EE 228 Homework 1 Report

Tianzi Luo
862251424

Question 1: Please see the code.

Here is the size of dataset
    train set & train set label:
    validation set & validation set label:
    test set & test set label:

```
train set shape (50000, 784)
train label shape (50000, 10)
validation image shape (10000, 784)
validation label shape (10000, 10)
test image shape (10000, 784)
test label shape (10000, 10)
```

Question 2: Please see the code.

Question 3: The role of batch size
Hyperparameters：
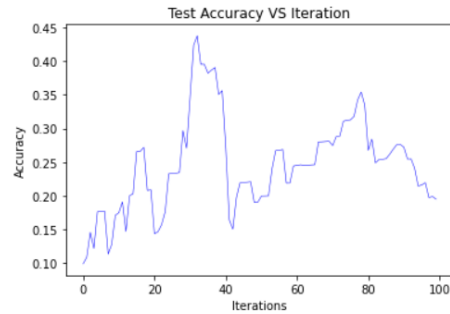   Batch size:1,10,100,1000;
    Iteration;
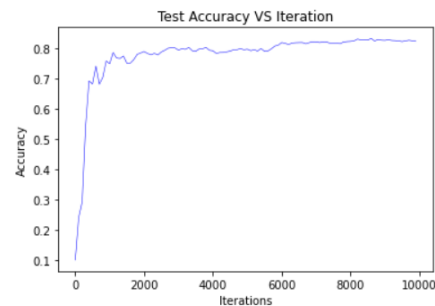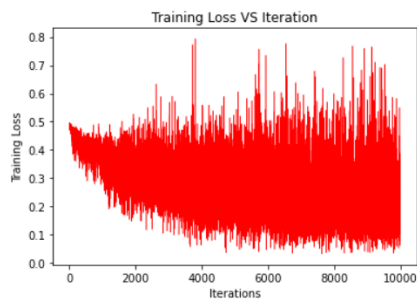   Learning rate;

B=1, determine learning rate & iterations
1.a When B=1,Iteration=100, learning rate(eta)=0.0001

--- 6.371381044387817 seconds ---
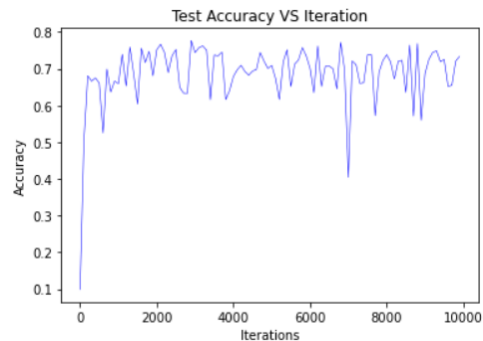

2.a When B=1,Iteration=10000, learning rate(eta)=0.0001

--- 7.043592214584351 seconds ---

The more iterations, the higher the accuracy. And the result becomes stable.

3.a When B=1,Iteration=10000, learning rate(eta)=0.01:
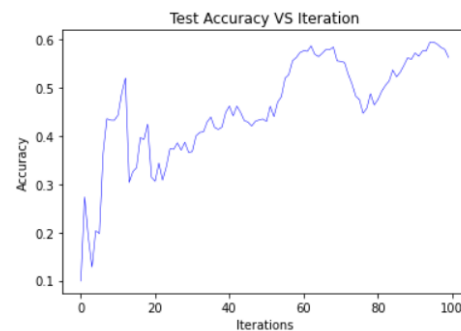
--- 6.988566637039185 seconds ---

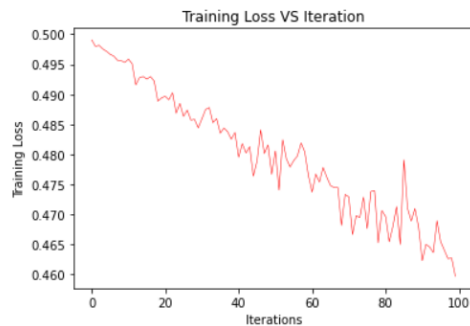So, when B=1, I think iteration=10000 and learning rate=0.01 is the best choice according to the results shown above. The accuracy is around 0.8 finally.

