ASSIGNMENT-3 (Part 2)

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

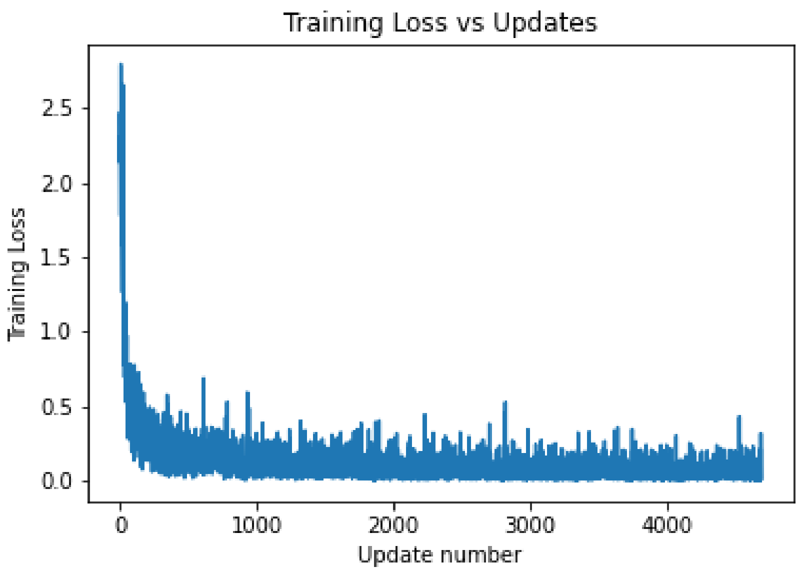
Title- Image classification using MNIST handwritten digits

