EE 228 HW#1 - Linear MNIST Classifier

Name: Shaorong Zhang
SID: 862394226

Apr 27, 2023

1. (2 pts) Write the code for downloading and formatting the data.
Solution: see HW1 - Code.ipynb.
2. (5 pts) Write the code for minibatch SGD implementation for your linear MNIST classifier.
Solution: see HW1 - Code.ipynb.
3. (7 pts) 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)

Solution:
First choose different learning rate and iteration of given batch size, The training time and final test accuracy shown in the Table 1.

Table 1: Hyper-parameters and results of different batch size

Batch size	Learning rate	Iteration	Training time (sec)	Final test accuracy
1	10^{-9}	80000	17.27	85.12%
10	10^{-8}	8000	2.25	85.08%
100	10^{-7}	2000	0.86	85.64%
1000	10^{-7}	2000	2.83	85.83%

Training loss and test accuracy of different batch size.

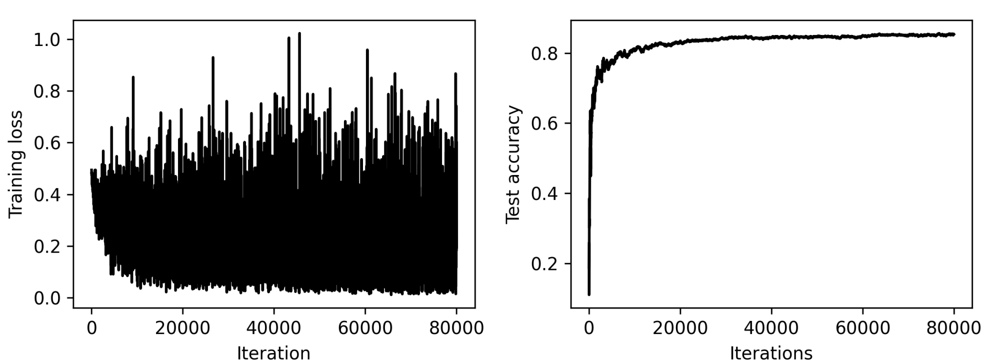

Figure 1: Training loss and test accuracy as $B = 1$

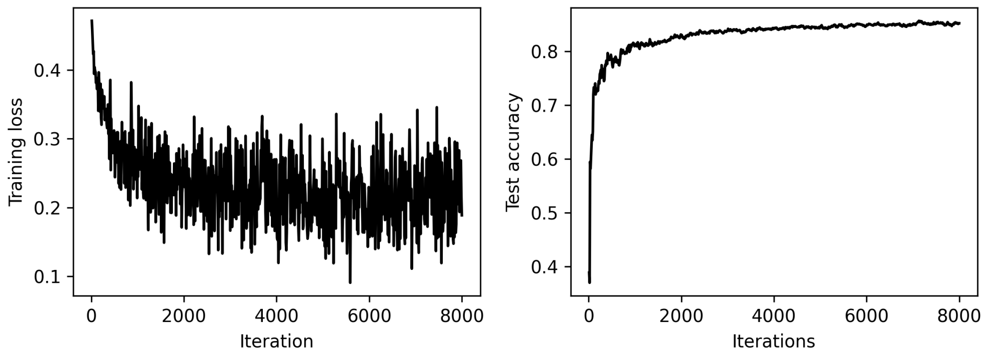

Figure 2: Training loss and test accuracy as $B = 10$

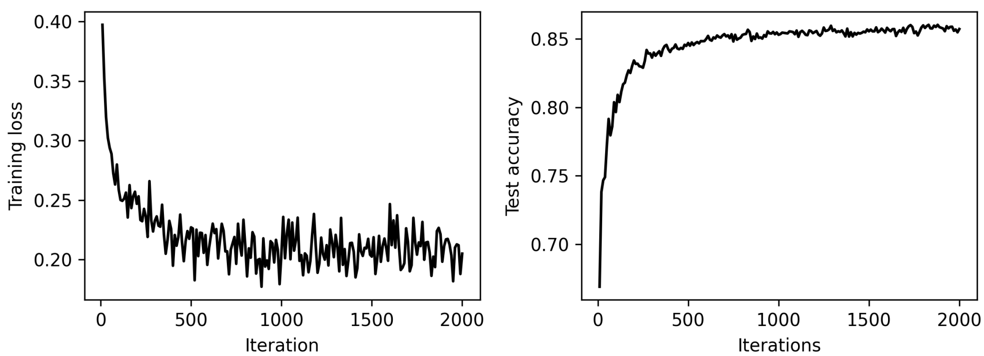

Figure 3: Training loss and test accuracy as $B = 100$

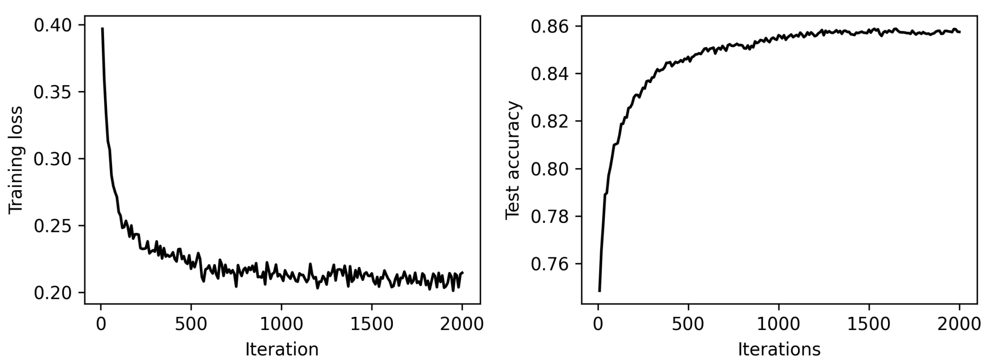

