BITS F464 - Machine Learning

Assignment-1C: Linear Perceptron

Made by -

Tanay Gupta (2018AAPS0343H)
Anusha Agarwal (2018A3PS0032H)
Kriti Jethlia (2018A7PS0223H)

Model Description

- The model is based on the idea that we iterate over the training examples and check if they are correctly classified for the current weights of the decision boundary. If they are correctly classified then the weights remain unchanged. However if they are incorrectly classified then the weights are updated accordingly. This continues till all the examples are classified correctly.

- It can be seen as an error minimization algorithm wherein the error function is the perceptron criterion

  $$E_p(w) = -\Sigma w^T x_n t_n$$

  Therefore we try to minimize $w^T x_n t_n$ over all misclassified examples.

- We apply stochastic gradient descent algorithm and updates are according to

  $$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E_p(w)$$

- If the dataset is linearly separable then we will find an exact solution in a finite number of steps. However if they aren't separable then the perceptron algorithm will never converge.

- Also it is important to note that each correction step of the perceptron algorithm reduces the contribution towards the total error from that particular misclassified example and in no way guarantees to reduce the total error function. The new weight may even misclassify some previously correctly classified examples.

Implementation

- First each dataset is split into training and testing sets (70:30 split) and the bias factor of 1 is added to them.

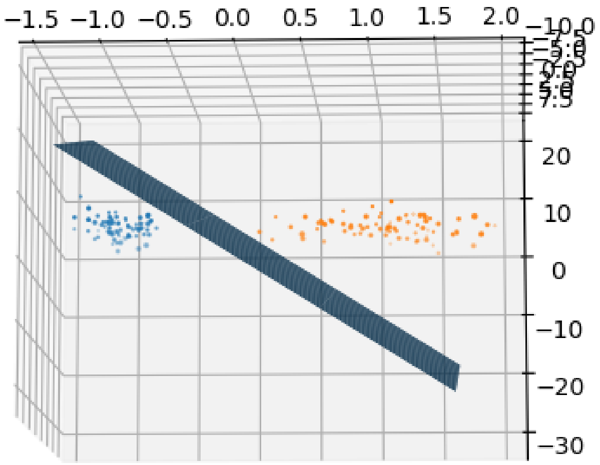
- The initial weights are set up as zeroes and run over the training set examples multiple times. The expected value and the predicted values are calculated and then compared. The weights are updated accordingly. If all the points are classified correctly then the algorithm stops or else it continues to iterate over the training set until either it converges or has reached the maximum number of iterations. We have set the maximum number of iterations to 10^6 for our implementation.

- Once we get weights from the training set we predict the values for the testing set and calculate the accuracy.

Observations

- Dataset 1 :
  - Training accuracy : 99.375%
  - Testing accuracy : 98.786%
- Dataset 2:
  - Training accuracy : 100%
  - Testing accuracy : 100%

Clearly Dataset 2 is more linearly separable.



Separation boundary for dataset 2

Limitations

- One of the major limitations of perceptron algorithm is that it will never converge if the data is not linearly separable.

- Also the algorithm is not easily generalizable and may not work well for classes greater than 2

- Also for a linearly separable dataset, there can be multiple solutions, the one found depends on the initialization of weights and the order in which data points are presented.