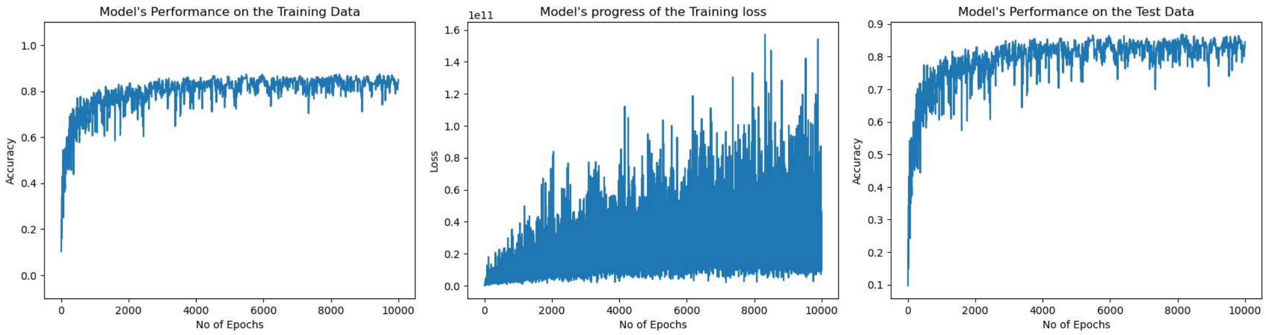
CS 228 - Linear MNIST Classifier
University of California, Riverside
Fall 2023
Hard Parikh
Email: hpari002@ucr.edu
Student ID: 862464272

Homework 1

This homework answers the problem set sequentially.

1. The role of batch size: Run your code with batch sizes B = 1, 10, 100, 1000. For each batch size,
   • determine a good choice of learning rate
   • pick ITR sufficiently large to ensure the (approximate) convergence of the training loss
   • Plot the progress of training loss (y-axis) as a function of the iteration counter t (x-axis)
   • Report how long the training takes (in seconds).
   • Plot the progress of the test accuracy (y-axis) as a function of the iteration counter t (x-axis).

   - For the batch size of 1, the following is the analysis of the best choice for the different hyper-parameters:-
   1. The best learning rate for the batch size of 1 was found to be 0.01 and the corresponding epochs were 10000. However, looking at the trend we can say that generally, a lower learning rate results in a better test accuracy for a smaller batch size, as in our case 1.
   2. For a constant learning rate and a constant batch size (1), as the number of epochs increases the test accuracy of the model increases.



(a) (b) (c)

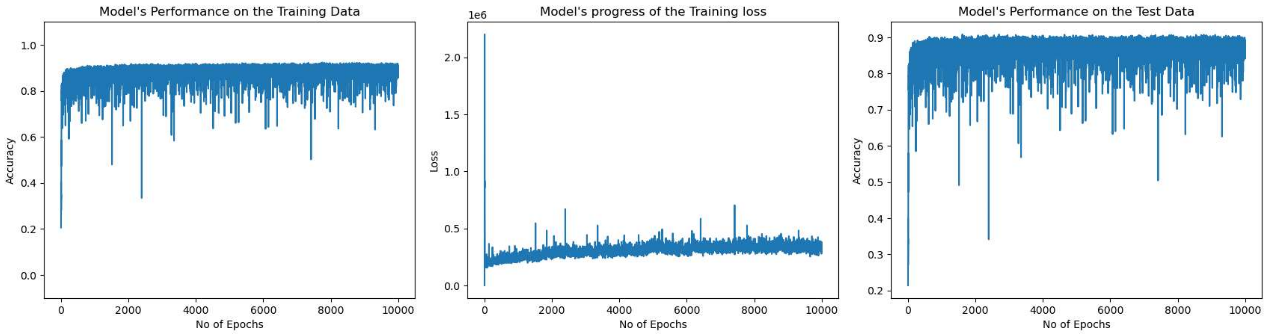
Figure 1: For Batch Size:1 (a) Training accuracy VS Epochs (b) Training loss VS Epochs (c) Test (Validation) accuracy VS Epochs

For the batch size of 10, the following is the analysis of the best choice for the different hyper-parameters:-
1. The best learning rate for the batch size of 10 was found to be 0.01 and the corresponding epochs were 10000. However, looking at the trend we can say that generally, a lower learning rate results in a better test accuracy for a smaller batch size, as in our case 10.
2. For a constant learning rate and a constant batch size (10), as the number of epochs increases the test accuracy of the model increases.