B=10, determine learning rate & iterations
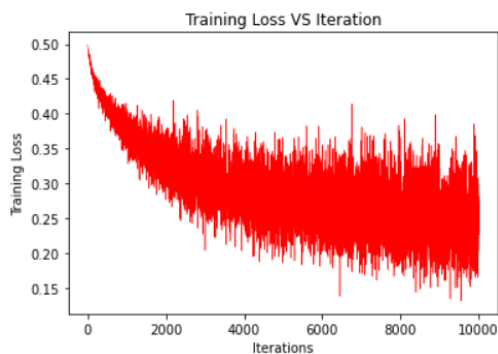1.b When B=10,Iteration=100, learning rate(eta)=0.0001

--- 6.371448755264282 seconds ---

Compared with the result of B=1, the accuracy increased and the training loss decreased significantly. So it's better than the previous example.
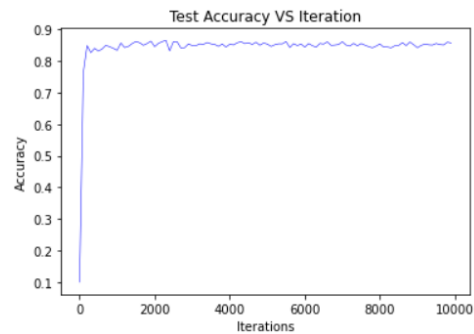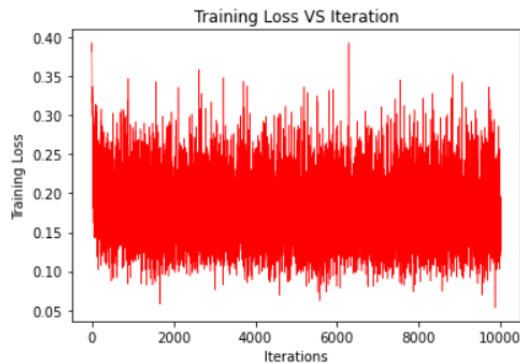
2.b When B=10,Iteration=10000, learning rate(eta)=0.0001

--- 7.489700078964233 seconds ---

3.b When B=10,Iteration=10000, learning rate(eta)=0.01

--- 7.483699083328247 seconds ---

The training loss and test accuracy will be influenced by learning rate and the number of iterations. Also the calculation time will increase as the iteration increases.
In this scenario, I think the learning rate = 0.01, iteration=10000 is the best choice.
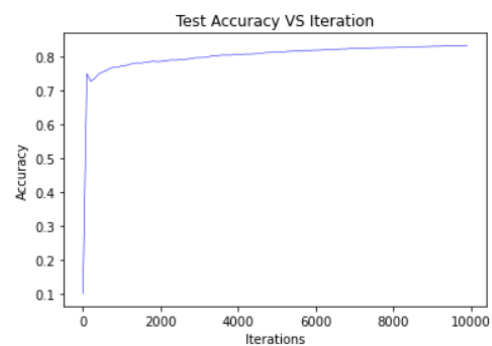

B=100, determine learning rate & iterations
1c. When B=100,Iteration=100, learning rate(eta)=0.0001

--- 6.555780649185181 seconds ---
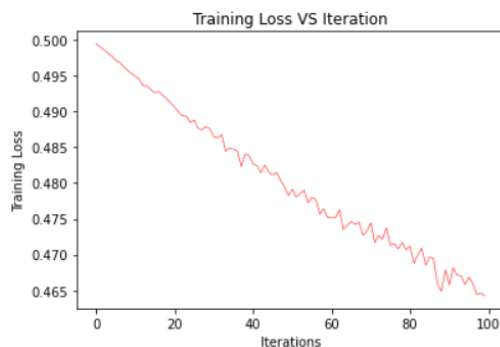
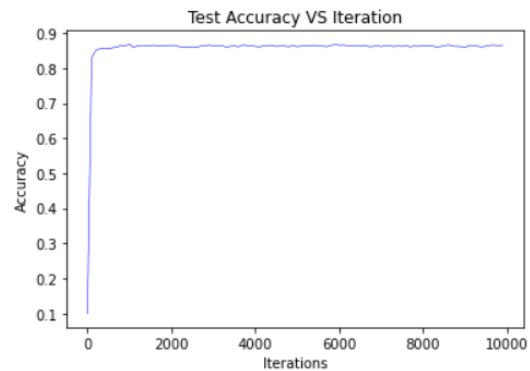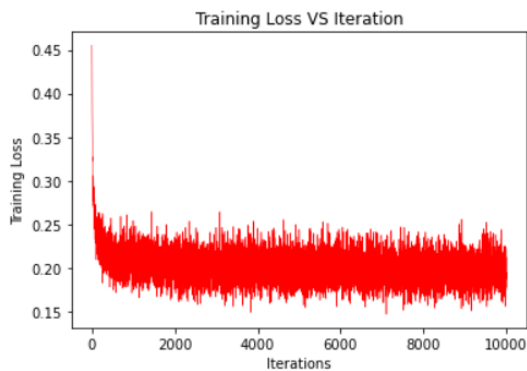2c. When B=100,Iteration=10000, learning rate(eta)=0.0001