Results –

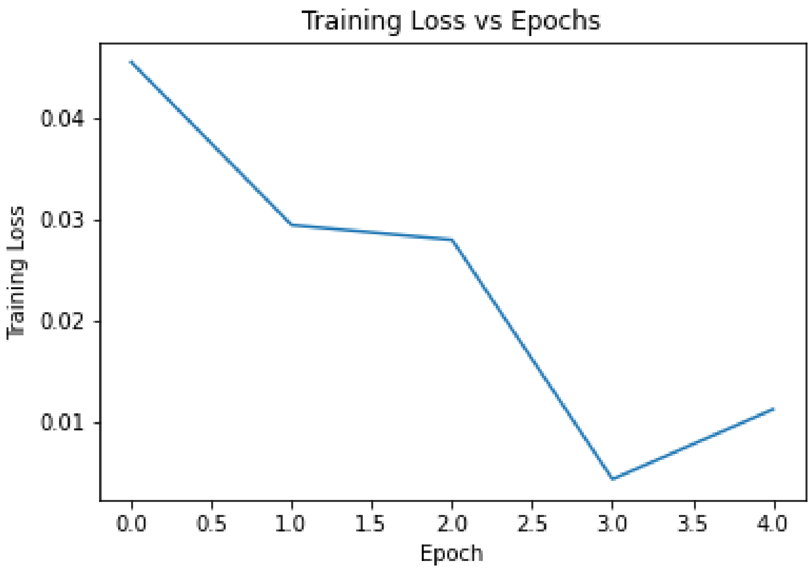
- **Mini Batch Gradient Descent**

  1) Learning rate = 0.2

  A) Plot of training loss with each update

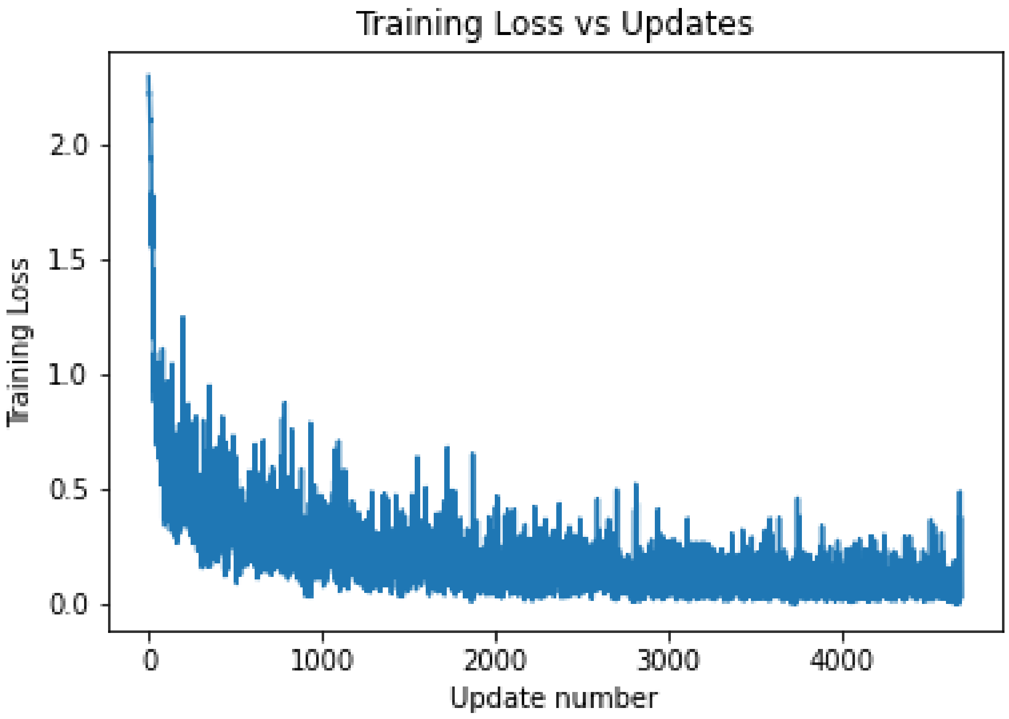


B) Plot of training loss versus epochs

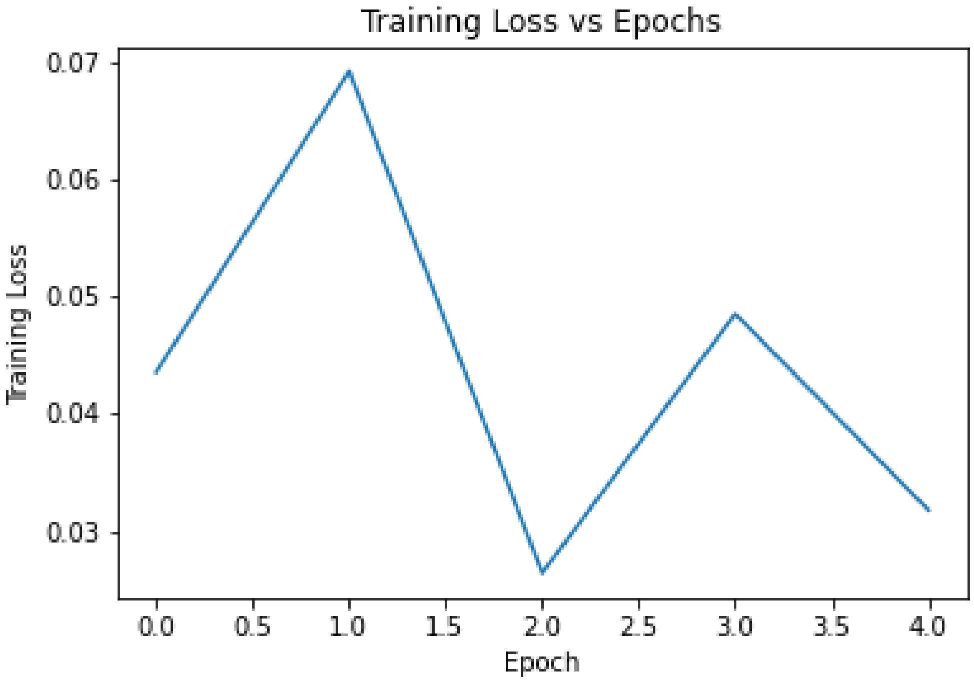


2) Learning rate = 0.02

A) Plot of training loss with each update

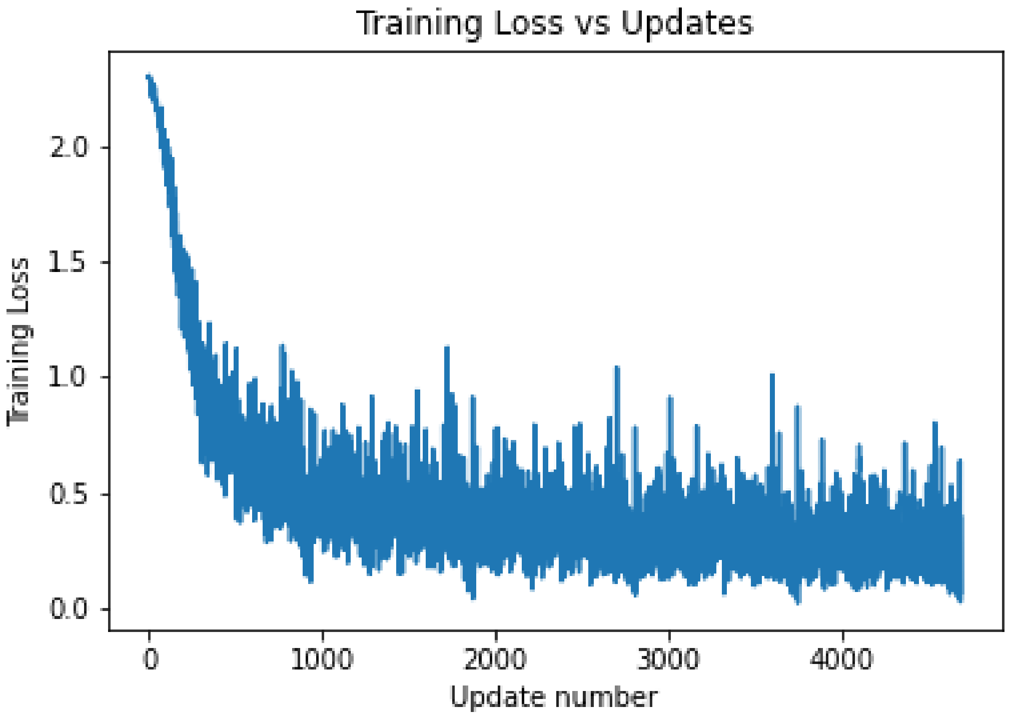


B) Plot of training loss versus epochs

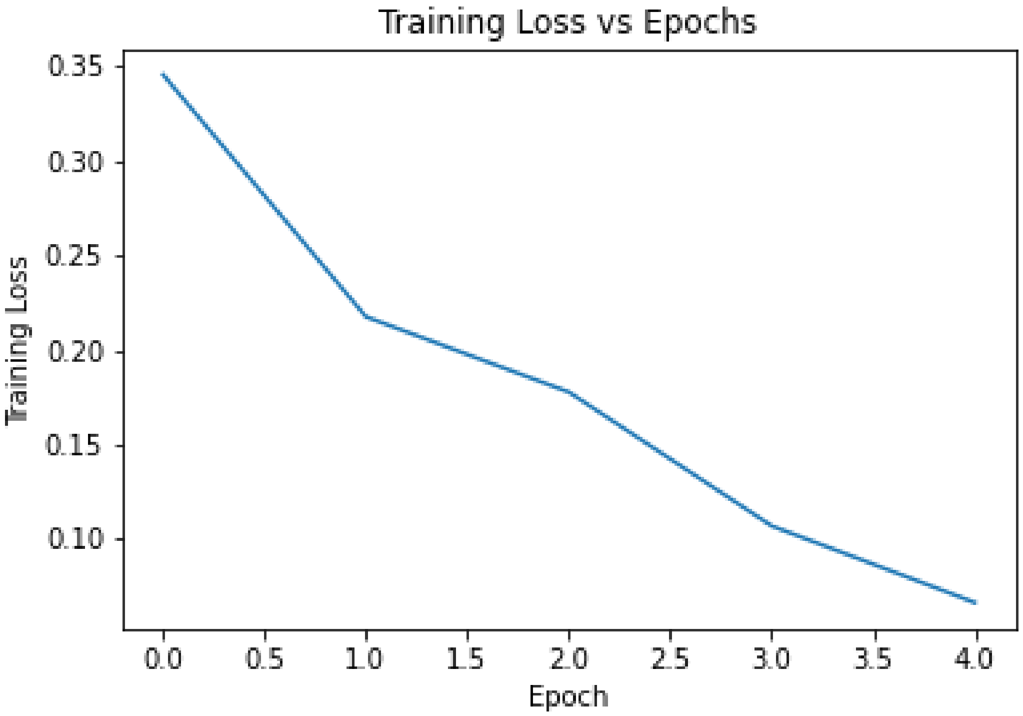


3) Learning rate = 0.002

