EE527: Machine Learning Laboratory

Assignment 10

Due Date: 18 April 2022

1. Application of the *Perceptron* in Classification of Normal and Shouted Speech using MFCC features. These features are extracted from speech samples of a number of speakers uttering a few sentences normally or by shouting. The features are divided into train-test splits and are made available in two csv files(**use the dataset of Assignment 8**). You are tasked to learn a discriminative model to classify normal and shouted speech. This example uses Perceptron as a discriminative model. Consider the .csv file *"Train_file.csv"* containing *86060* instances of 61-dimensional arrays. The first *60* dimensions of the array contain the feature values for a particular instance and the last dimension contains its label. The label can be either *'0'* or '1'.

(a) The perceptron input $\boldsymbol{x} \in \mathbb{R}^{60}$ and predicted output $\hat{y} \in (0,1)$ are related as follows.

$$u = \boldsymbol{\omega}^T \boldsymbol{x} + b \ (\boldsymbol{\omega} \in \mathbb{R}^{60})$$
$$\hat{y} = (1 + e^{-u})^{-1}$$

Learn the weight vector $\boldsymbol{\omega}$ and bias b from the train dataset (*Train_file.csv*). Do not use any Scikit-Learn functions. Write your own functions for perceptron learning.

(b) Read *"Test_file.csv"* consisting of *21516* instances of 61 dimensional arrays. For each array, the first 60 dimensions contain the feature values for the test data and the last dimension contains its *actual label*. Predict the label of each data instance from the testing set using the learned perceptron and compare the *predicted* and *actual* labels. Report the *class-wise F1-scores* for both classes and the *overall accuracy*.

2. Download the MiniBooNE dataset from UIUC Machine Learning Repository. This dataset is imbalanced. First, sample n number of samples (say *11000*) from both the classes and set that aside as the test set D_{tst}. Consider the remaining imbalanced data as the training set D_{trn}.

(a) Use the perceptron module in Scikit-learn python toolbox to learn a perceptron from D_{trn}. Report the *class-wise F1-scores* of this perceptron on D_{tst}.

(b) Balance the dataset D_{trn} using KMeansSmote algorithm from the imbalanced-learn python tool box to generate the balanced training dataset D_{trn}^{b}. Use the perceptron module in Scikit-learn python toolbox to learn a perceptron from D_{trn}^{b}. Report the *class-wise F1-scores* of this perceptron on D_{tst}.

(c) Report your observations on the two test performances.

Reference: https://www.jmlr.org/papers/volume18/16-365/16-365.pdf

3. Consider the MNIST Handwritten Digit Recognition dataset used in earlier assignments.

(a) Use the perceptron module in Scikit-learn python toolbox to learn a perceptron to perform *10* category classification over the given dataset. Use *SoftMax* as the activation function for the *10* output nodes. Report the *class-wise F1-scores* and the *overall accuracy*.

(b) Fold back the weight vectors of the *10* perceptrons as images and visualize the same.

(c) Use the Multilayer Perceptron (MLP) module in Scikit-learn python toolbox to learn the MLP (with a single hidden layer) to perform *10* category classification over the given dataset. Use *SoftMax* as the activation function for the *10* output nodes. Experiment with the number of nodes in the hidden layer. Report the *class-wise F1-scores* and the *overall accuracy*. Report results for different number of nodes in the hidden layer. Also, compare the performance with experiment in performed in **(a)**.

(d) Try improving the performance of the MLP by adding more hidden layers. Experiment with the number of nodes in each hidden layer. Report your best performance (in terms of *class-wise F1-scores* and *overall accuracy*) on the MNIST dataset.