Perceptron for Handwritten Digits Recognition

Sourabh Dilip Powar

February 5, 2018

Abstract

Perceptron Learning algorithm (PLA) is machine learning algorithm, for supervised learning of binary classifiers where functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector

1 Introduction

Perceptron algorithm for Handwritten Digits Recognition takes traning data consisting label and 16x16 pixel image. MNIST (Modified National Institute of Standards and Technology) handwritten digit database is used for both training and testing the learning algorithm. Input are the images of digit 1 and 5 with label 1 and 5 correspondingly. Algorithm plots the image of the digit, calculate two features symmetry and intensity based on the input pixels and plot the graph of symmetry verses intensity. Learning algorithm learns weight and classify data based on features. .

2 Implementation

2.1 Plot the image

This function is used for plot image and save it.It tkes two images from train data with shape (2, 16, 16). The shape represents total 2 images and each image has size 16 by 16. Creates a subplot using matplotlib library and save the image with name as Image.png shown below.

2.2 Plot the Features

This function is used for plot train features with shape (1561, 2). The shape represents total 1561 samples and each sample has 2 features. Creates a graph using matplotlib library and save the image with name as Feature.png shown below. This function is used for plot a 2-D scatter plot of the features and save it.

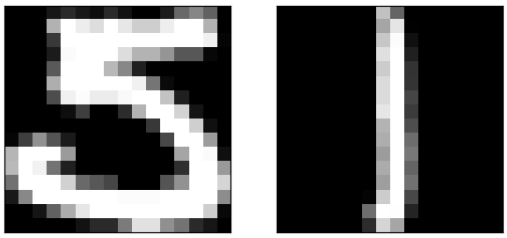

Figure 1: This is the image was used as Input to the PLA algorithm.

2.3 Plot the Test Results graph

This function is used for plot the test data with the separators and save it. Creates a graph using matplotlib library and save the image with name as Result.png shown below. This function is used for plot a 2-D scatter plot of the test data's label with shape (424,1) with 1 for digit number 1 and -1 for digit number 5.

2.4 Calculate the Accuracy

Accuracy of both training and test data is calculated based on misclassified images. First 5 iteration will calculate accuracy by changing the number of iteration of learning algorithm and nest 5 cases will calculate based of varying learning rate. The experimental results for given training and test data are shown in Accuracy figure below.

3 Conclusion and Results

The accuracy and result plot shows that we cannot achieve 100 percent. This is because the data set needed for the PLA classification must be linearly separable. PLA is classic binary classification algorithm. For the results in above experiment shows success in implementing the four steps to conclude for the PLA implementation. Accuracy for the training and test dataset provided by proposed implementation are high for finite iterations taken on non linear separable dataset.

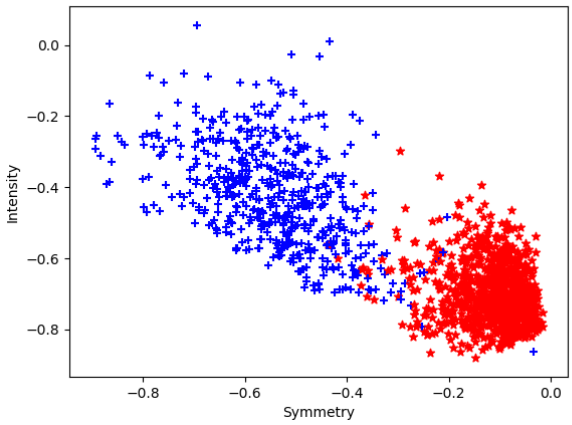

Figure 2: This is the graph of features symmetry and intensity of training data for PLA algorithm.

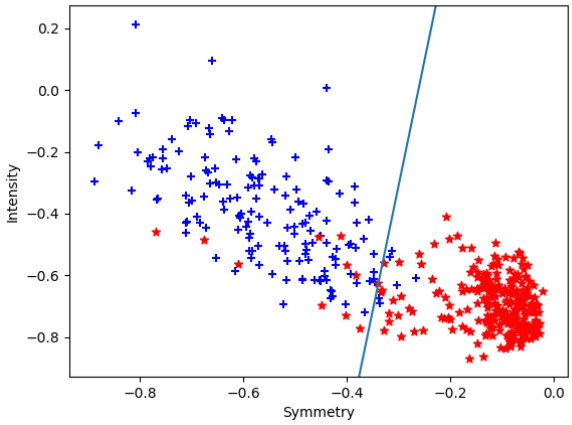

Figure 3: This is the result of test data with separator.

```
========== RESTART: C:\Assignment1\code\main.py ==========
play with data done!
play with features done!
Case 1 train accuracy:0.981422  test accuracy: 0.959906
Case 2 train accuracy:0.976938  test accuracy: 0.948113
Case 3 train accuracy:0.976938  test accuracy: 0.948113
Case 4 train accuracy:0.976297  test accuracy: 0.948113
Case 5 train accuracy:0.976297  test accuracy: 0.948113
Case 6 train accuracy:0.976297  test accuracy: 0.948113
Case 7 train accuracy:0.976297  test accuracy: 0.948113
Case 8 train accuracy:0.976297  test accuracy: 0.948113
Case 9 train accuracy:0.976297  test accuracy: 0.948113
Case 10 train accuracy:0.976297  test accuracy: 0.948113
accuracy test done!
play with result done!
>>>
```

Figure 4: This is accuracy of 10 cases for training and test data provided.