

# 基于 Pytorch 的 CNN 实现

171860659 吴紫航

## 实现步骤

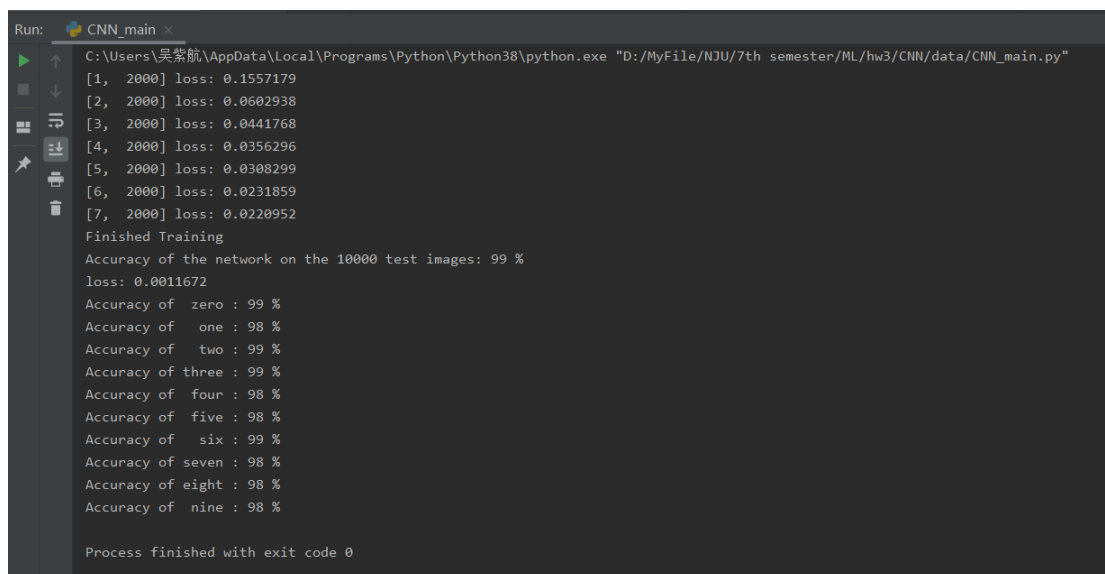
- 1.加载并标准化 Mnist 数据集（60000 训练集和 10000 测试集）
- 2.定义卷积神经网络模型
- 3.定义损失函数
- 4.用训练集数据训练网络参数
- 5.用测试集数据测试训练成果
- 6.人工调整超参数

