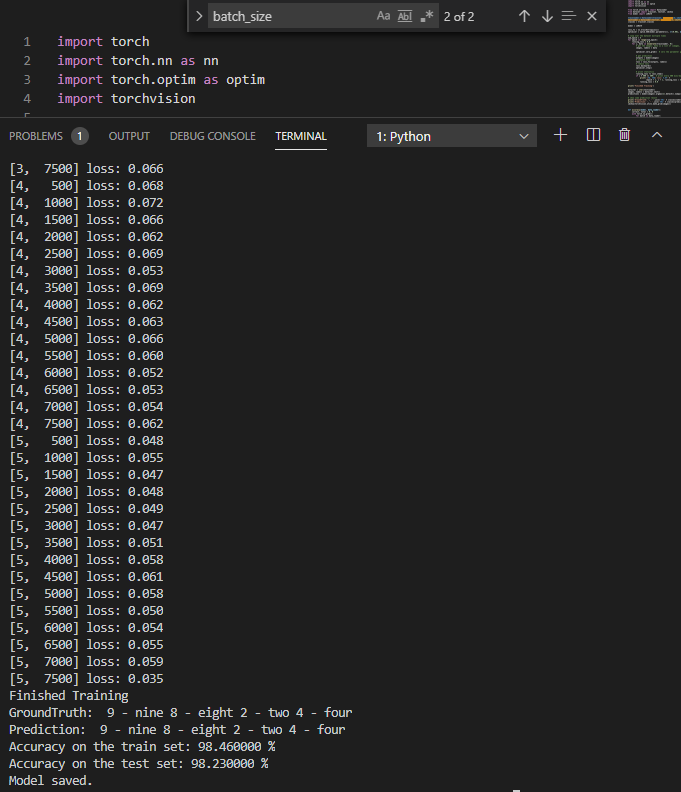
Image Classification via CNN

Chiu Ho Yin(20078151G)

1. Input image: 1x28x28
2. Conv layer:
   * kernel\_size: 5x5
   * out\_channels: 16
   * activation: ReLU
   * **input shape: [8, 1, 28, 28]**
   * **output shape: [8, 16, 24, 24]**
   * **trainable parameters: 400**
3. Max pooling:
   * kernel\_size: 2x2
   * **input shape: [8, 16, 24, 24],**
   * **output shape: [8, 16, 12, 12]**
   * **trainable parameters: 0**
4. Conv layer:
   * kernel\_size: 3x3
   * out\_channels: 32
   * activation: ReLU
   * **input shape: [8, 16, 12, 12]**
   * **output shape: [8, 32, 10, 10]**
   * **trainable parameters: 4608**
5. Max pooling:
   * kernel\_size: 2x2
   * **input shape: [8, 32, 10, 10],**
   * **output shape: [8, 32, 5, 5]**
   * **trainable parameters: 0**
6. Conv layer:
   * kernel\_size: 1x1
   * out\_channels: 8
   * activation: ReLU
   * **input shape: [8, 32, 5, 5]**
   * **output shape: [8, 8, 5, 5]**
   * **trainable parameters: 256**
7. FC layer:
   * out\_features: 64
   * activation: ReLU
   * **input shape: [8, 200],**
   * **output shape: [8, 64]**
   * **trainable parameters: 12800**
8. FC layer:
   * out\_features: ?? (to be inferred by you)
   * activation: None
   * **input shape: [8, 64],**
   * **output shape: [8, 10]**
   * **trainable parameters: 640**



In this assignment, I downloaded MNIST dataset first, and train this model with 8 batch size and with 5 epochs, the loss has reduced from 2.303 to 0.035 during training,

The final accuracy of the model on the train set is 98.46% and the accuracy on the test set is 98.23%.

The trained model is saved as “model.pth”.