CS 534 Machine Learning Programming Assignment - 1

Project group

1) Nikhil Lingutla (931-692-837)

Linear Regression

Using Batch Gradient Descent

1) Initial weight(seed) used is a zero vector
   Learning rate is fixed as 0.01
 The weight vector learned from the training set is as follows

$$W_0 = -3.407620e-01$$
$$W_1 = -4.033085e-01$$
$$W_2 = 3.706733e-01$$
$$W_3 = 1.980380e+00$$

The convergence of the algorithm can be visually verified by following plot

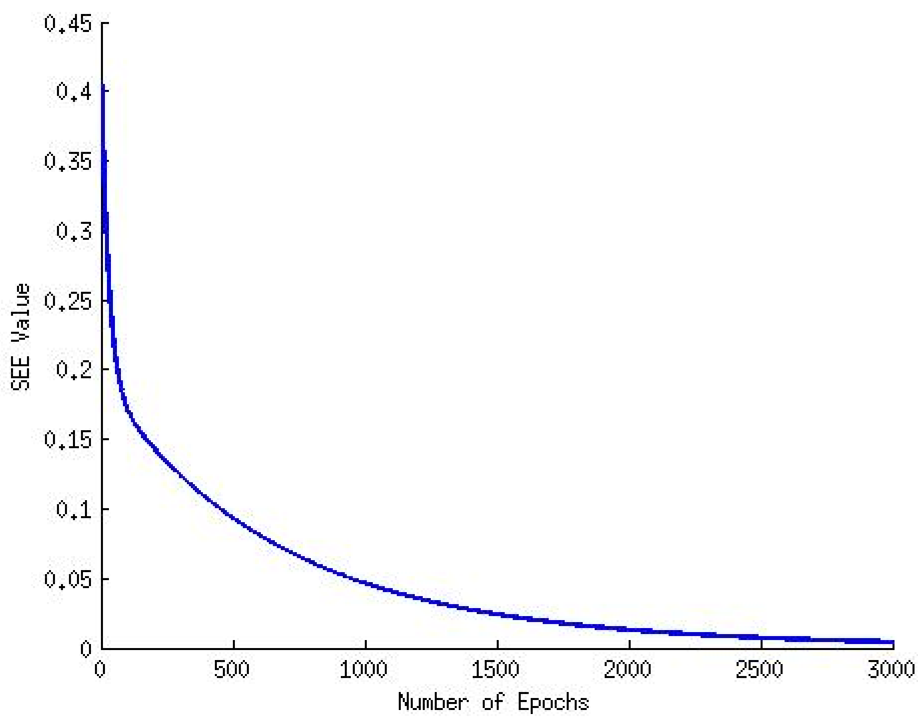

Fig 1.1 : Plot between SEE values and Number of Epochs for training set using Batch Gradient descent

Sum of Squared Error on test dataset using Batch Gradient Descent

The sum of squared error for the testing set after learning from the training set is as follows **4.184672e-03**

Using Stochastic Gradient Descent
Initial weight(seed) used is a zero vector
Learning rate is fixed as 0.01
Number of epoches = 50
The weight vector learned from the training set is as follows

$$W_0 = \text{-5.784571e-01}$$
$$W_1 = \text{-3.527796e-01}$$
$$W_2 = \text{5.028406e-01}$$
$$W_3 = \text{2.252782e+00}$$

