MACHINE LEARNING

BITS F464

ASSIGNMENT 1

Done By:

Rohan Maheshwari	2017B4A70965H
Keshav Kabra	2018AAPS0527H
Godhala Meganaa	2017B3A70973H

Assignment 1C: **Linear Perceptron**

Model Description :

Linear Perceptron is one of the basic supervised algorithm for supervised learning. It is used to classify the points by finding a hyperplane which separates it. To find the optimal weight vector which classifies the dataset, we use stochastic gradient descent approach.

Implementation:

The dataset contains different points with a target it belongs to , where 1 belongs to points which are classified as positive and 0 indicates that the points are classified as negative. We convert the dataset to a json file for easy implementation.

- We divide the dataset into train and test, the basis of dividing the points are selecting randomly 70% of the points as training dataset and 30% as testing dataset
- Initially start with a zero weights vector, and classify the dataset

  if
  x1*w1+x2*w2+x3*w3+.....+xn*wn > 0
  Predicted value = 1
  else
  x1*w1+x2*w2+x3*w3+.....+xn*wn < 0
  Predicted value = 0
- Compare the predicted values with the target value, count the points which are wrongly classified
- Objective function is to minimize the number of miss classifications, which can be expressed as

  Min sum(-tn*Wt*X) over n data points

- We alter the weights until we find zero miss classifications after each iteration, we take 1000000 iterations as the maximum limit of iterations to proceed.
- To alter the weights we use the approach of stochastic gradient descent, we take learning rate to be 0.2 and alter the weights.
- When we reach the point where the miss classifications are zero , stop altering the weights vector since we found the optimal weights vector which classifies the training dataset correctly.Let this be denoted by W.

Results:

- **Dataset-1:**
  Given dataset contains 1372 data points, after randomly splitting the dataset into train and test, we find the optimal weights at 5th iteration , where we found that the no of miss classifications are zero and the data is perfectly linearly separable. We use this weights vector W on the testing dataset and predict the class it belongs to , we were able to predict the data-points with **100% accuracy** in this case.

```
accuracy of epoch 0 -----> 97.142857
loss of epoch 0 is 2.857143

no of misses are %d 6
accuracy of epoch 1 -----> 99.142857
loss of epoch 1 is 0.857143

no of misses are %d 2
accuracy of epoch 2 -----> 99.714286
loss of epoch 2 is 0.285714

no of misses are %d 1
accuracy of epoch 3 -----> 99.857143
loss of epoch 3 is 0.142857

no of misses are %d 4
accuracy of epoch 4 -----> 99.428571
loss of epoch 4 is 0.571429

no of misses are %d 1
accuracy of epoch 5 -----> 99.857143
loss of epoch 5 is 0.142857

no of misses are %d 0
getting the weights
2.0
-0.043279024999999804
1.9492456125000004
10.666949226999998
Choose Files No file chosen
printing test accuracy
100.0
```

- **Dataset-2:**

  Given dataset contains 1000 data points, after randomly splitting the dataset into train and test, we find the optimal weights after running for 1000000 iterations , where we found that the no. of miss classifications are non-zero and the data is not linearly separable. We use the weights vector with minimum miss classification until 1000000 iterations and use this W on the testing dataset and predict the class it belongs to , we were able to predict the data-points with greater than **<u>97% accuracy</u>** in this case.

```
no of misses are %d 4
accuracy of iteration 345 -----> 99.583333
loss of iteration 345 is 0.416667

no of misses are %d 6
accuracy of iteration 346 -----> 99.375000
loss of iteration 346 is 0.625000

no of misses are %d 2
66.0
-52.980812749998286
-33.42601999999575
-40.11883199999988
-11.702656500000074
Choose Files  No file chosen          Upload widget is onl
printing test accuracy
97.33009708737865
```

  Hence we conclude that dataset 1 is **more linearly separable** than dataset 2 because we got a 100% accuracy which implies that we were able to separate the classes perfectly which was not the case with the second dataset.

Limitations of linear Perceptron:

- The output values are binary, can take only two values, hence cannot classify the dataset with more than 2 features
- The model can only classify the points that are linearly separable, If there exists no hyperplane, then the model cannot classify those points accurately, the model does not converge for non linearly separable sets.