README File for Machine Learning Programming Assignment 1

Project group

1) Nikhil Lingutla (931-692-837)

Folder /LinearRegression/
   1. **LinRegBatchGD.m**

(Performs Linear regression using Batch Gradient Descent on the regression-train dataset and plots the convergence curve between number of epochs and Sum of standard square error. Also predicts the sum of squared error on regression-test dataset)

Folder /LinearRegression/
   2. **computeCostBatch.m**

(Computes the cost i.e sum of squared error provided inputs as feature vectors and target vectors and initial weights)

Folder /LinearRegression/
   3. **batchGradientDescent.m**

(Given feature vectors, target vectors, initial weights and learning rate, this functions computes the final weight vector using batch gradient descent algorithm)

Folder /LinearRegression/
   4. **LinRegStochasticGD.m**

(Performs Linear regression using Stochastic Gradient Descent on the regression-train dataset and plots the convergence curve between number of updates and Sum of standard square error. Also predicts the sum of squared error on regression-test dataset)

Folder /LinearRegression/
    5. **computeCostStochastic.m**

(Computes the cost i.e sum of squared error provided inputs as feature vectors and target vectors and initial weights**)**

Folder /LinearRegression/
    6. **stochasticGradientDescent.m**

(Given feature vectors, target vectors, initial weights and learning rate, this functions computes the final weight vector using stochastic gradient descent algorithm**)**

Folder /LinearRegression/
    7. **BatchStochasticComp.m**

(Compares the weights calculated by the Batch gradient descent and Stochastic gradient descent and plot the convergence curves to compare them visually**)**

Folder : /Perceptron/
    8. **batchPerceptron.m**

(Given the dataset, initial weight vector and learning rate, this function returns the weight vector learned for the decision boundary. It also returns the number of misclassified points per epoch in the form of a vector)

Folder : /Perceptron/
    9. **batchPerceptronDemo.m**

(Given a dataset, this functions plots the scatter plot of the two classes in the dataset, calculates the decision boundary using the batchPerceptron algorithm and plots it on the scatter plot. It also calculates the number of errors per epoches and plot the classification error against number of epoches)

Folder : /Perceptron/
    10.**votedPerceptron.m**

(This function takes in the dataset and the number of epochs and returns the weights in a matrix arranges column wise with the stability of each column is stored in counts matrix corresponding column**)**
Folder : /Perceptron/
    11. **votedPerceptronDemo.m**

(This function takes in a dataset and calculates the decision boundary using the voted perceptron algorithm. This function also compares the **Best weight** strategy and **Average weight** strategy by visualizations**)**