A) Plot of training loss with each update



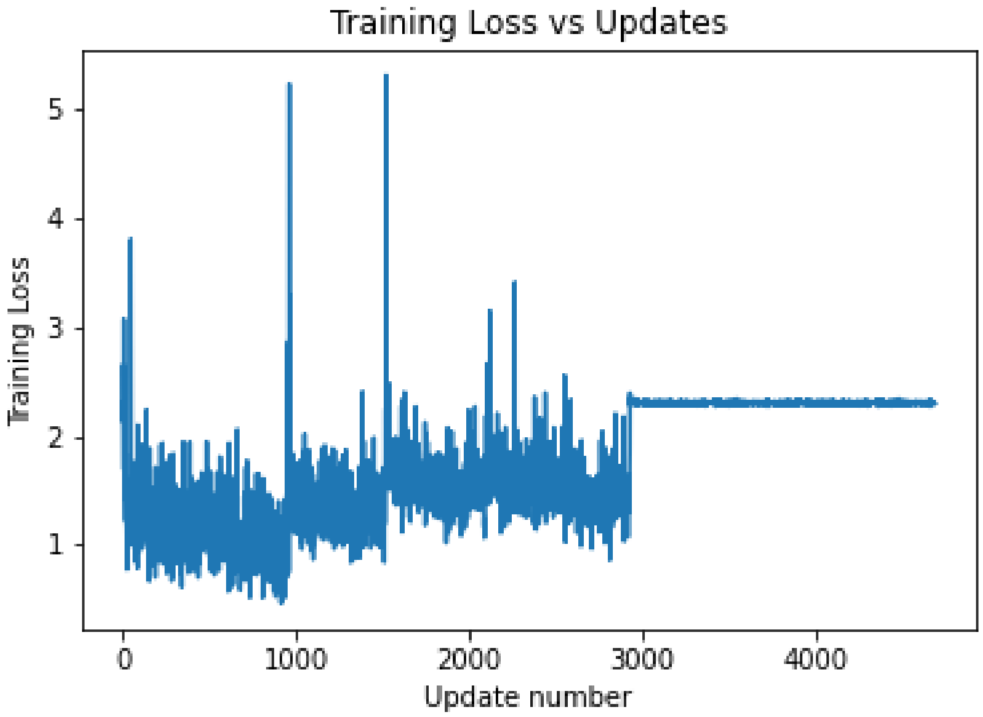
B) Plot of training loss versus epochs



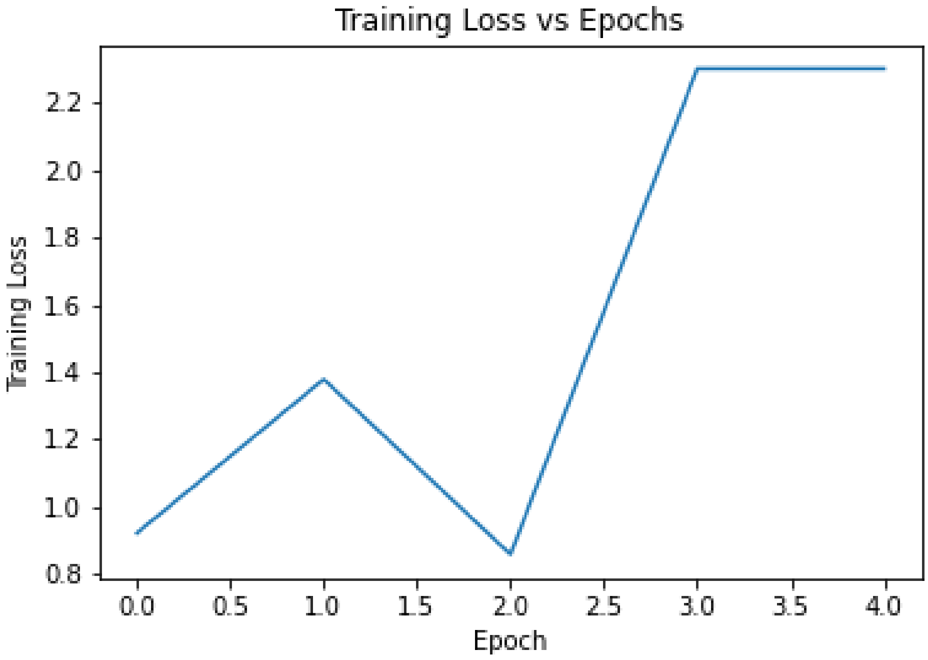
- **Momentum Gradient Descent**

  1) Learning rate = 0.2

  A) Plot of training loss with each update

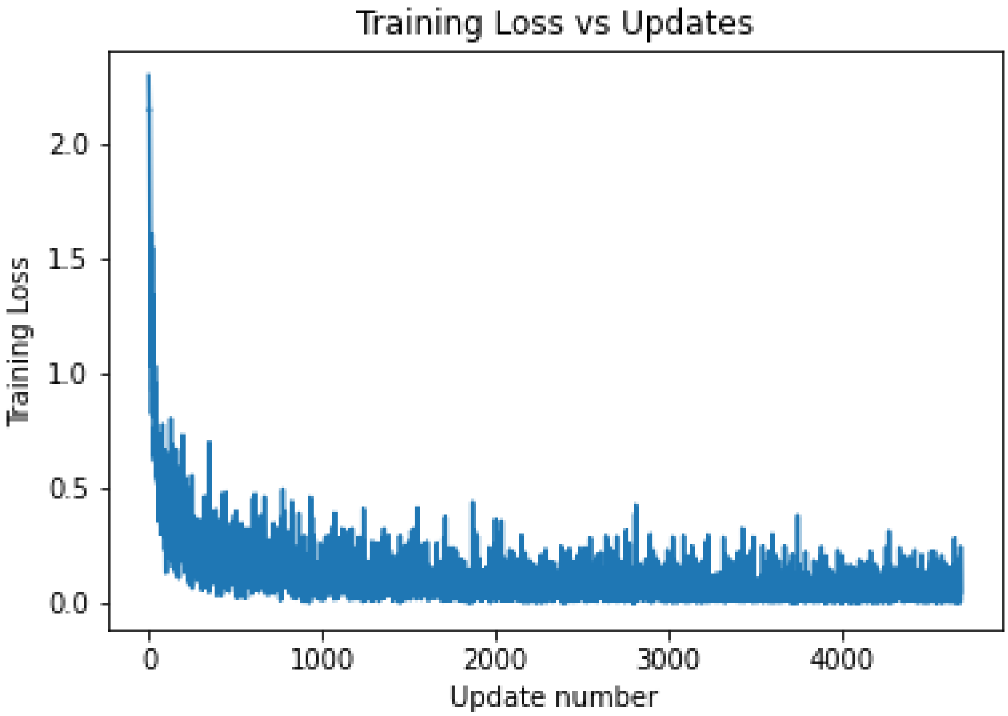


  B) Plot of training loss versus epochs

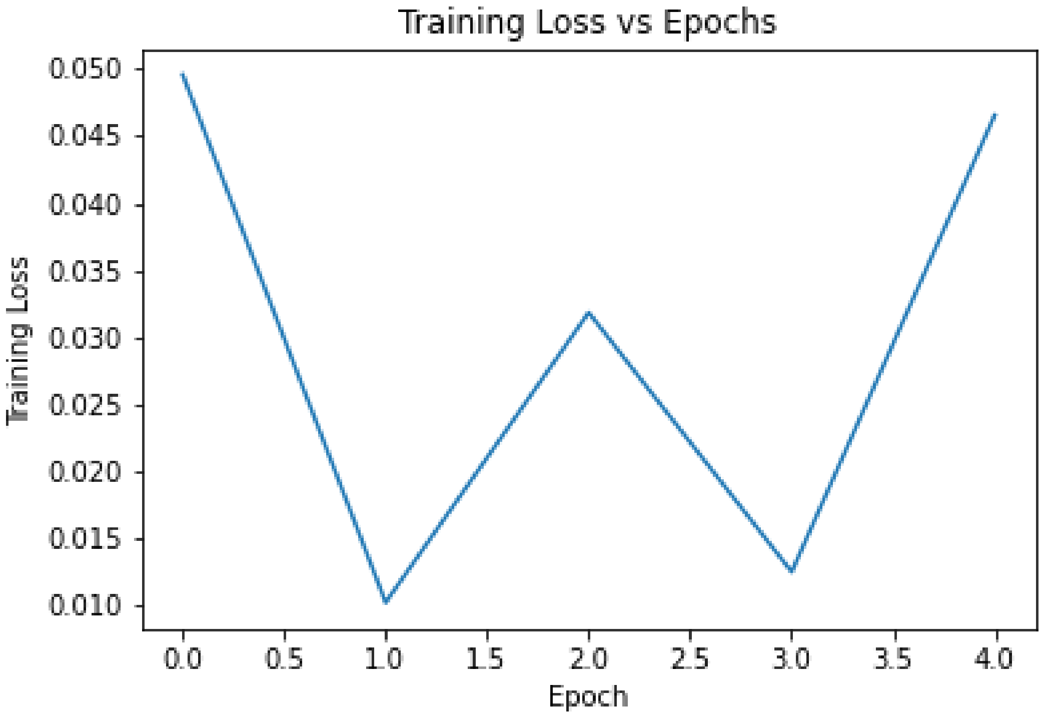


2) Learning rate = 0.02

A) Plot of training loss with each update

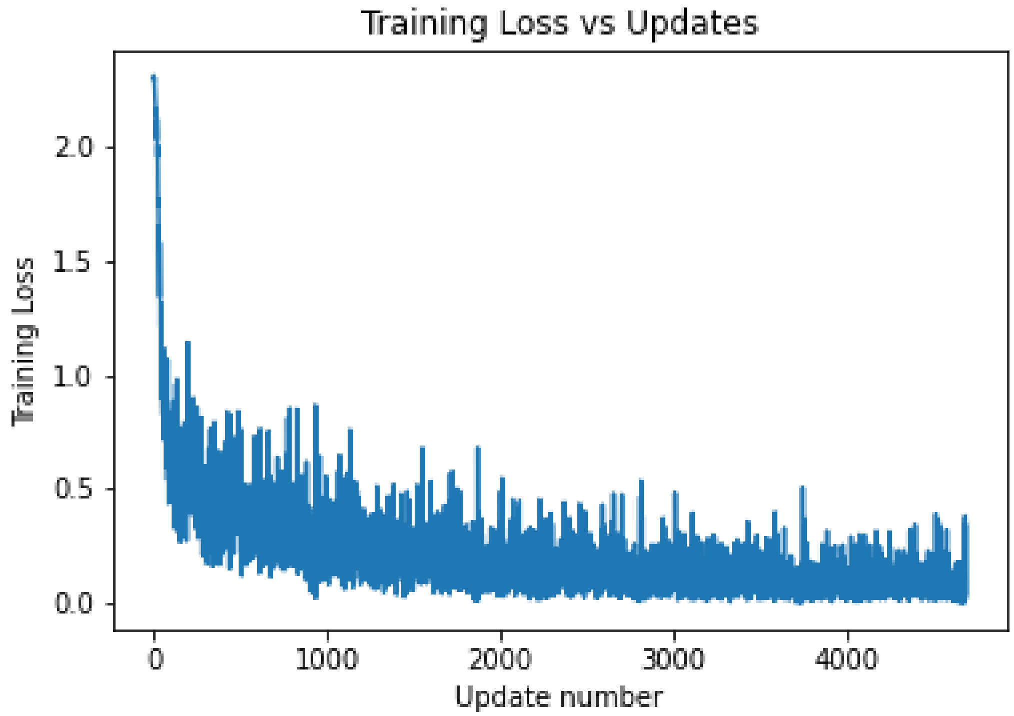


