ASSIGNMENT-3 (Part 1)

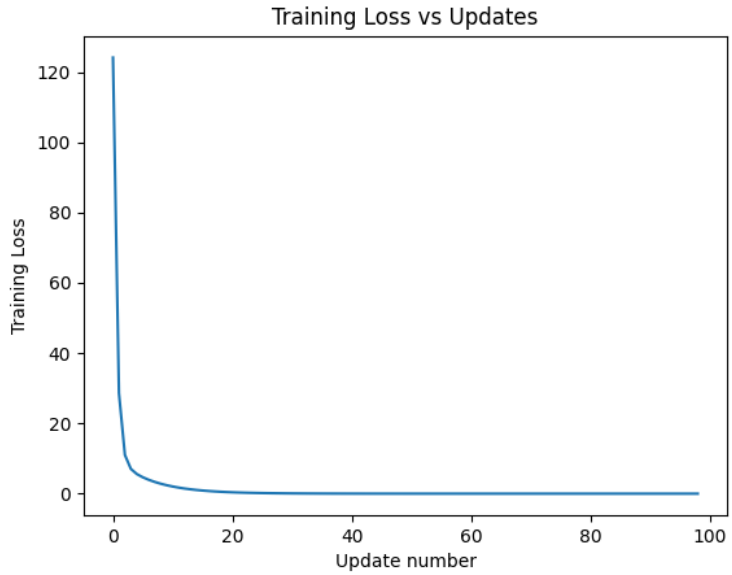
Submitted by- Vandit Sharma **Roll No.-** 17EC10060