## 测试性能

对应第 3 题第 1 问

运行代码 CNN\_main.py 会打印测试性能

即：training loss、test accuracy、validation loss、分类别 accuracy

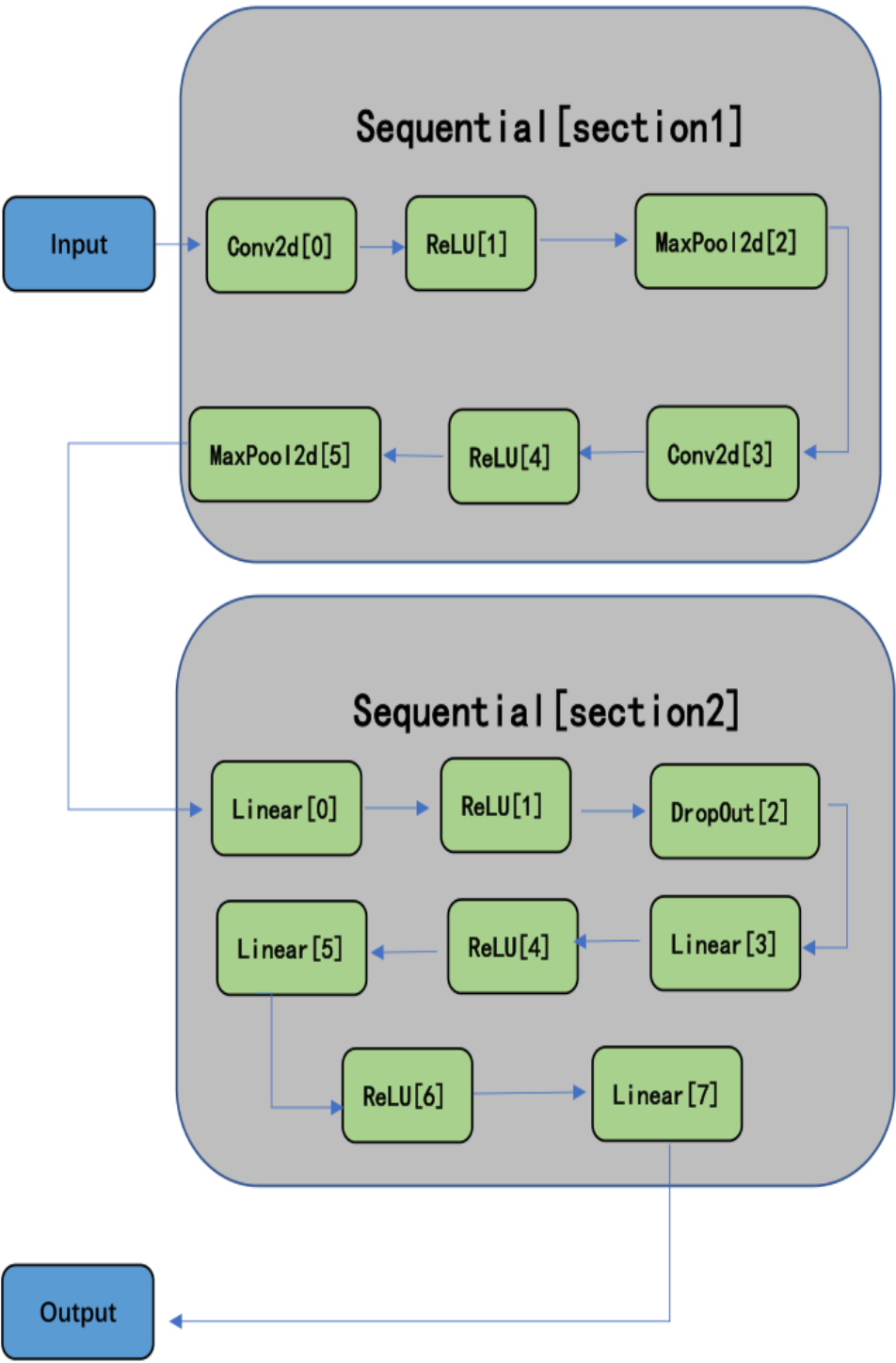


```
Run: CNN_main x
C:\Users\吴紫航\AppData\Local\Programs\Python\Python38\python.exe "D:/MyFile/NJU/7th semester/ML/hw3/CNN/data/CNN_main.py"
[1, 2000] loss: 0.1557179
[2, 2000] loss: 0.0602938
[3, 2000] loss: 0.0441768
[4, 2000] loss: 0.0356296
[5, 2000] loss: 0.0308299
[6, 2000] loss: 0.0231859
[7, 2000] loss: 0.0220952
Finished Training
Accuracy of the network on the 10000 test images: 99 %
loss: 0.0011672
Accuracy of zero : 99 %
Accuracy of one : 98 %
Accuracy of two : 99 %
Accuracy of three : 99 %
Accuracy of four : 98 %
Accuracy of five : 98 %
Accuracy of six : 99 %
Accuracy of seven : 98 %
Accuracy of eight : 98 %
Accuracy of nine : 98 %

Process finished with exit code 0
```

# 网络结构

对应第 3 题的第 2 问，本题的神经网络结构图如下



## 单个输入图片的数据流变换过程为：

### (1) Section1-Conv2d[0]

输入图片( $28 \times 28 \times 1$ ); 卷积核( $3 \times 3 \times 64$ ); padding=1; stride=1

输出( $28 \times 28 \times 64$ )

### (2) Section1-ReLU[1]

输入( $28 \times 28 \times 64$ )

输出( $28 \times 28 \times 64$ )

### (3) Section1-MaxPool2d[2]

输入( $28 \times 28 \times 64$ )

输出( $14 \times 14 \times 64$ )

### (4) Section1-Conv2d[3]

输入图片( $14 \times 14 \times 64$ ); 卷积核( $3 \times 3 \times 128$ ); padding=1; stride=1

输出( $14 \times 14 \times 128$ )

### (5) Section1-ReLU[4]

输入( $14 \times 14 \times 128$ )

输出( $14 \times 14 \times 128$ )

### (6) Section1-MaxPool2d[5]

输入( $14 \times 14 \times 128$ )

输出( $7 \times 7 \times 128$ )

### (7) Section2-Linear[0]

输入( $7 \times 7 \times 128$ )

输出(1024)

**(8)** Section2-ReLU[1]

输入(1024)

输出(1024)

**(9)** Section2-Dropout[2]

输入(1024)

输出(1024)

**(10)** Section2-Linear[3]

输入(1024)

输出(120)

**(11)** Section2-ReLU[4]

输入(120)

输出(120)

**(12)** Section2-Linear[5]

输入(120)

输出(84)

**(13)** Section2-ReLU[6]

输入(84)

输出(84)

**(14)** Section2-Linear[7]

输入(84)

输出(10)

**即最后得到 10 个类的预测概率**

## 优化方法和超参数确定

对应于第 3 题第 3 问。

采用了两种优化方法，超参数确定如下：

(1) Adam:

训练轮次 epoch=7

每批次数目 Batch size=30

总准确率：99%

(2) SGD:

