

# 工作周报

(week 3)

## 目录

- 手写识别
  - ENVIRONMENT
  - 程序
  - 结果
  - REFERENCE
    - Pytorch
    - D2L
- 单目测距

## 手写识别

### ENVIRONMENT

在运行手写识别有关代码前，需要先安装conda并且建立一个conda虚拟环境。命令如下：

```
conda create -n pytorch python=3.9
```

在建立虚拟环境后激活该环节，并在环境中安装相关python库。激活命令如下：

```
conda activate pytorch
```

安装python相关库的命令如下：

```
conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch
conda install -n pytorch ipykernel --update-deps --force-reinstall
conda install -c conda-forge matplotlib
conda install pandas
conda install scikit-learn
```

### 程序

基于MNIST数据库训练手写识别的程序如下：

[用pytorch通过MNIST数据库实现手写识别](#)

在该程序中，并没有使用相对复杂的卷积神经网络，只是简单使用了两个二维卷积层，并且直接和两个线性层连接组成最后的神经网络。理论上来说，这里直接使用LeNet效果会更好，最终结果会更快收

敛，但是使用简单的神经网络计算量更少，学习一个周期的时间更短。

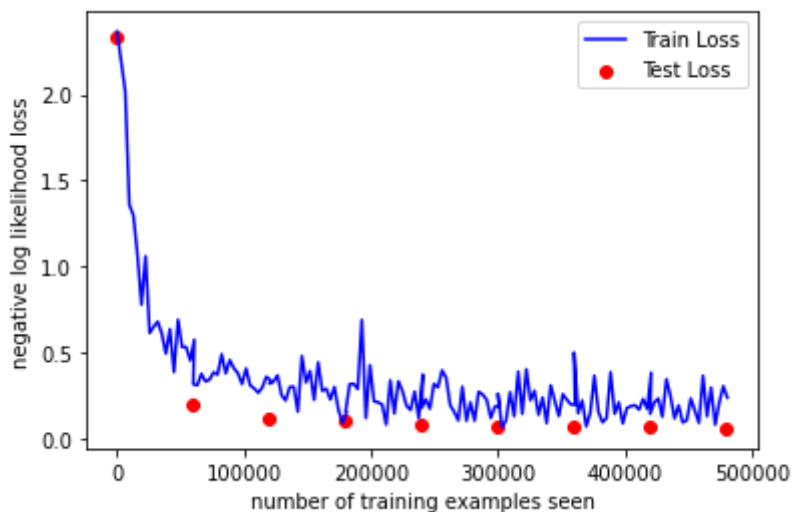
```
Net(  
  (conv1): Conv2d(1, 10, kernel_size=(5, 5), stride=(1, 1))  
  (conv2): Conv2d(10, 20, kernel_size=(5, 5), stride=(1, 1))  
  (conv2_drop): Dropout2d(p=0.5, inplace=False)  
  (fc1): Linear(in_features=320, out_features=50, bias=True)  
  (fc2): Linear(in_features=50, out_features=10, bias=True)  
)
```

## 结果

在经过八次epoch的训练后，准确率的变化如下表所示：

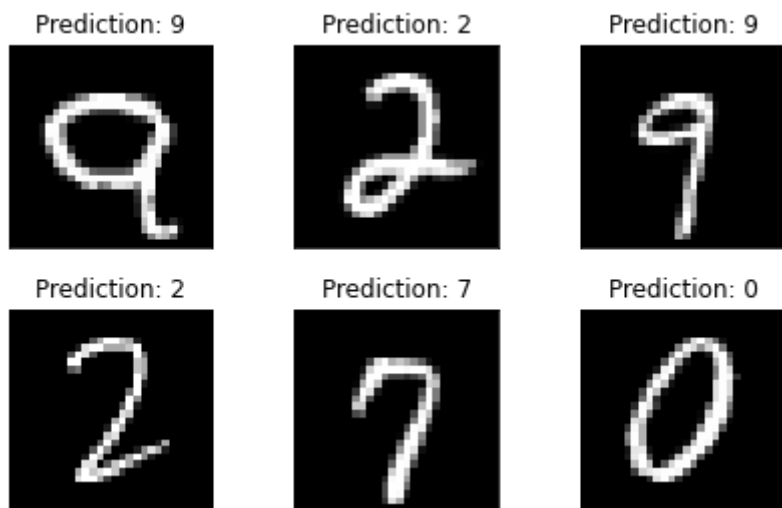
Epoch	Test Loss	Accuracy (number)	Accuracy (%)
1	0.1903	9421/10000	94%
2	0.1199	9622/10000	96%
3	0.0987	9693/10000	97%
4	0.0826	9749/10000	97%
5	0.0715	9770/10000	98%
6	0.0636	9799/10000	98%
7	0.0648	9800/10000	98%
8	0.0576	9817/10000	98%

```
Test set: Avg. loss: 0.1903, Accuracy: 9421/10000 (94%)  
Test set: Avg. loss: 0.1199, Accuracy: 9622/10000 (96%)  
Test set: Avg. loss: 0.0987, Accuracy: 9693/10000 (97%)  
Test set: Avg. loss: 0.0826, Accuracy: 9749/10000 (97%)  
Test set: Avg. loss: 0.0715, Accuracy: 9770/10000 (98%)  
Test set: Avg. loss: 0.0636, Accuracy: 9799/10000 (98%)  
Test set: Avg. loss: 0.0648, Accuracy: 9800/10000 (98%)  
Test set: Avg. loss: 0.0576, Accuracy: 9817/10000 (98%)
```



最后的识别准确率大致能达到为98%左右，想要进一步提高准确率就需要构建更优的神经网络或者使用一些技巧来促进神经网络更加收敛。

最后，给出关于手写识别示例作为参考：



## REFERENCE

### Pytorch

[linear Neural Network](#)

[LeNet5 Neural Network](#)

[simple CNN](#)

### D2L

[LIMU\\_d2l, install d2l\(limu\)](#)

# 单目测距