The convergence of the algorithm can be visually verified by following plot

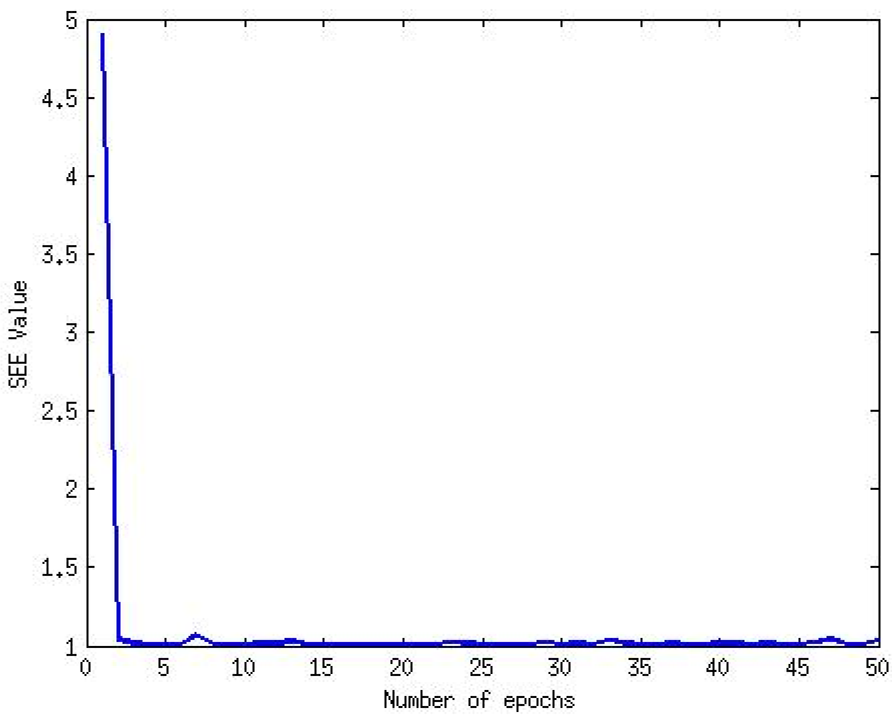

Fig 1.2 : Plot between SEE values and Number of Epochs for training set using Stochastic Gradient descent

Sum of Squared Error for the test set using stochastic Gradient Descent is as follows

The sum of squared error for the testing set after learning from the training set is as follows **3.877308e-01**

2) Convergence test for Batch and stochastic Gradient descent

Number of epoches = 500
Initial weight (seed) vector = ones vector
Learning rate = 0.01

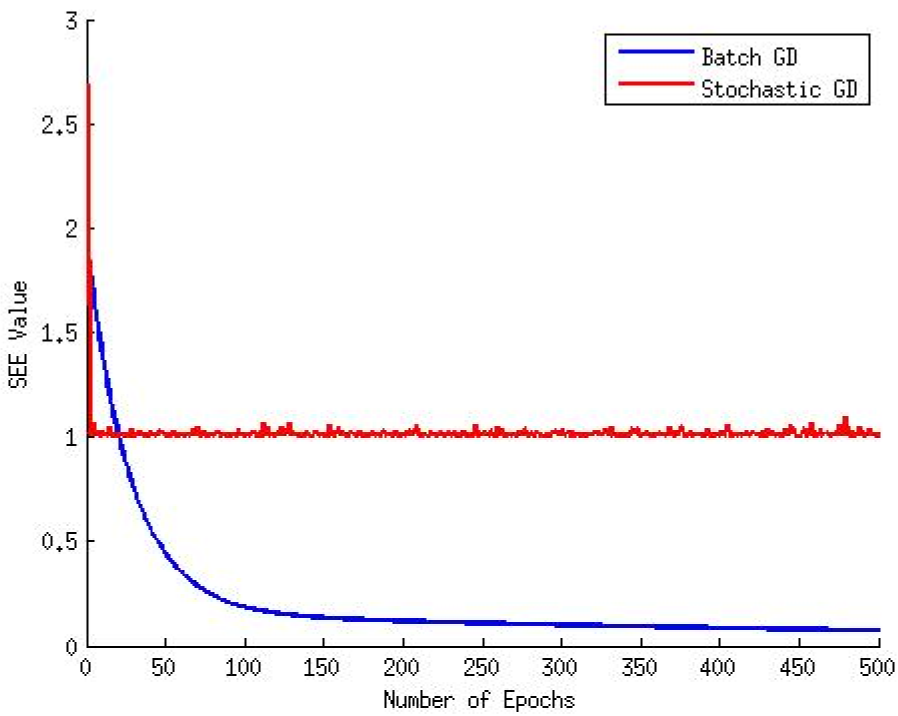

Fig 1.3 Plot showing the convergence of Batch gradient descent and Stochastic Gradient descent

Number of epoches = 500
Initial weight (seed) vector = zeros vector
Learning rate = 0.01