B) Plot of training loss versus epochs

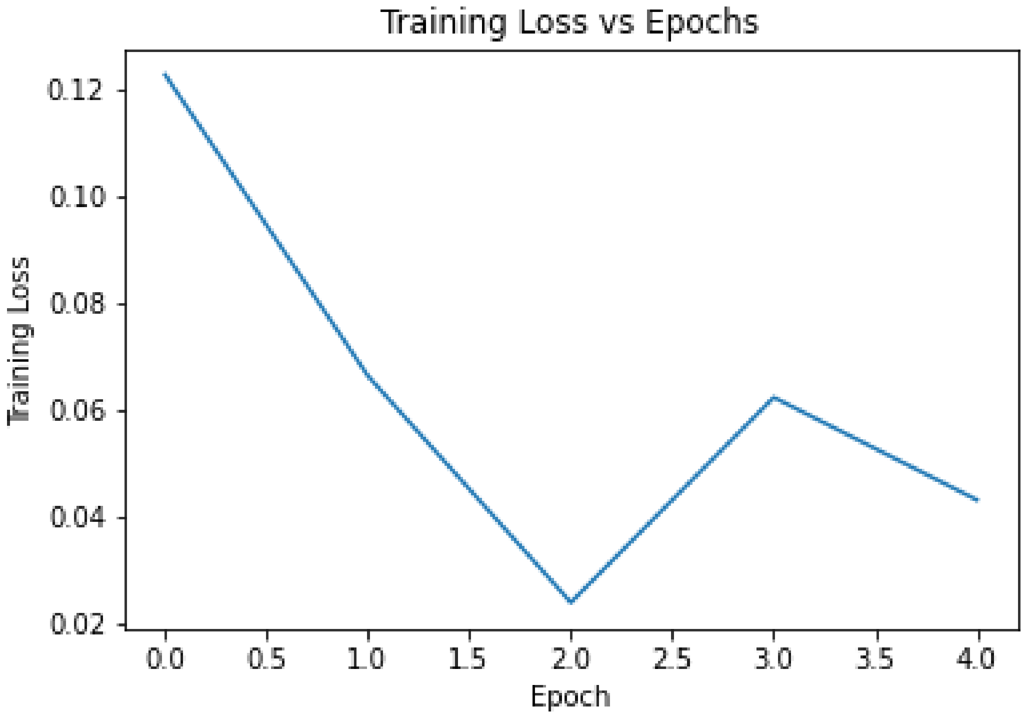


3) Learning rate = 0.002

A) Plot of training loss with each update



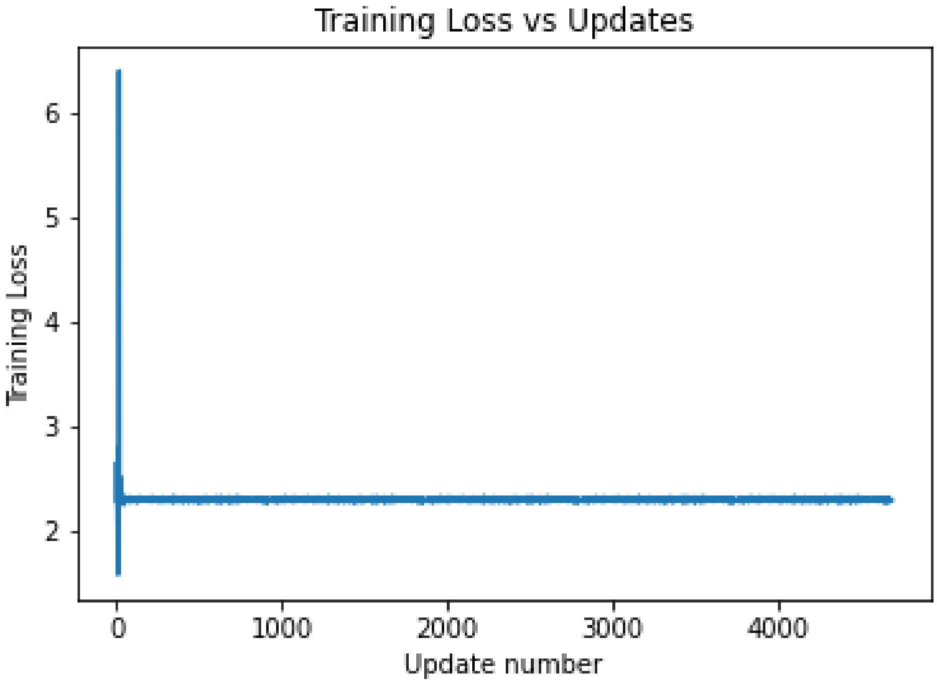
B) Plot of training loss versus epochs



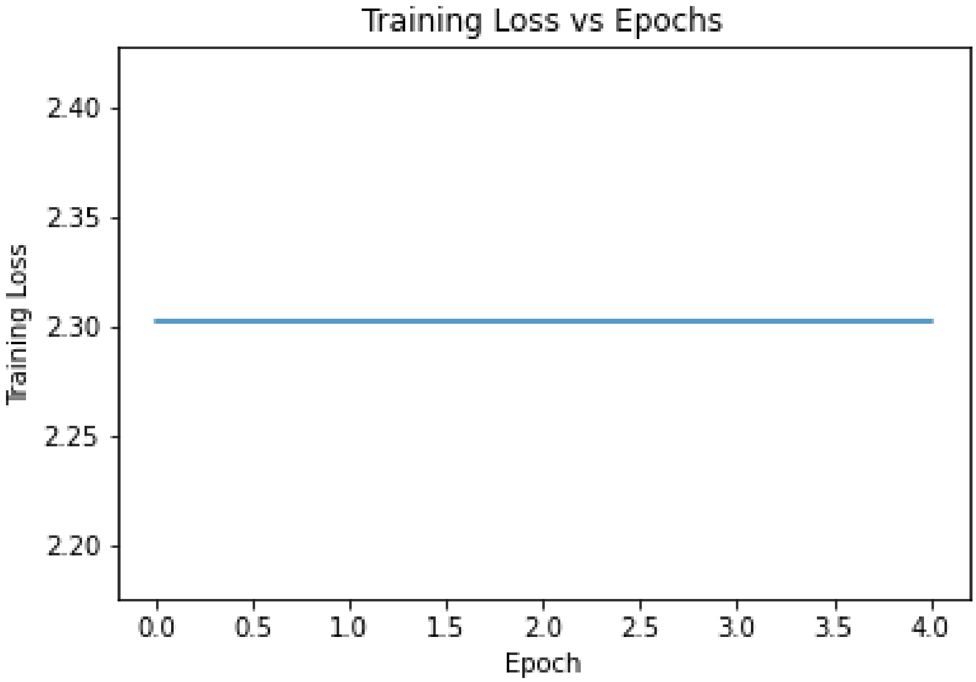
- **Nesterov Gradient Descent**

  1) Learning rate = 0.2

  A) Plot of training loss with each update

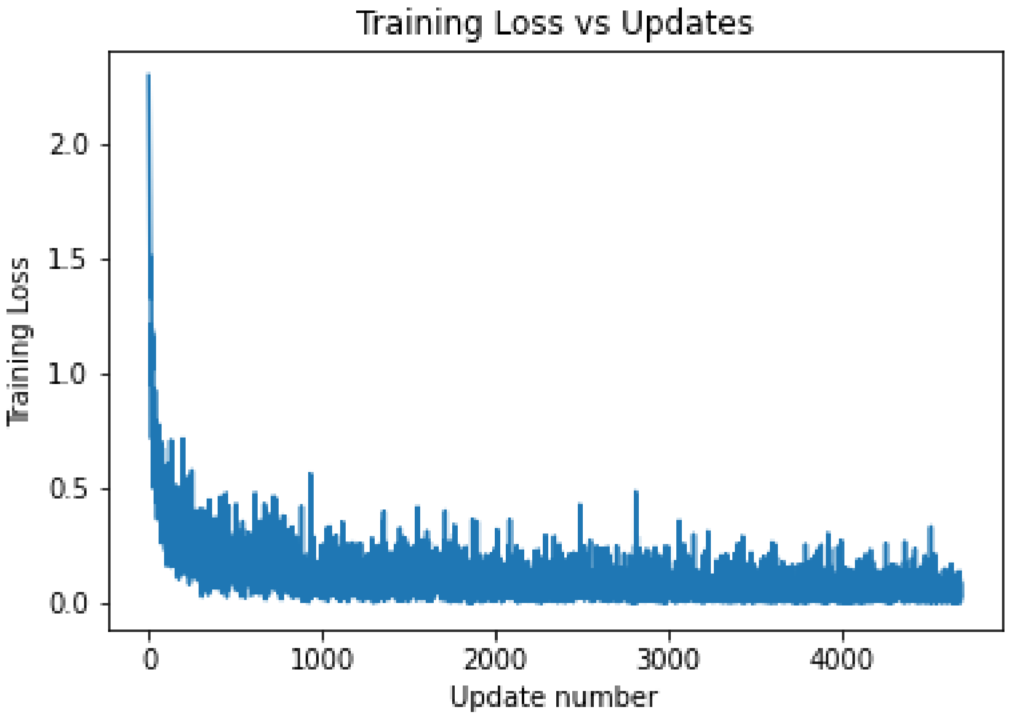


  B) Plot of training loss versus epochs

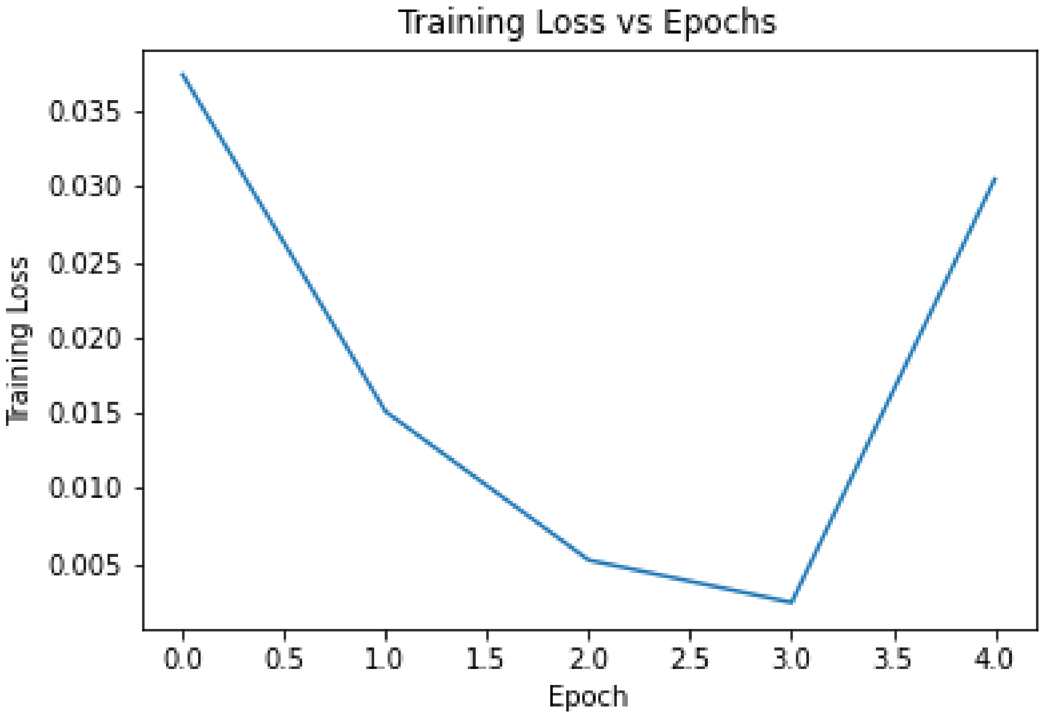


