Machine Learning Assignment 3 Report

Manasvi Sagarkar - 20CS60R28
Shreyas More - 20CS60R34

November 2020

1 Introduction

In this assignment, we compare Support Vector Machines and Artificial Neural networks for classifying emails as spam.

2 Data Pre-processing

The dataset has only numerical features such as longest run of capital letters, average length of run of capital letters etc. So we did not need to handle categorical features. We dropped the data-points that had missing attribute values. For SVM, we used a min max scaler to scale the dataset.

3 Support Vector Machine

We have implemented Support Vector Machine using the inbuilt library of sklearn which is the svm.SVC. The question dictates us to make use of three different kernels namely *"linear"*, *"quadratic"* and *"radial basis function"*. These kernels are inbuilt into the support vector machines and we have made use of them accordingly.
The *"quadratic"* kernel needs to be set as *"poly"* kernel with degree as 2. We perform 5 fold stratified cross validation to calculate the train accuracy and then test the model on the test data to get the test accuracy. On setting the C value equal to 1, the accuracies obtained for *"Linear*, *"Quadratic* and *"Radial basis function"* kernel for the SVM are **90.99%**, **90.01%** and **92.18%** respectively when the C value was set to 1.
We also traint the SVM for the different kernels for different values of C and print their training and test accuracies in a tabular form.

4 MLP Classifier

For the Multi Layer Perceptron Classifier, we used the sklearn MLPClassifier. The output layer needs to return 1 is it classifies as spam and 0 otherwise. So

the output layer has 2 nodes, one for each label. The number of nodes in the input layer will be the number of features so in our dataset, we have 57 input nodes since we have 57 features.
The MLPClassifier in sklearn has the following parameters by default- one hidden layer with 100 nodes, activation function is 'relu',optimizer is 'adam', L2 penalty is 0.0001, batch size is 200 if there are atleast 200 samples, learning rate is 0.001, solver iterates until convergence or for maximum 200 iterations and convergence is said to be reached if loss does not improve by atleast 0.0001 in the last 10 iterations.
We set the optimizer to Stochastic gradient descent. Further, since 200 iterations were not sufficient for convergence for smaller learning rates, we set the maximum iterations to 2000. Then we varied the hidden layer sizes and learning rates to compare the different models.
We found that the best model was the one with two hidden layers with 2 and 3 nodes respectively. This model performed best when we used a learning rate of 0.0001 but was better than the other models for all sufficiently small learning rates (all learning rates except 0.1 and 0.01). The best accuracy of our best model was **79.78 percent** at learning rate 0.0001. We only varied the learning rates and hidden layer sizes and kept all other parameters fixed to the values mentioned above. A learning rate that is too high may cause divergent behaviour which explains why models trained with smaller learning rates generally outperformed models trained with learning rates of 0.1 and 0.01. A learning rate that is too low will update the loss too slowly can hence may stop at a suboptimal point. Thus we find that 0.0001 works best as a learning rate for our models. Two hidden layers mean that our model can represent functions with any kind of shape. Using too many nodes in the hidden layer leads to overfitting while using too few nodes leads to underfitting. The optimal point for our dataset appears to be achieved for the model with 2 hidden layers and 2 and 3 nodes respectively in each hidden layer.

5 Comparision between SVM and MLP Classifier

The MLP classifier gave an accuracy of **79.78%** while the SVM model with *"radial basis function"* kernel gave an accuracy of **98.1%** on test data when the C value was set to 80. Thus, SVM performed better than MLP classifier.