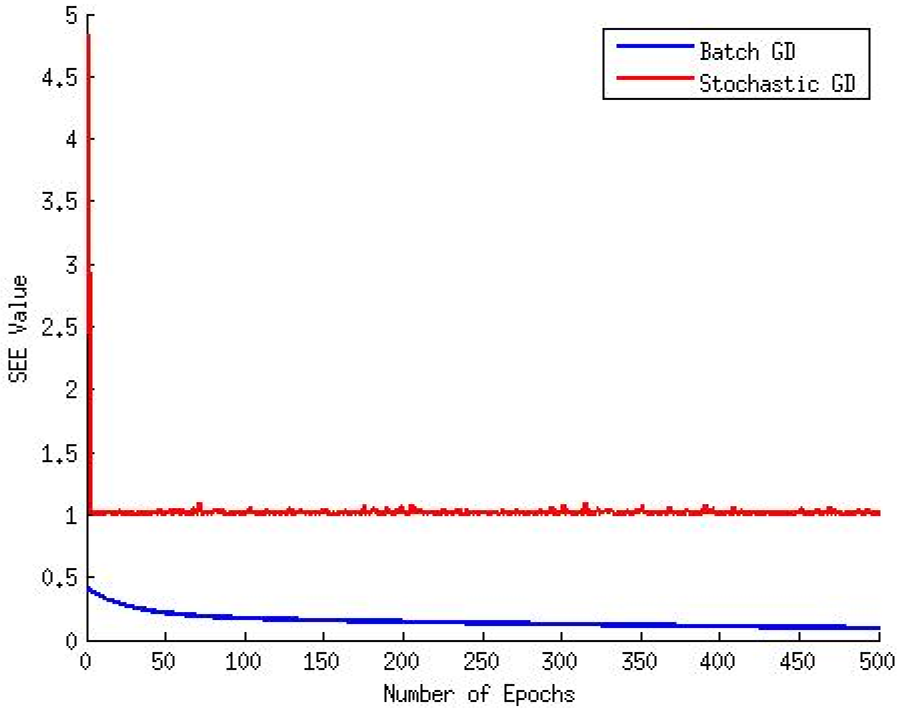

Fig 1.4 Another plot showing convergence comparison of Batch and Stochastic Gradient descents

Number of epoches = 500
Initial weight (seed) vector = ones vector
Learning rate = 0.03

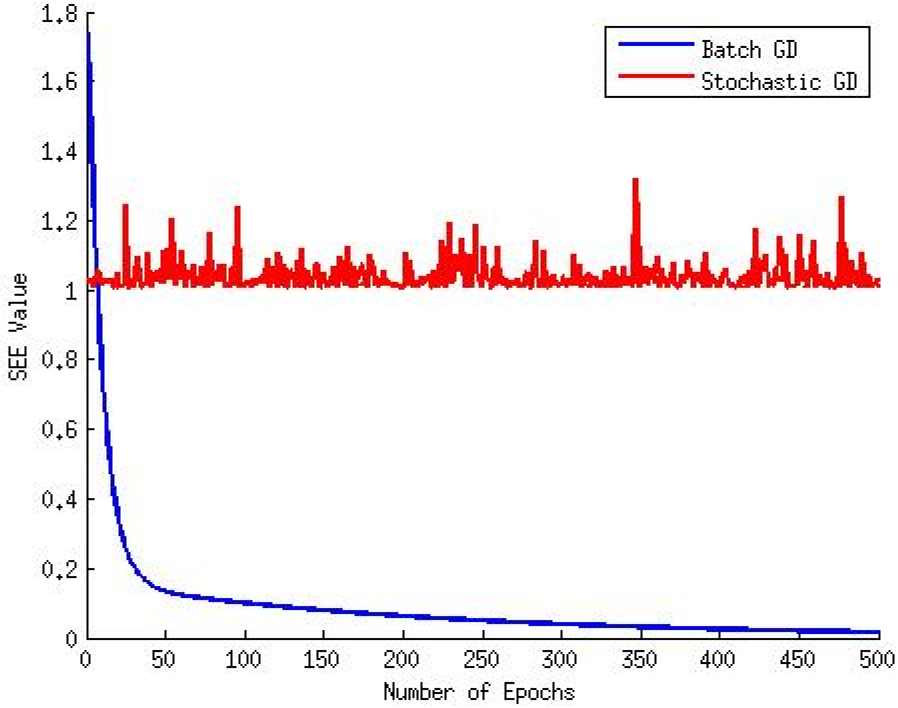

Fig 1.5 Another plot showing convergence comparison of Batch and Stochastic Gradient descents