2) Learning rate = 0.02

A) Plot of training loss with each update

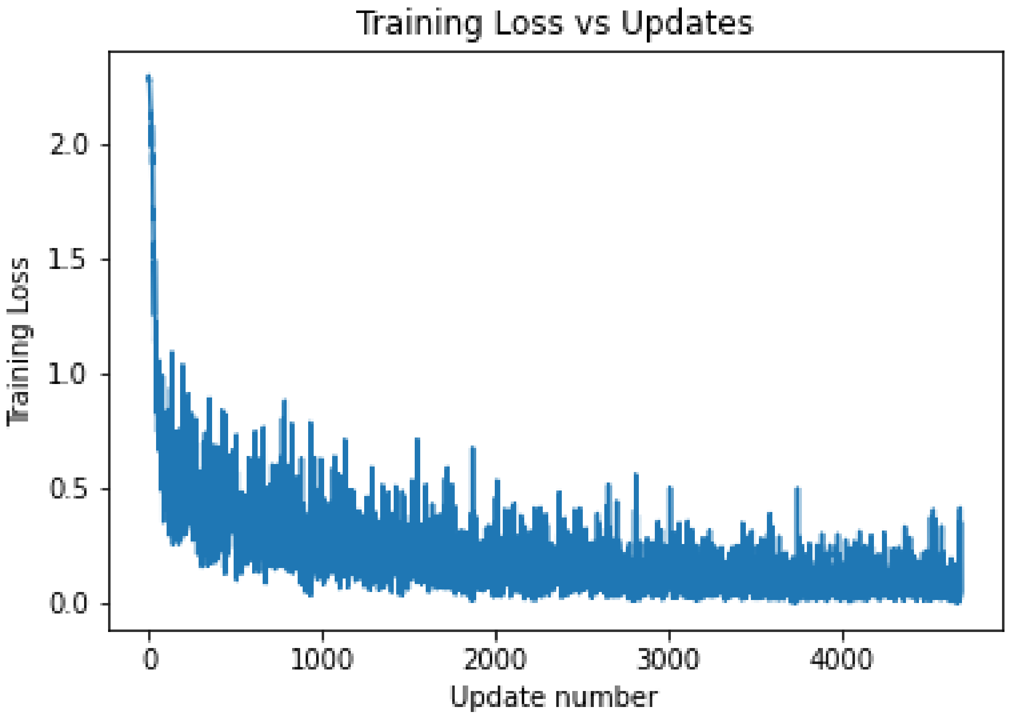


B) Plot of training loss versus epochs

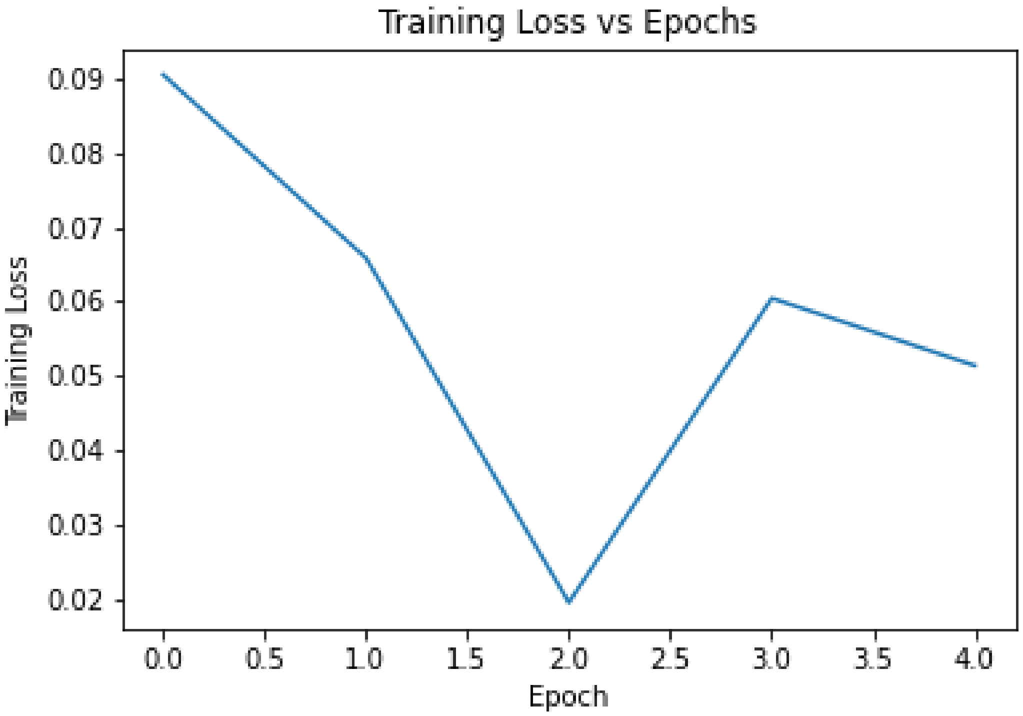


3) Learning rate = 0.002

A) Plot of training loss with each update



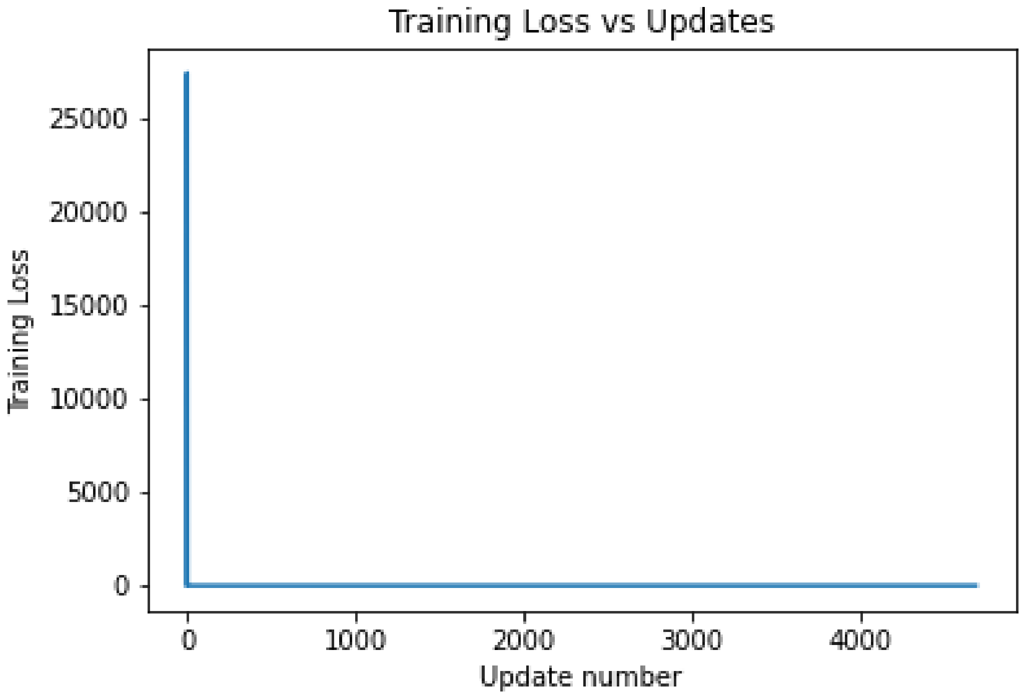
B) Plot of training loss versus epochs



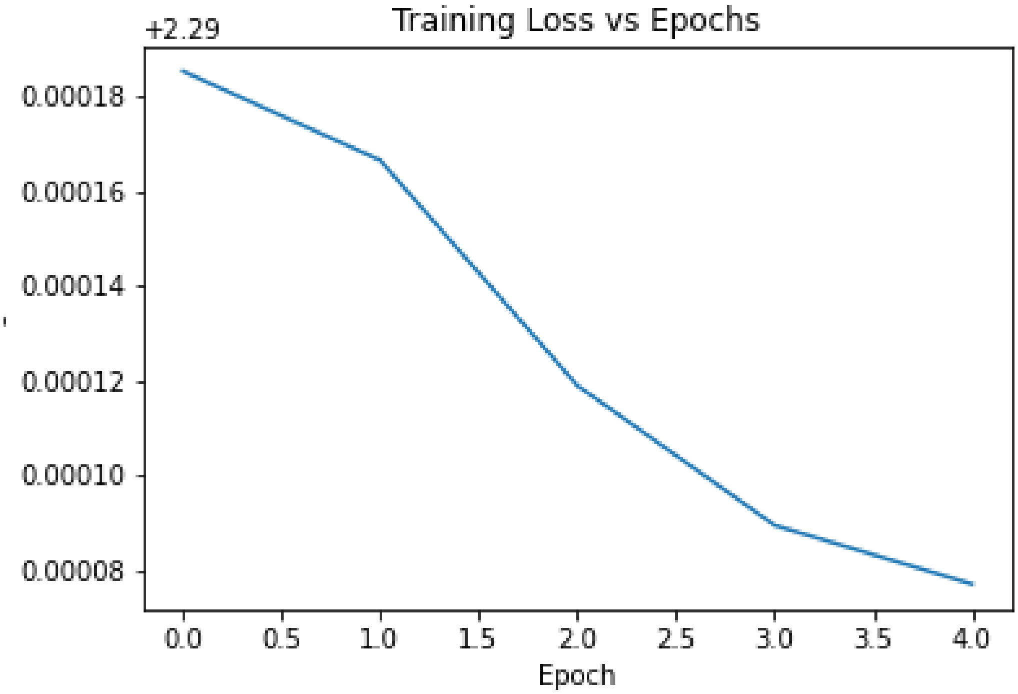
- **Adam Optimizer**

  1) Learning rate = 0.2

  A) Plot of training loss with each update

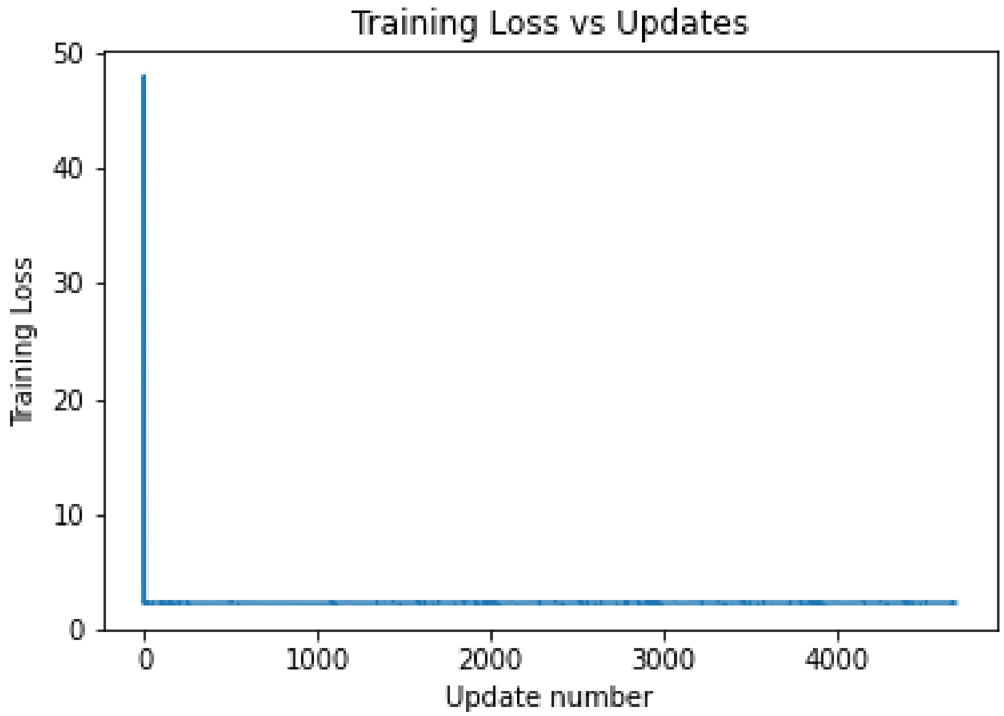


