

实验一：手写数字识别

一、实验目的

1. 掌握卷积神经网络基本原理
2. 掌握 PyTorch（或其他框架）的基本用法以及构建卷积网络的基本操作
3. 了解 PyTorch（或其他框架）在 GPU 上的使用方法

二、实验环境 本次编程实验主要基于miniconda和pytorch框架进行，使用的各个库的版本号如下：

- python 3.7.16
- pytorch 1.10.0
- torchvision 0.11.0
- numpy 1.21.6
- tensorboard 2.11.2
- cuda 11.8

三、网络架构及参数设置 最终的网络架构使用了LeNet模型取得比较好的效果，中途也利用了实验指导书提供的卷积神经网络来进行实验。实验指导书CNN 流程：卷积(Conv2d)-> 激励函数(ReLU)->池化(MaxPooling)->卷积(Conv2d)-> 激励函数(ReLU)->池化(MaxPooling)->展平多维的卷积成的特征图->接入全连接层(Linear)->输出，最终正确率达到99%左右，满足大于98%的条件 最终结果参数设置

- 优化器：采用Adam优化器
- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数
- 超参设置：batch_size=256, num_epochs=30, lr=0.01
- 模型：LeNet

LeNet模型介绍

LeNet是发布最早的卷积神经网络之一，它结构简单，只有五层，包括两个卷积层和三个全连接层。该网络在当时的一个主要应用场景是手写数字识别。该网络中，第一个卷积层有六个不同的卷积核，每个卷积核都将接收输入图片并对其进行卷积操作和激活，得到特征图的一个通道的输出，所以多个卷积核可以改变特征图的通道数。汇聚层（池化层）不能改变通道数，也没有参数可以学习，只能缩小特征图的尺寸。第二个汇聚层的输出展平成向量后输入全连接层，经过三个全连接层的计算后输出10种可能结果的概率。概率最大的输出类别即为该网络给出答案

四、实验过程及结果 在取得本次实验的最终结果前，首先我进行了几次调参实验。

实验一： 在本次代码中，采用的参数设置为

- 优化器：采用SGD随机梯度下降优化器
- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数
- 超参设置：batch_size=256, num_epochs=30, lr=0.01

同时使用LeNet模型，得到的结果如下图所示

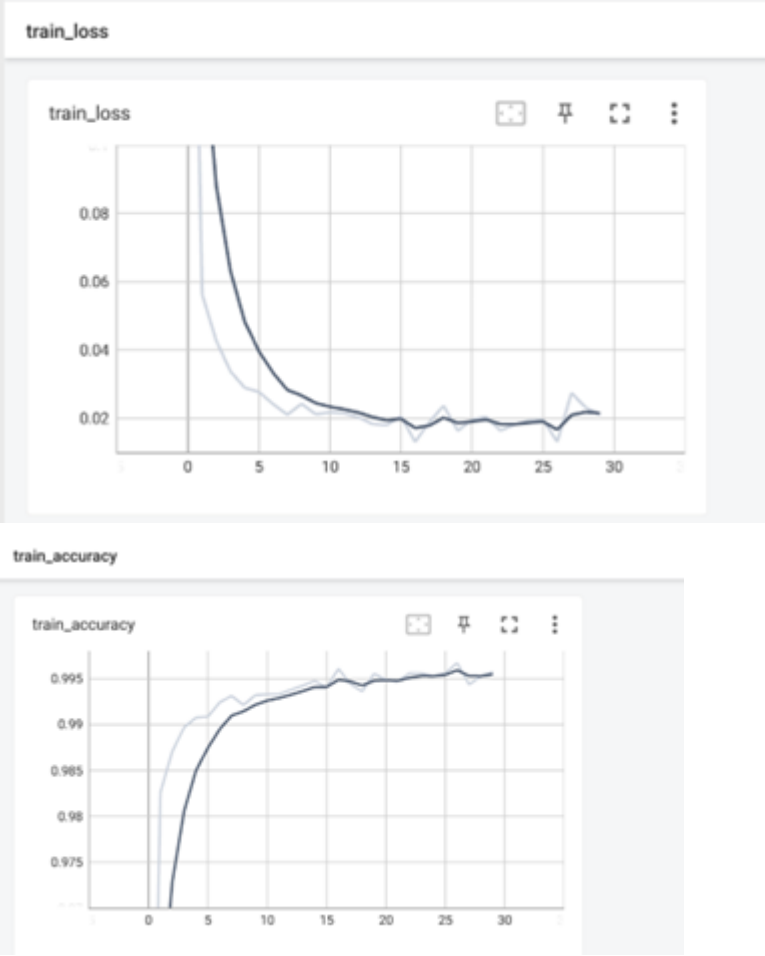
```
整个训练集上的loss为 0.114, 整个训练集上的准确率为 0.965, 整体测试集上的准确率为 0.964
现在是第24轮训练
整个训练集上的loss为 0.110, 整个训练集上的准确率为 0.967, 整体测试集上的准确率为 0.967
现在是第25轮训练
整个训练集上的loss为 0.105, 整个训练集上的准确率为 0.968, 整体测试集上的准确率为 0.967
现在是第26轮训练
整个训练集上的loss为 0.102, 整个训练集上的准确率为 0.970, 整体测试集上的准确率为 0.964
现在是第27轮训练
整个训练集上的loss为 0.100, 整个训练集上的准确率为 0.970, 整体测试集上的准确率为 0.967
现在是第28轮训练
整个训练集上的loss为 0.096, 整个训练集上的准确率为 0.971, 整体测试集上的准确率为 0.972
现在是第29轮训练
整个训练集上的loss为 0.093, 整个训练集上的准确率为 0.971, 整体测试集上的准确率为 0.967
模型已保存
```