(a) (b) (c)

Figure 2: For Batch Size:10 (a) Training accuracy VS Epochs (b) Training loss VS Epochs (c) Test (Validation) accuracy VS Epochs

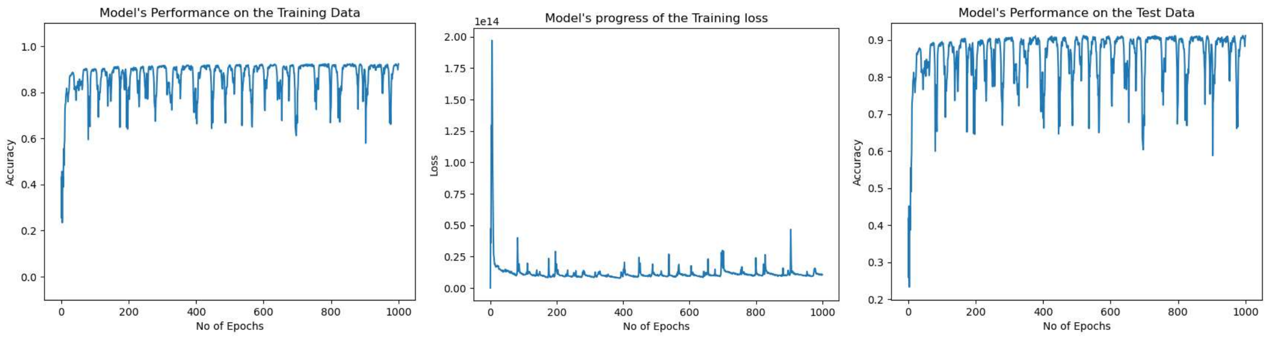
For the batch size of 100, the following is the analysis of the best choice for the different hyper-parameters:-
1. The best learning rate for the batch size of 100 was found to be 0.00001 and the corresponding epochs were 10000. However, looking at the trend we can say that generally, a lower learning rate results in a better test accuracy for a smaller batch size, as in our case 100.
2. For a constant learning rate and a constant batch size (100), as the number of epochs increases the test accuracy of the model increases.



(a) (b) (c)

Figure 3: For Batch Size:100 (a) Training accuracy VS Epochs (b) Training loss VS Epochs (c) Test (Validation) accuracy VS Epochs

For the batch size of 1000, the following is the analysis of the best choice for the different hyper-parameters:-
1. The best learning rate for the batch size of 1000 was found to be 0.01 and the corresponding epochs were 1000. Looking at the trend we can say that generally, a higher learning rate results in a better test accuracy for a larger batch size, as in our case 1000.
2. For a constant learning rate and a constant batch size (1000), as the number of epochs increases the test accuracy of the model increases. However, the number of epochs required for a better accuracy for a larger batch size is smaller than for a smaller batch size.



(a) (b) (c)

Figure 4: For Batch Size:1000 (a) Training accuracy VS Epochs (b) Training loss VS Epochs (c) Test (Validation) accuracy VS Epochs

Analysis on the training time of the model:-

- While training with the best parameters (learning rate:0.01 and epochs:10000) for the batch size 1, it took 2003.044 seconds for the training to complete.
While training with the best parameters (learning rate:0.01 and epochs:10000) for

the batch size 10, it took 2052.87 seconds for the training to complete.
While training with the best parameters (learning rate:0.00001 and epochs:10000) for the batch size 100, it took 2052.20 seconds for the training to complete.
While training with the best parameters (learning rate:0.01 and epochs:1000) for the batch size 1000, it took 217.9 seconds for the training to complete.

From the above it can be said that the training time is directly proportional to the number of epochs and inversely proportional to the batch size and magnitude of the learning rate.