  B) Plot of training loss versus epochs

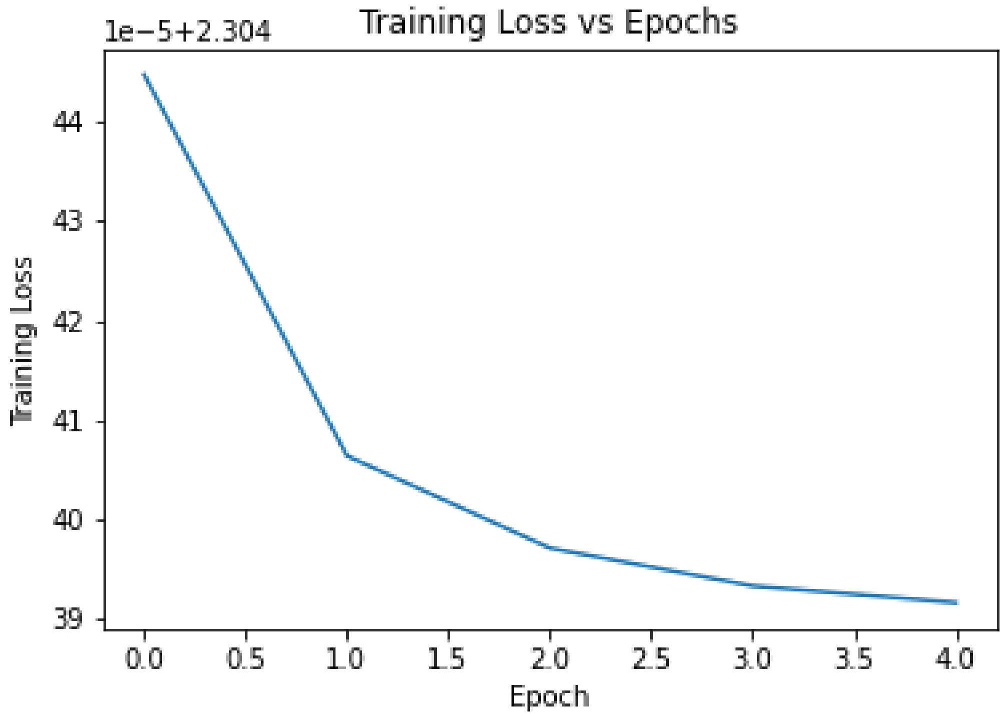


2) Learning rate = 0.02

A) Plot of training loss with each update

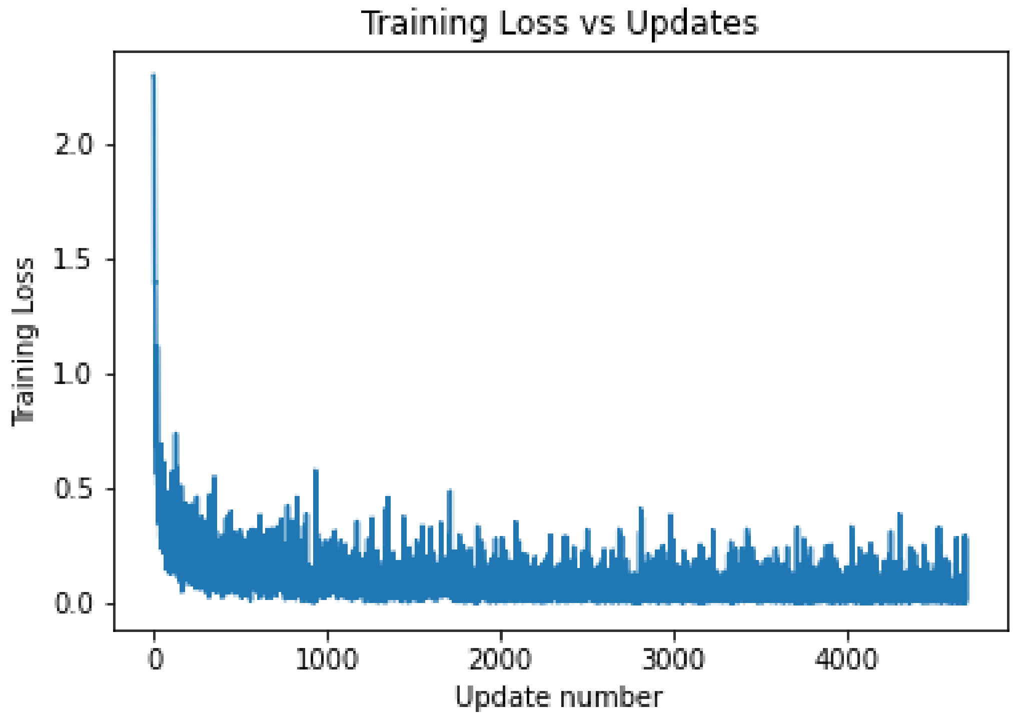


B) Plot of training loss versus epochs

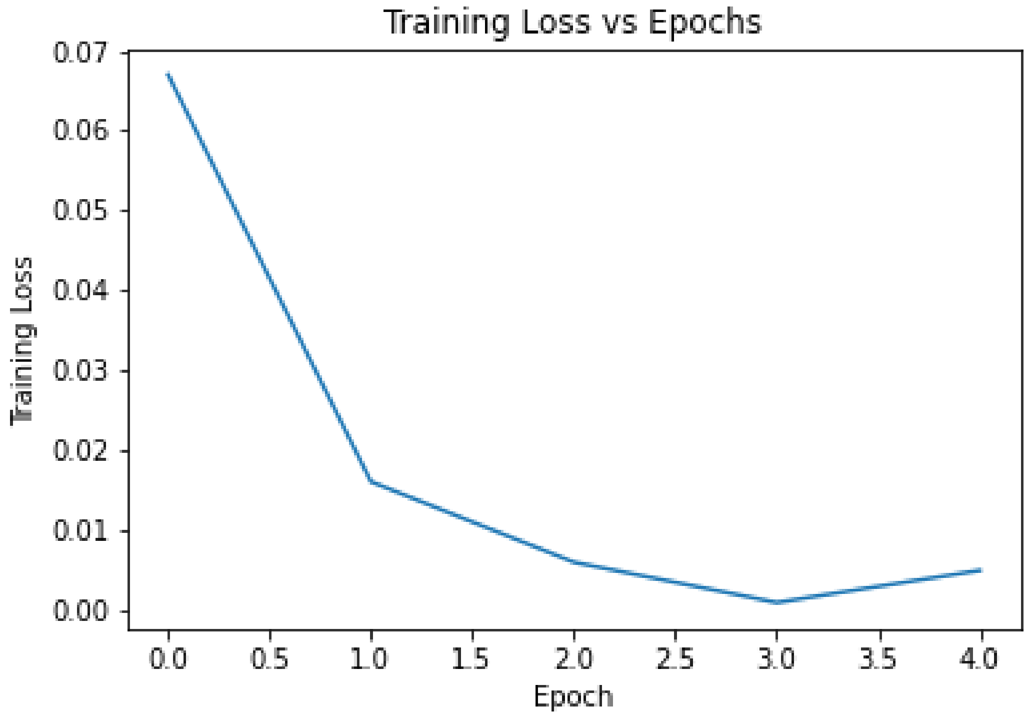


3) Learning rate = 0.002

A) Plot of training loss with each update

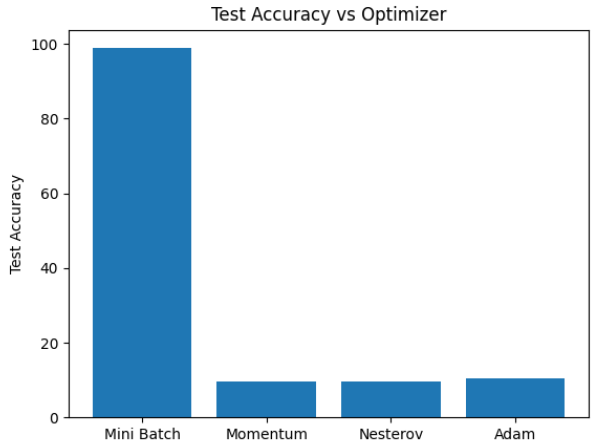


B) Plot of training loss versus epochs

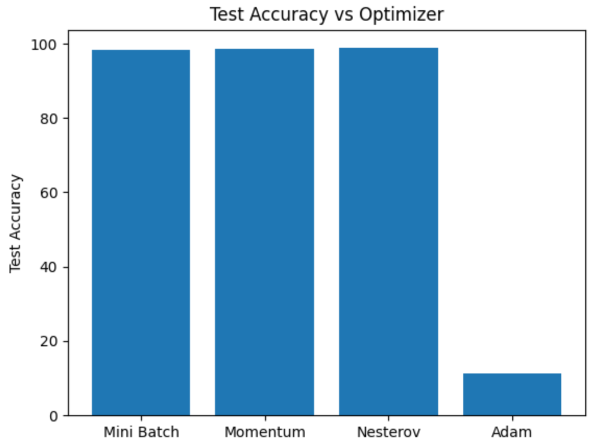


- **Plots of Accuracy vs Optimizer**

