 or 1. Then I tried SVM since the results were not satisfactory with neural network , SVM gave good results.

Approach

- Single layer Neural Net (NN)
- Multiple layer Neural Net (with one hidden layer)
- Support Vector Machine

Single Layer NN

Started with single layer neural network where there are in total 14 features that are divided into two groups A and B which are vectors and similarly weightA and weightB for the weights of the features inside it. C denotes Choice.

It is a basic implementation where there is a layer of input and output and sigmoid function is used to calculate final value for the class , while training if the output is equal to expected value mentioned in data then it goes to next data element and if not then weights are updated appropriately.

Prediction for this implementation is in predictSNN.xls file

Multiple Layer NN

In this one hidden layer was implemented as hidden layer and back propagation was applied to update the weights.

But this code was not able to predict values correctly only 1 or 0's were predicted.

Support Vector Machine (SVM)

Since SVM is a heavy algorithm I used matlab library directly on the data and it could predict the output with very high accuracy.

Prediction is in predictSVM.xls