--- 17.0068302154541 seconds ---

The calculation time increased significantly this time.

3c. When B=100,Iteration=10000, learning rate(eta)=0.01

`--- 16.914649724960327 seconds ---`

The training loss becomes stable(0.2) and accuracy is about 0.85 finally.
In this scenario, I think iteration=10000 and learning rate=0.01 is the best choice since the training loss and test accuracy are stable in the end.

B=1000, determine learning rate & iterations
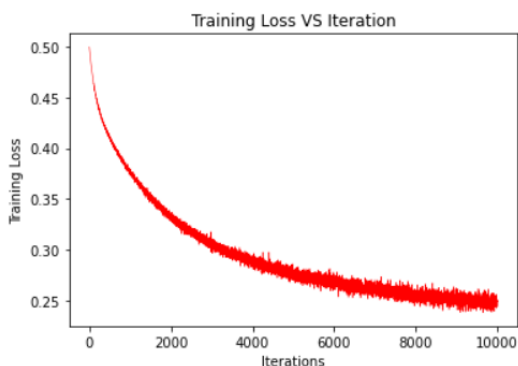### 1d. When B=1000,Iteration=100, learning rate(eta)=0.0001

`--- 7.376863718032837 seconds ---`
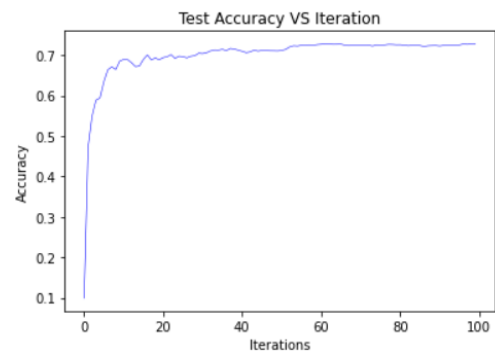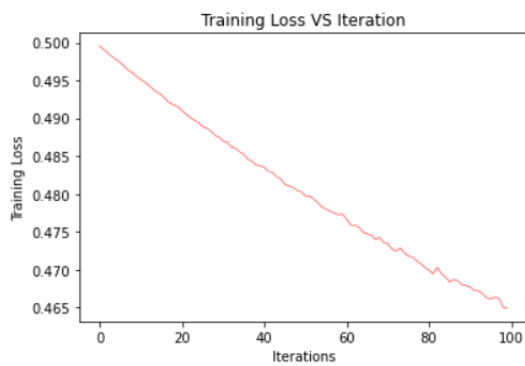
Compared with the first scenario, it is obvious that the batch size has a great impact on the training loss and test accuracy. The bigger the batch size, the better the result.
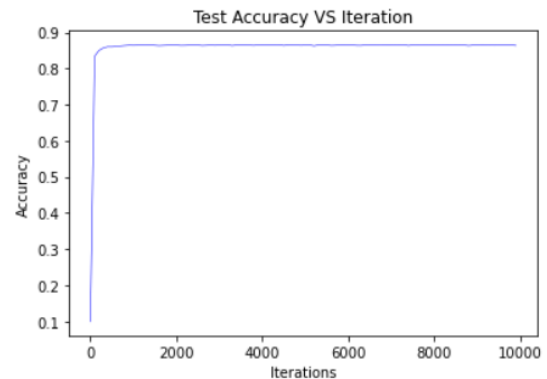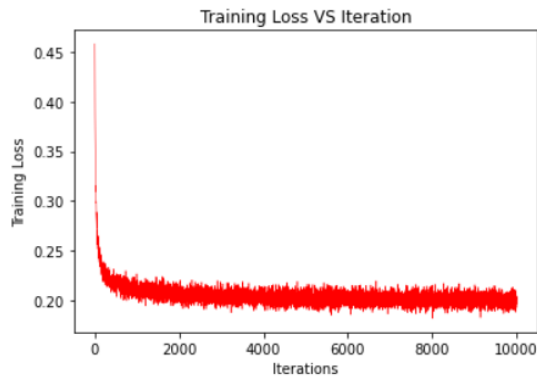
2d. When B=1000,Iteration=10000, learning rate(eta)=0.0001

