6000B Deep-Learning Project1

Student Name: HE, Jialiang
Student No.: 20460919

Data description

The project task can be described as a binary classification problem. The given training data consists of 3220 observations and 57 separated continuous features. Each observation corresponds to a training label.

Data preprocessing

- Feature selection

  From the feature distribution of our given data points, there are many zero values in each column. Therefore, it is necessary to consider the importance and redundancy of those features and conduct a feature selection process before building a model.

  After some experiments, I chose a method based on F-test, which can estimate the degree of independence between two random variables. To some extent, the multicollinearity problem can be solved. Finally I selected 40 features to do the modeling step.

- Data normalization

  It can be checked that the value range of the features are different. Especially the last two features whose range are up to hundred while others are single digits.

  To eliminate the effect of value range of features, I used normalization method to scale each feature into standard normal distribution of zero mean and one standard deviance. After the normalization process, the distribution of each feature will remain unchanged.

Classification modeling

Several basic classifiers had been tried but I preferred ensemble methods to make the classification accuracy as high as possible.

I chose to build a gradient boosting classifier. Gradient boosting is an ensemble method of several weak tree models. It builds an additive model in forward stage-wise fashion, and it generalize them by allowing optimization of an arbitrary differentiable loss function.

The number of boosting stages was set to be 300 to deal with the classification problem.

Cross validation

To validate our model performance, usually we should split the data set into training set and testing set. In this problem, I used 5-fold cross validation that the whole data set was divided by 5 parts and each time I used one part to be the testing set while others to be training set. So, the model should be built 5 times to get the average accuracy.

Metric

In this problem, I just use accuracy, which is the proportion of correctly classified data points, to measure the model performance.

Finally, I got around 95% accuracy from 5-fold cross validation.