Title- Implementing various Gradient Descent Methods

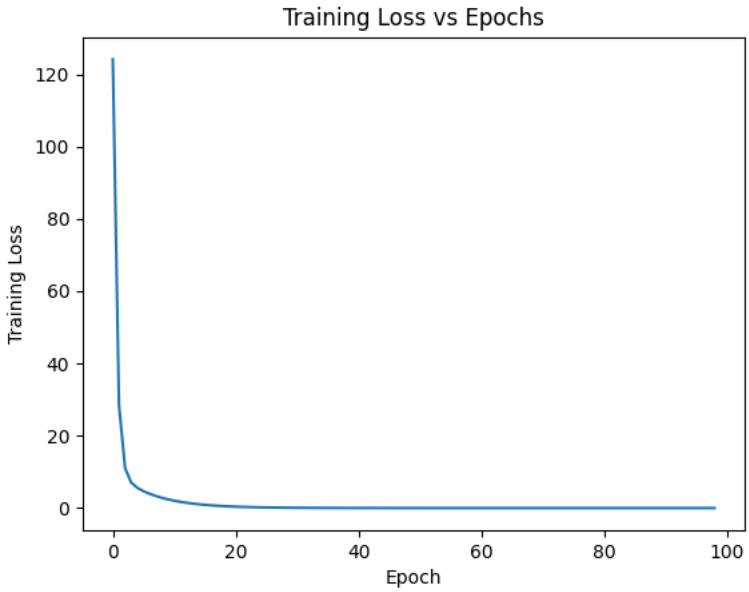
Results –

- **Batch Gradient Descent**

  A) Plot of training loss with each update

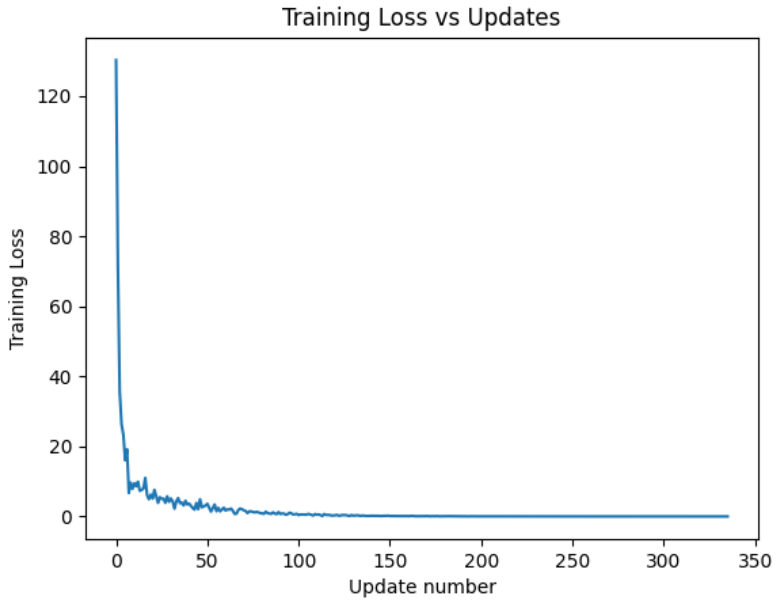


  B) Plot of training loss versus epochs

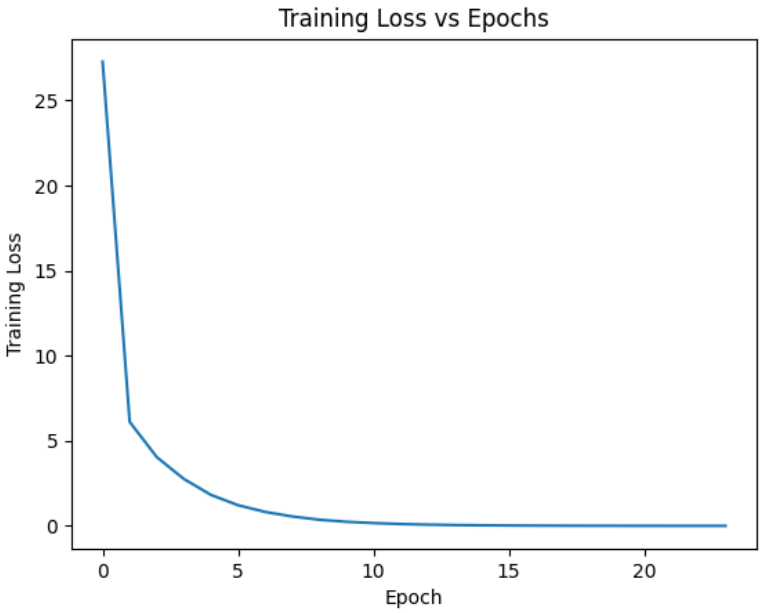


- **Mini Batch Gradient Descent**

  A) Plot of training loss with each update



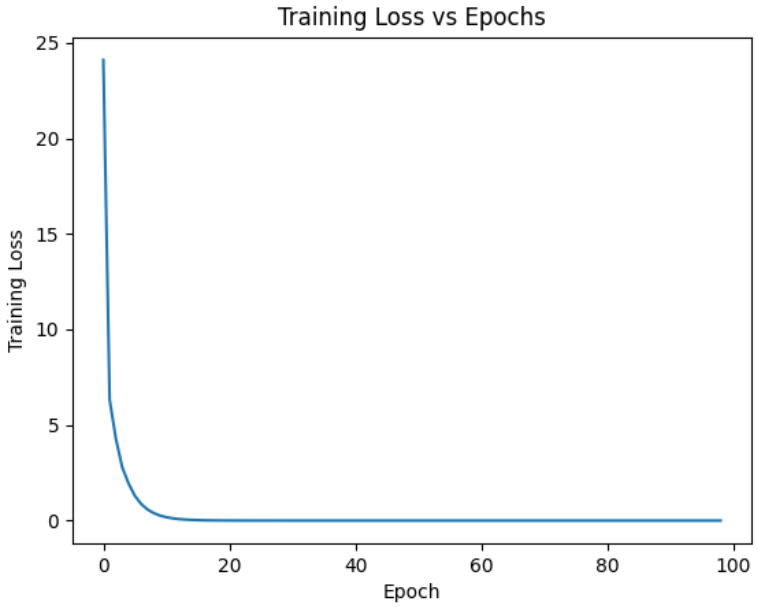
  B) Plot of training loss versus epochs