训练轮次 epoch=6

每批次数目 Batch size=30

学习速率 lr=0.0001

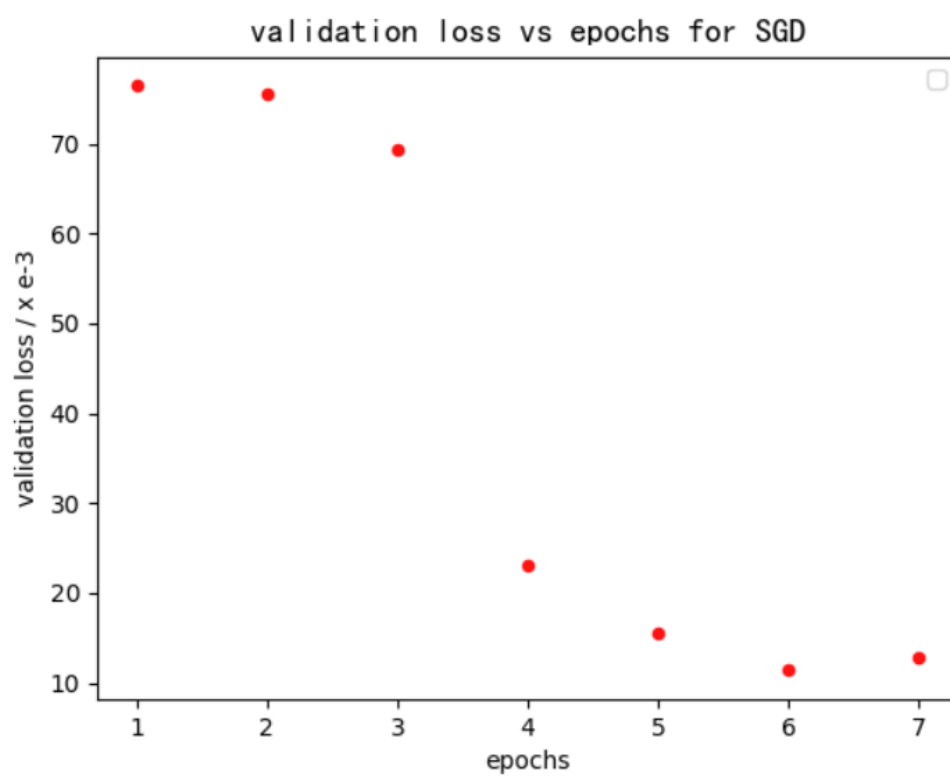
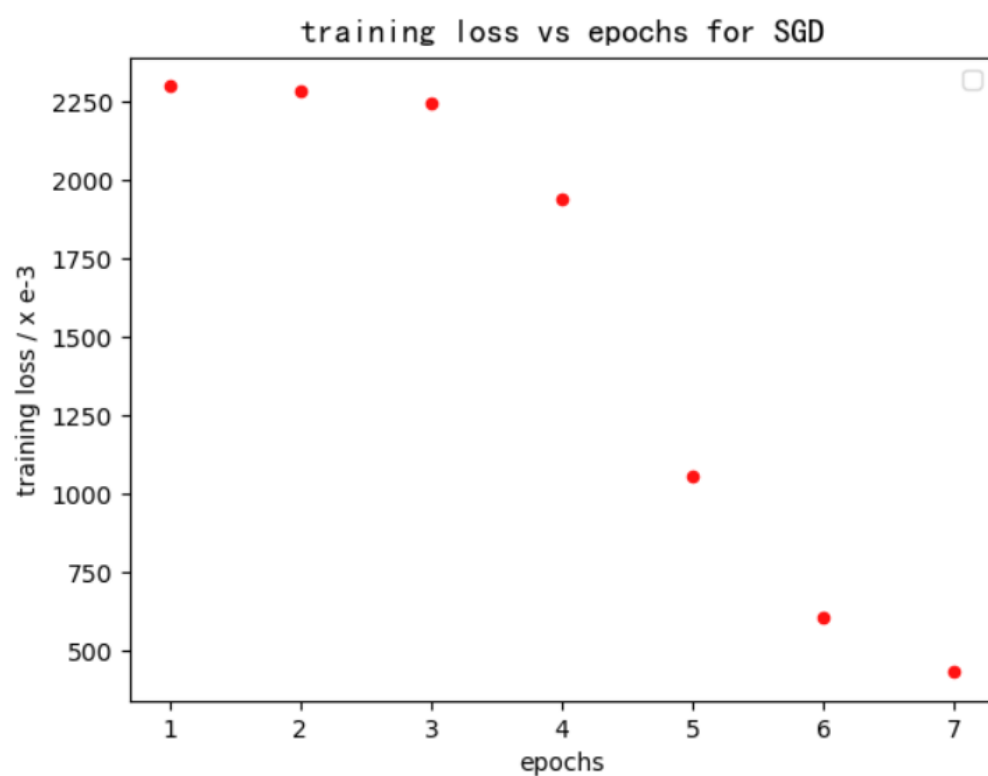
动量 momentum=0.9

总准确率：90%

## Loss 可视化

对应于第 3 题第 4 问。

(1) 当优化算法为 SGD，loss-epochs 曲线如下



(2) 当优化算法为 adam, loss-epochs 曲线如下

