Assignment 1

r12922054 邱信瑋

Experiment Setting

I. Hardware Specification

- CPU : Intel(R) Core(TM) i5-8500 CPU 3.00GHz 3.00 GHz
- GPU : NVIDIA GeForce GTX 1060 6GB

II. Package Version

- python 3.8.8
- torch 1.7.0+cu101
- torchvision 0.8.1+cu101
- numpy 1.23.5
- tqdm 4.64.1
- matplotlib 3.6.2

III. Testing Images

Testing images as the one shown in Figure 1 drawn by matplotlib.

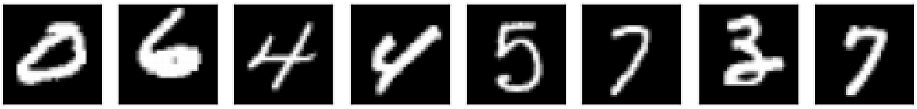

Figure 1: testing images.

IV. All the experiment parameters and details in q2

In this paragraph, I will describe the details of this implementation, as follows.

1. How to convert testing data to images?

First, reading data from *json* file and the data format is list. I converted the data format from *list* to *numpy.* The reason for doing this is to facilitate subsequent feeding of the model. Second, I used the *matplotlib* package to convert the information in *numpy arrays* into images. It is worth noting here that I used the *subplot* function in the *matplotlib* package and removed the display of the x-axis and y-axis.

2. What are my hyperparameters for model training?

(a) batch_size : 64 (b) epoch : 30 (c) loss function : CrossEntropyLoss (d) optimizer : Adam(learning rate : 0.0003, weight_decay : 1e-5)

3. What have I tried and used for model training?

(a) Data augmentation. (b) Use function *clip_grad_norm* to avoid gradient explosion. (C) Use *torch.save* to save the best model weights from training and use it to evaluate the testing images. (d) Use the *tqdm* package to view the status of training.

4. Some processing needs to be done on the testing data.

(a) Because the value of the training data is between 0-1, it is necessary to normalize the value of the testing data to 0-1.

5. How to concatenate prediction results?

First, store the predicted results of each test image into a *list*, and then use the function *concatenate* in the *numpy* package to flatten the *numpy arrays* stored in the *list.*