- **Stochastic Gradient Descent**

  A) Plot of training loss with each update

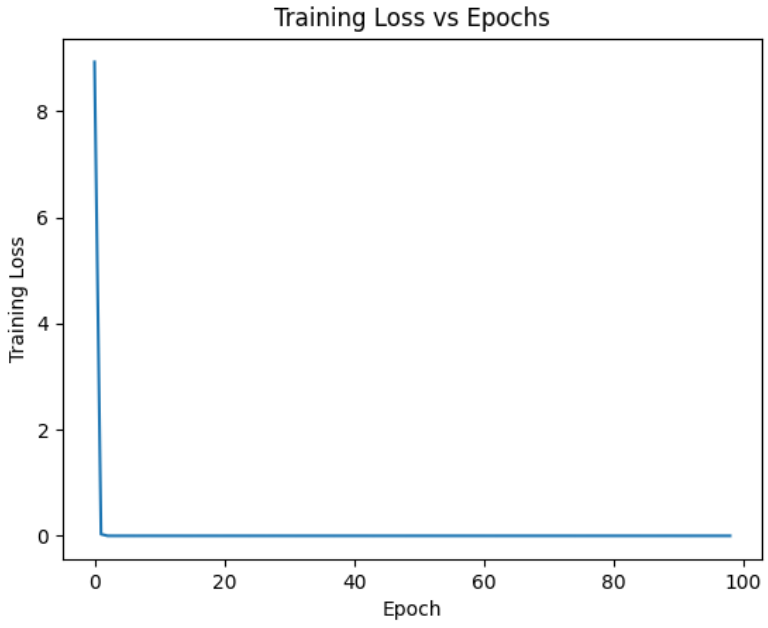
  B) Plot of training loss versus epochs



- **Momentum Gradient Descent**

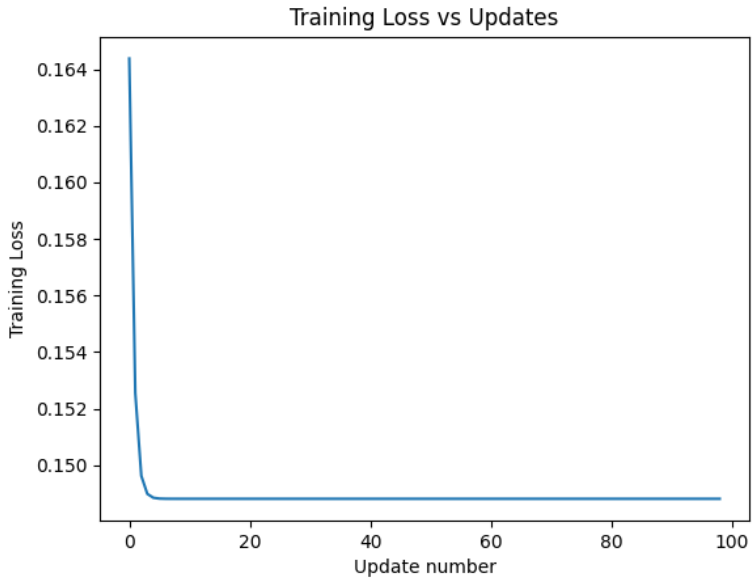
  A) Plot of training loss with each update