`--- 106.21946954727173 seconds ---`

3d. When B=1000,Iteration=10000, learning rate(eta)=0.01

--- 106.7957124710083 seconds ---

From these experiments, we can conclude that when we increase the batch size and the number of iterations, decrease the learning rate, the training loss decreases and the accuracy increases significantly and they all become stable finally. But the cost is the calculation time increases at the same time.
I think iteration=10000 and learning rate=0.01 is the best choice for all scenarios.
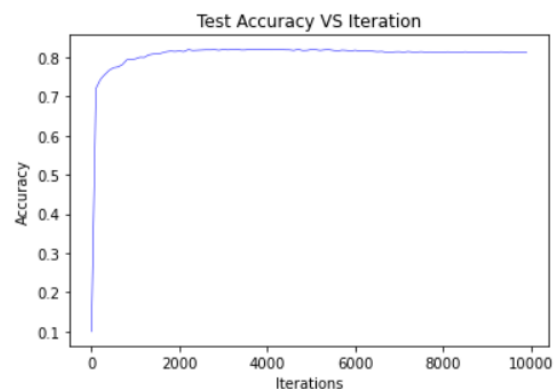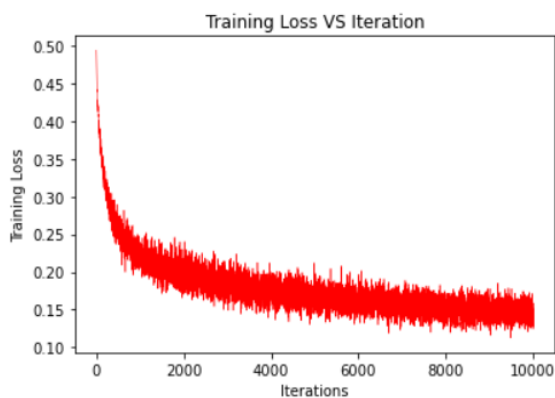
Question 5: The role of training dataset size