Conclusion : From the above convergence plots, it is very obvious that the Batch Gradient descent converges closer to the minimum than Stochastic Gradient descent for the given regression-train dataset

2) Perceptron

1) **classification error on the training set as a function of the number of training epoches**

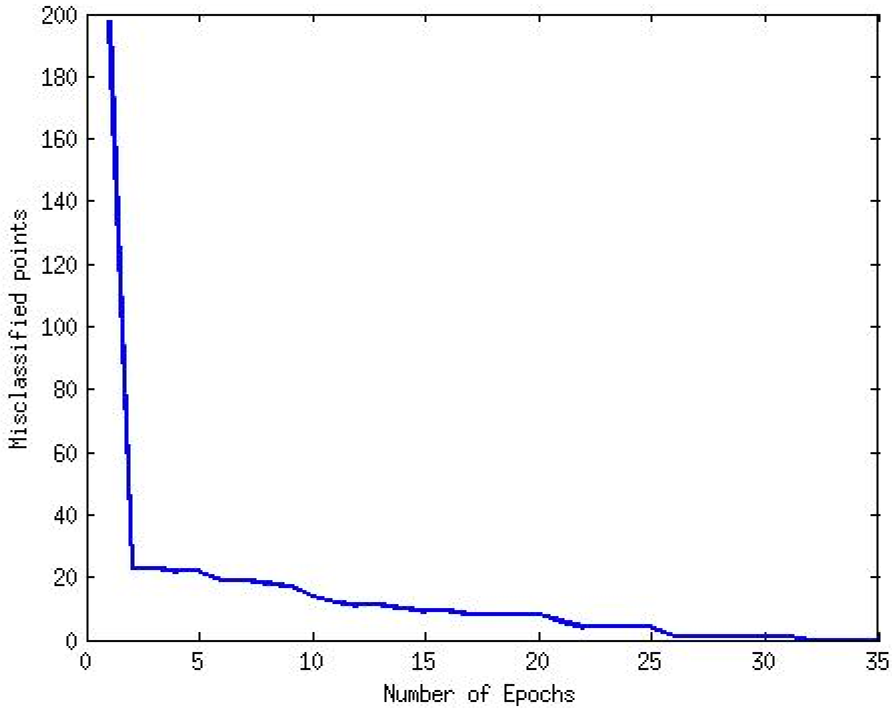

Fig 2.1.1 Plot showing the decrease in Number of misclassified points as the number of epoches increases.

The weights learnt by the batch perceptron algorithm are as follows

weights =

$W_0 =$ 1.5000
$W_1 =$ -0.6289
$W_2 =$ -0.5215

2) **Visualization of Data (Scatter plot with Decision boundary)**
A scatter plot of the training data with decision boundary