  B) Plot of training loss versus epochs

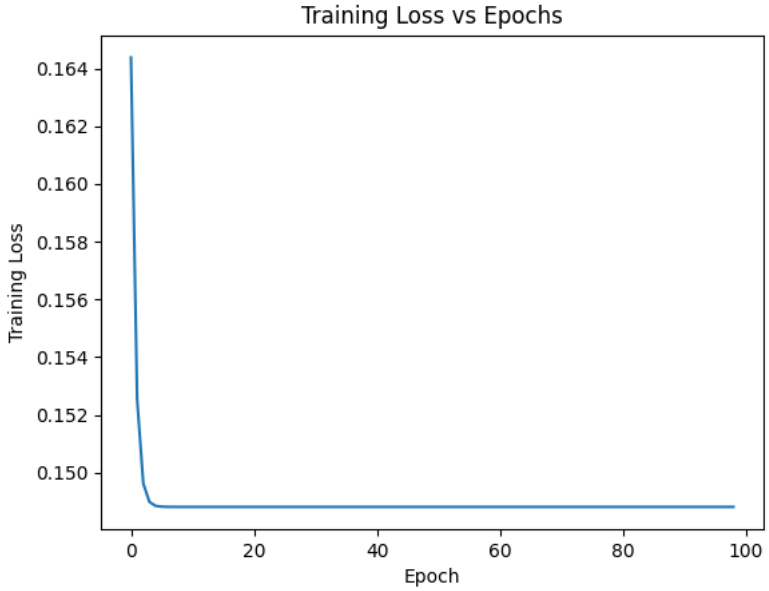


- **Adam Gradient Descent**

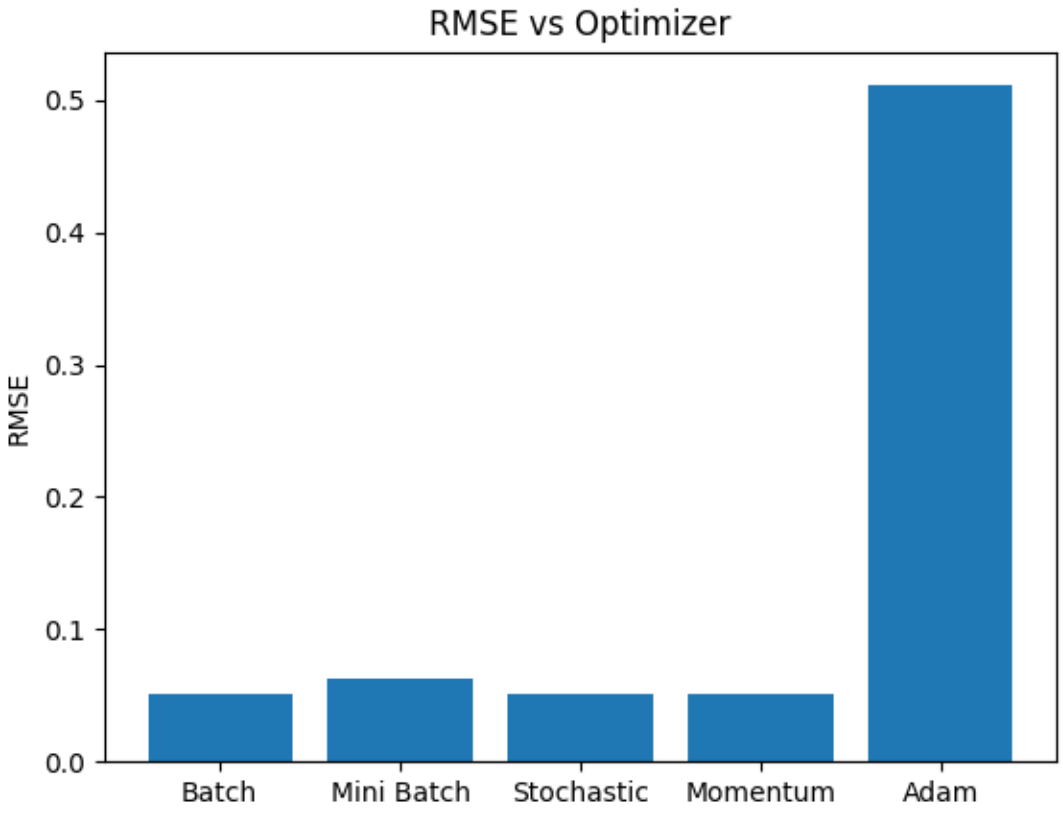
  A) Plot of training loss with each update



  B) Plot of training loss versus epochs



Plot of test Root Mean Square Error (RMSE) vs optimizer



Discussion –

- The training loss decreased sharply with both number of updates and number of epochs
- In cases where the number of updates occurring was not the same as the number of epochs, such as Mini Batch Gradient Descent and Stochastic Gradient Descent, the training loss vs update plot had many sharp jumps (represented by peaks) in the curve
- Such sharp peaks were absent in algorithms where we considered the entire training set before making updates, such as Batch Gradient Descent
- The peaks were evened out for all cases in the training loss vs epoch plot, as the training loss consistently went down after each epoch
- From the plots, Momentum Gradient descent was found to converge the fastest. This is evident from the sharp fall in training loss

- Mini Batch Gradient descent algorithm was found to be the fastest during a test run of my algorithm
- The time taken by each algorithm is given below-

Algorithm	Time taken
Batch Gradient Descent	**0.044876813888549805**
Mini Batch Gradient Descent	**0.015956401824951172**
Stochastic Gradient Descent	**0.11768555641174316**
Momentum Gradient Descent	**0.10572195053100586**
Adam Gradient Descent	**0.06681418418884277**

- Adam Optimization was found to have the largest Root Mean Square Error