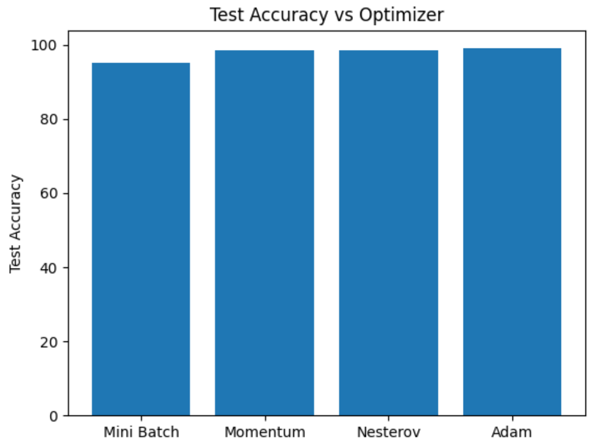
1) Learning rate = 0.2



2) Learning rate = 0.02



3) Learning rate = 0.002



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
- 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
- The algorithms diverged in some cases when the learning rate was high (0.2). For eg. Momentum Gradient Descent. This is evident form the training loss vs update curve and the poor accuracy.
- After 5 epochs and different learning rates, Adam Optimizer algorithm was found to be the most accurate (99.03%) when learning rate was set to 0.002.
- The accuracies for different algorithms for different learning rates have been provided below-

Algorithm	Accuracy (L = 0.2)	Accuracy (LR = 0.02)	Accuracy (LR = 0.002)
Mini Batch Gradient Descent	**98.9**	**98.45**	**95.14**
Momentum Gradient Descent	**9.58 (Diverged)**	**98.66**	**98.49**
Nesterov Gradient Descent	**9.58 (Diverged)**	**98.83**	**98.47**
Adam Optimizer	**10.32 (Diverged)**	**11.35 (Diverged)**	**99.03**