2. Comment on the role of batch size.

   -As a result of the training of the model on different combinations of batch sizes, learning rates and number of epochs, we can conclude the following relationship between them:-

   Batch Size and Learning Rate: A smoother convergence path can be achieved by increasing the batch size because it gives the model more training samples with each update. You might need to increase the learning rate as batch size increases in order to offset the smaller but more accurate updates. With bigger batch sizes, a lower learning rate might lead convergence to delay.

   Batch Size and Number of Epochs: Because the model gains more information with each iteration, a bigger batch size can frequently lower the number of epochs needed to reach convergence. Training is accelerated as a result. On the other hand, because each update is noisier and the model requires more iterations to converge, lower batch sizes may require more epochs.

3. The role of training dataset size: Let us reduce the training dataset size. Instead of N = 50, 000, let us pick a subset S' of size N' from the original dataset without replacement and uniformly at random. Fix batch size to B = 100. Repeat the steps above for N' in 100, 500, 1000, 10000. Comment on the accuracy as a function of dataset size.

   - The next task was to test the role of the training dataset size. The test accuracies that we achieved after the training of the model on different dataset size(100,500,1000,10000) and a fixed batch size of 100 is as follows:-

   Test accuracy for dataset size 100:-60.75
   Test accuracy for dataset size 500:-80.63
   Test accuracy for dataset size 1000:-83.77
   Test accuracy for dataset size 10000:-87.94

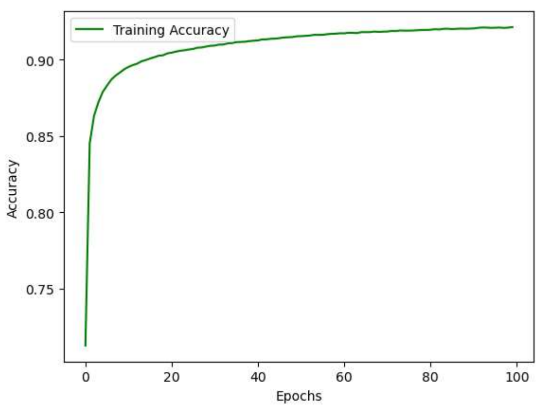
   We can clearly state that a larger dataset typically leads to better test accuracy by enabling the model to learn more robust and generalizable patterns. Conversely, smaller datasets can hinder test accuracy due to overfitting and limited generalization.

4. Simpler Life: Run the linear MNIST classifier with batchsize B = 100 over the full dataset by using PyTorch or Tensorflow. Use same learning rate and initialization W0 = 0. Verify that it is consistent with your handcoded algorithm by comparing your results (the accuracy and training loss plots).

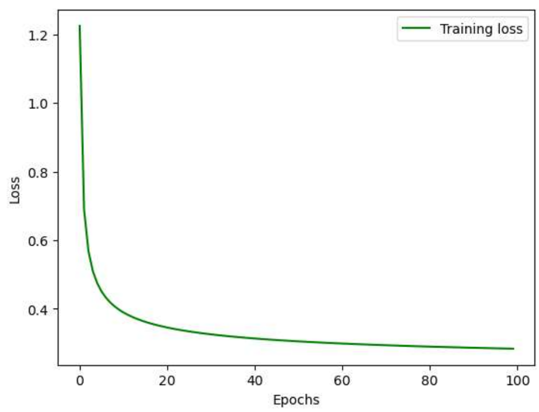
   - For the batch size of 100, number of epochs 100 and learning rate of 0.01, the model that we trained using the tensorflow, got a test accuracy of 92.10 percent whereas, the model that we built and trained without using tensorflow got a test accuracy of 86.15 percent.

   The 6 percent accuracy difference between the custom model and the TensorFlow model can be attributed to TensorFlow's optimized architecture, tuned hyperparameters, and data preprocessing.

   The training loss and accuracy graphs for the Tensorflow based model is as follows:-



(a)



(b)

Figure 5: Tensorflow Model (a) Training accuracy VS Epochs (b) Training loss VS Epochs

The training loss and accuracy graphs for the Custom based model is as follows:-

(a)

(b)

Figure 6: Custom Model (a) Training accuracy VS Epochs (b) Training loss VS Epochs

From the above discussion and visualization, it can be stated that there is a clear similarity in the training loss and accuracy graphs for the tensorflow model and the custom model.