Figure 4: Training loss and test accuracy as $B = 1000$

4. Comment on the role of batch size.

Solution:

a. Choosing an appropriate batch size can accelerate the training process;

b. As batch size grows, the training time usually first decline and then rise again, so the best batch size is not the largest nor the smallest.

5. (6 pts) 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.

Solution:

The results are shown in Figure 5, as training data size grows, the final test accuracy also increase, which means that in this case, more data brings better test accuracy.

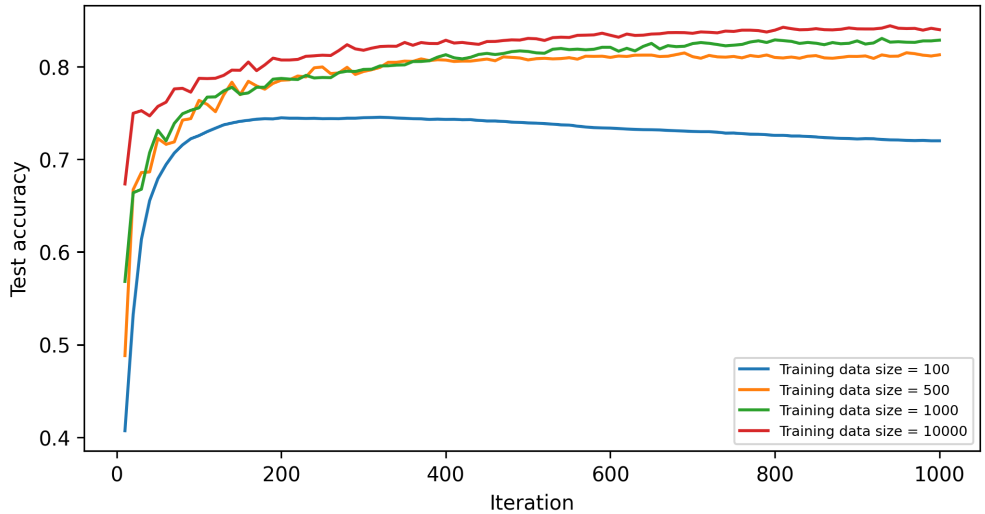

Figure 5: Test accuracy of different training data size

6. (Bonus 5 pts) 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 $W_0 = 0$. Verify that it is consistent with your handcoded algorithm by comparing your results (the accuracy and training loss plots).

Solution: The results are shown in the Figure 6, which varified the consistent between handcoded algorithm and PyTorch model.

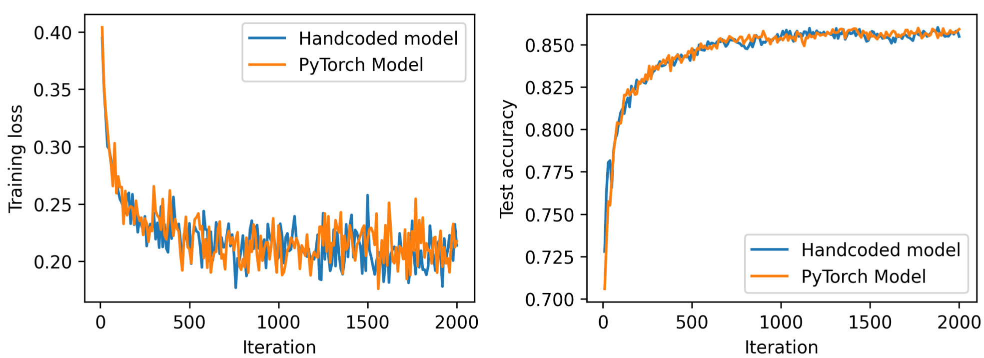

Figure 6: Training loss and test accuracy between Handcoded model and PyTorch model