Report

Nour ElDin Khaled
Kareem Mokhtar
Narihan Ellaithy

Exercise 1 (Classification):

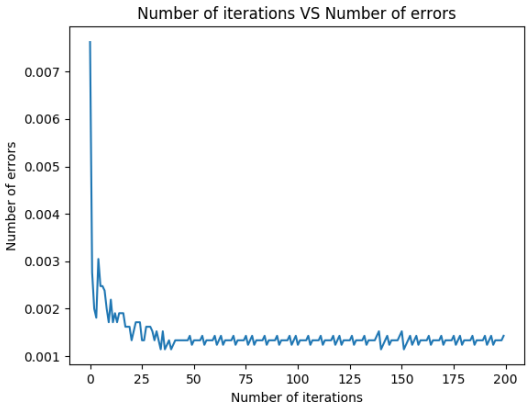
The data is classified into genuine or fraudulent based on columns V2 versus V11.

Steps:

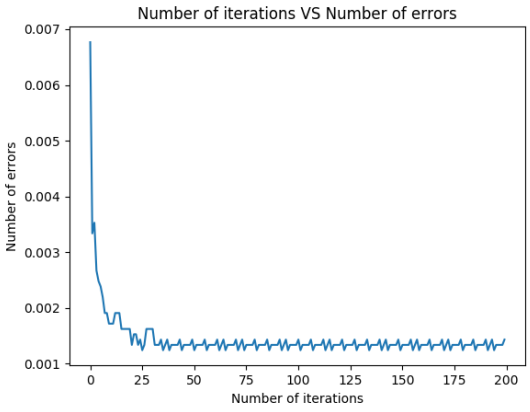
1) Reading the dataset and assigning it to data frame (df), the X values are read from the cvs as the values of columns v2 and v11 and Y is the first value from the left (class).
2) In order to multiply X by the weights we must add a column for the bias (W0), so we appended to X a column of ones with the length of X as rows and one column.
3) Generating weights randomly for w0,w1 and w2
4) Initialize error history array
5) Training the perceptron for a specified number of iterations in each one we loop over each row of X and calculate the dot product of current weights with rows of X. Then activation (step function) is created =1 if dot product is greater than 0 else it's a 0. Then we initialize term that is the multiplication of the learning rate with the activation function minus Y column multiplied by the rows of X. We subtract the term from the weights. The errors is incremented if the activation is not equal to the value of Y[i] else it's a 0.
6) ErrorHistory array is appended with the errors divided length(X) which is the mean squared error rule
7) Plot points V2 versus V11 with Y[i] ==1 (pass) or Y[i]==0 (fail)
8) Plotting classification line y axis in range -5 to 15 and x axis -w0 - w2 * y /w1 which is the equation of line
9) Setting titles and labels to plotting then showing it
10) Plotting mean error versus iterations in figure 2 from 0 to number of iterations and the error history to give us the convergence graph

Learning rate = 0.2 versus 0.8:
Using the learning rate 0.8 it converges at 25 iterations. But with learning rate 0.2, it converged after around 35-40 iterations. Therefore when the learning rate increased, it converged faster.

Learning rate 0.2



Learning rate 0.8



Exercise 2 (Regression):

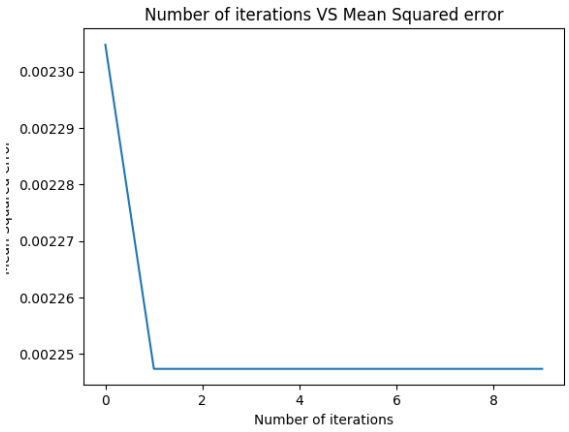
The data is classified into time versus distance based on columns ELAPSED_TIME versus DISTANCE and best fit line is drawn.

Steps:

1) Reading the dataset and dropping the null values
2) Normalizing the data as we can't use the given values because in the training process and duo to multiplying them with the weights the numbers get bigger exponentially.
3) Appending a ones column to multiply with bias
4) Splitting the data into a Training and Test sets with the ratio 0.8 and 0.2 respectively.
5) Generating random weights
6) Initialize Error History array
7) Training the perceptron for a specified number of iterations in each one we loop over each row of X and creates a dot product of random weights with rows of X. Then we initialize term that is the multiplication of the learning rate with the dot minus Y_train column multiplied by the rows of X. We subtract the weights from term. The square of the error is then added to the summation of the squared errors in this iteration.
8) ErrorHistory array is appended with the errors divided length(X_train) which is the mean squared error rule
9) Predict the output for the test data
10) Scatter points in X_test versus Y_test then plot the line that acts as the best fit line to predict the results of the input X_test
11) Set the titles and labels then show the plot
12) Plotting mean error versus iterations in figure 2 from 0 to number of iterations and the error history to give us the convergence graph.

This model converges after 1 iteration when using the learning rate 0.2 and 0.8

Learning rate 0.2



Learning rate 0.8