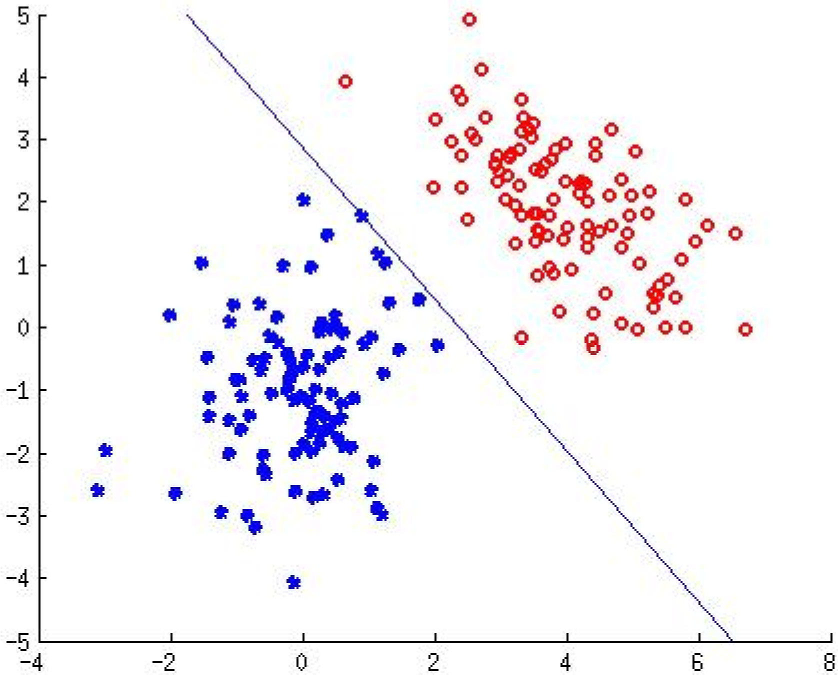

Fig 2.1.2) Linear boundary plotted between two different classes

Voted Perceptron

1) A plot of the classification error on the training set as a function of the number of training epoches

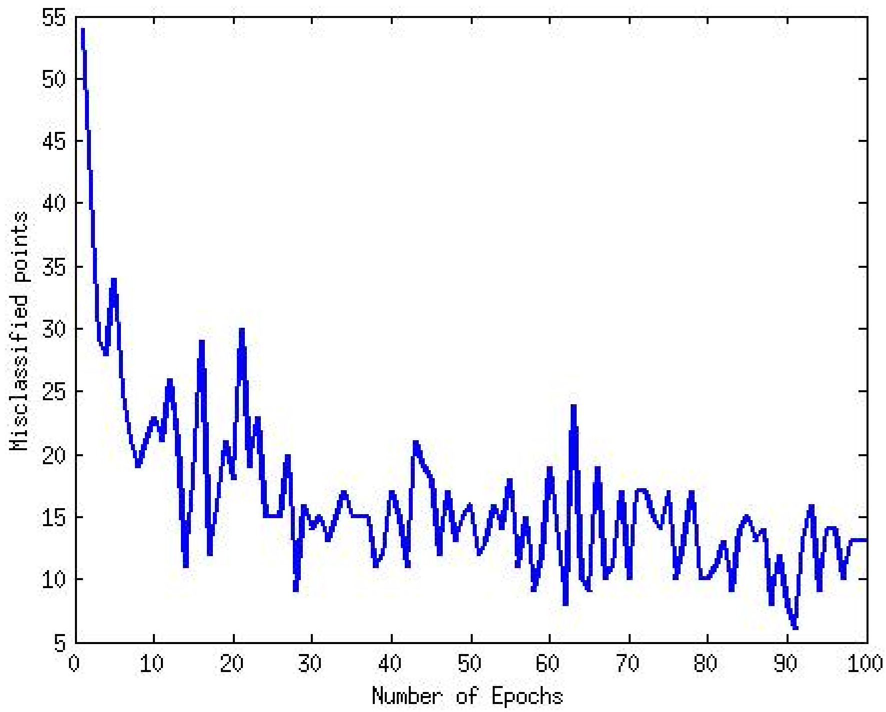

Fig 2.2.1 Plot showing the misclassification error versus epoches (100 epoches)

2) visualizing the final decision boundary produced by voted perceptron algorithm (highest count weight)

Weights (Best count)