N'=100,B=100,iteration=10000 and the learning rate=0.01
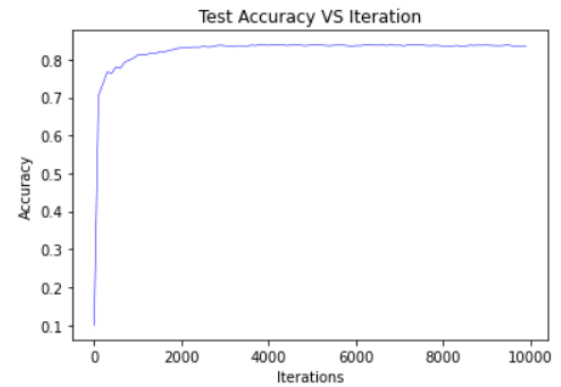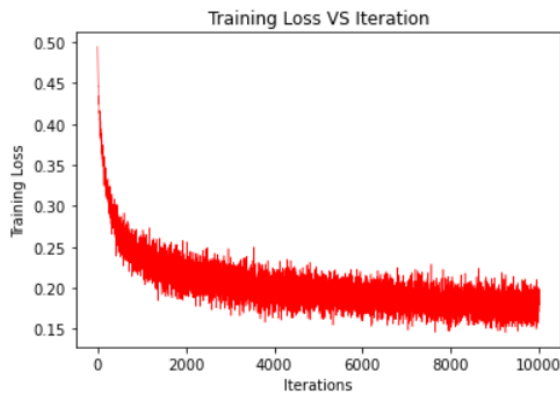
--- 16.932878017425537 seconds ---
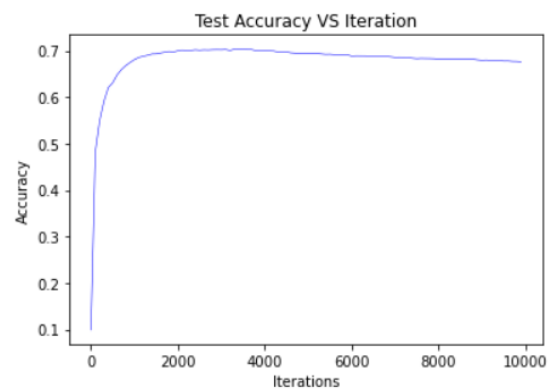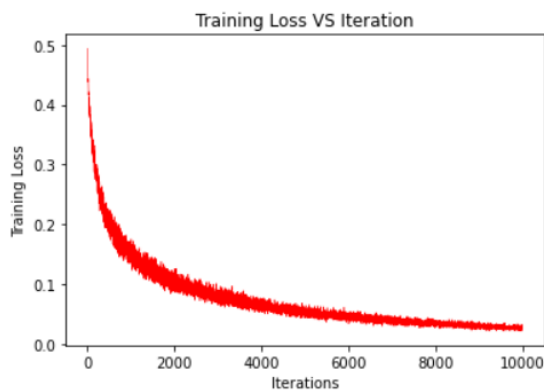
N'=500,B=100,iteration=10000 and the learning rate=0.01
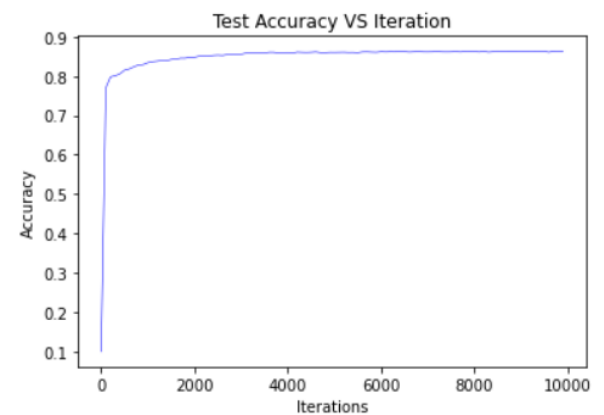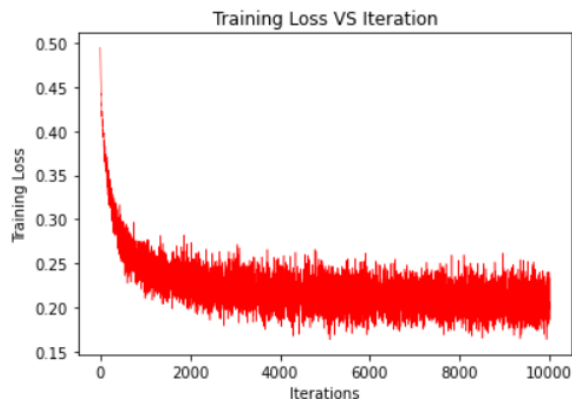
--- 17.376596689224243 seconds ---

N'=1000,B=100,iteration=10000 and the learning rate=0.01

--- 18.76704740524292 seconds ---

N'=10000,B=100,iteration=10000 and the learning rate=0.01
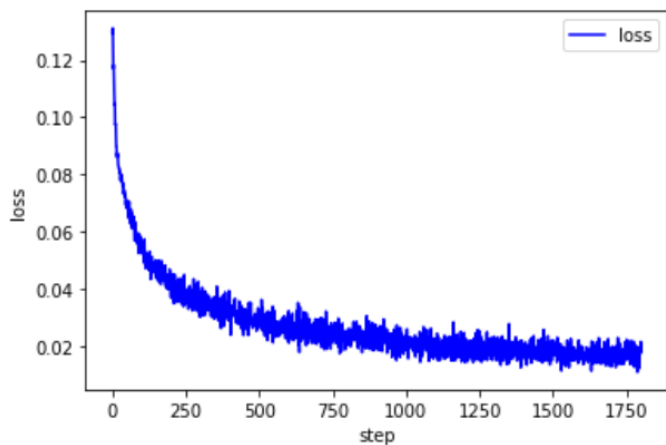
--- 16.889658451080322 seconds ---

From these experiments we can see that the training loss is stable when the size of the training data increased, but the test accuracy increased significantly.

Question 6: Simpler Life
Please see the code.
The result looks better than my hand coded algorithm. accuracy=0.94

I think the reason is I need to increase the iteration in my algorithm and there are more parameters in pytorch to participate in optimization. Like bias.