$W_0 = 117.0000$
$W_1 =$ -10.3000
$W_2 =$ -41.9000

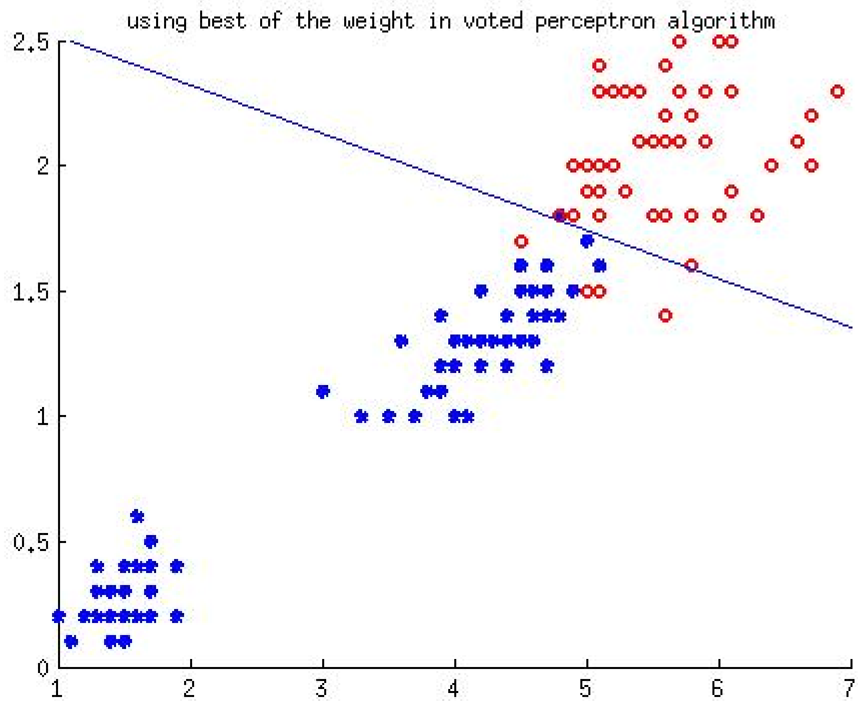

Fig 2.2.2) Final boundary obtained using the strategy of weight corresponding to highest count

3) visualizing the final decision boundary produced by voted perceptron algorithm (average of weights)

Weights (Average)

$W_0 = 1.2713$
$W_1 =$ -0.0907
$W_2 =$ -0.5148

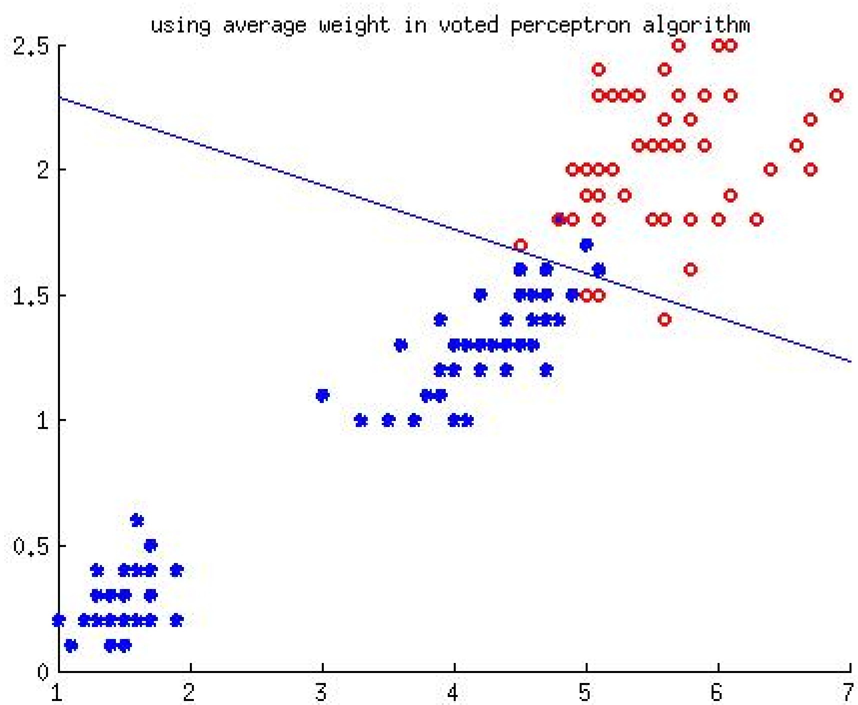

Fig 2.2.3) Decision boundary obtained by calculating the average of weights

Comments : Based on the decision boundary plotted using best weights and average weights, we observed

1. The decision boundary obtained by best weight strategy correctly classified a class completely and includes some misclassifications of the other.
2. On the other hand using the average weighted strategy misclassification have been made from both the classes.
3. The above two approaches are not same and which one to follow depends on the user requirement.