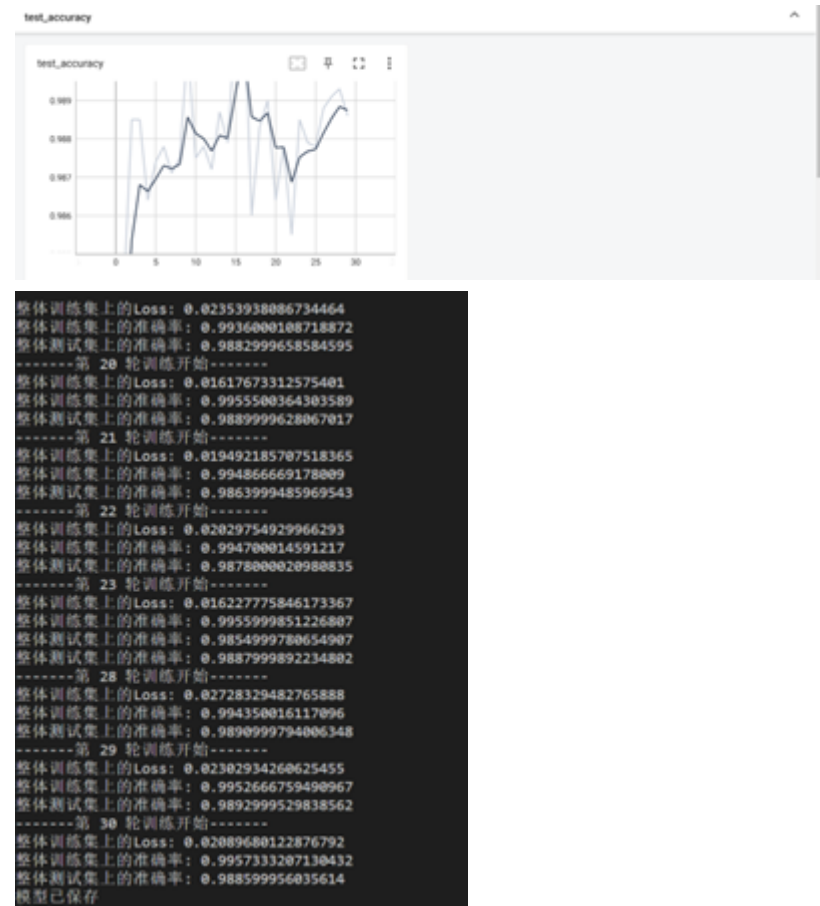
由此可见，训练完后测试集准确率仅达到96.7%，明显迭代次数不够，优化器选择SGD收敛速度较慢。

实验二：在本次代码中，采用的参数设置为

- 优化器：采用Adam梯度下降优化器
- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数
- 超参设置： batch_size=256, num_epochs=30, lr=0.01

采用实验指导书提供的网络模型，得到的训练损失，训练准确率，测试准确率如下图所示：



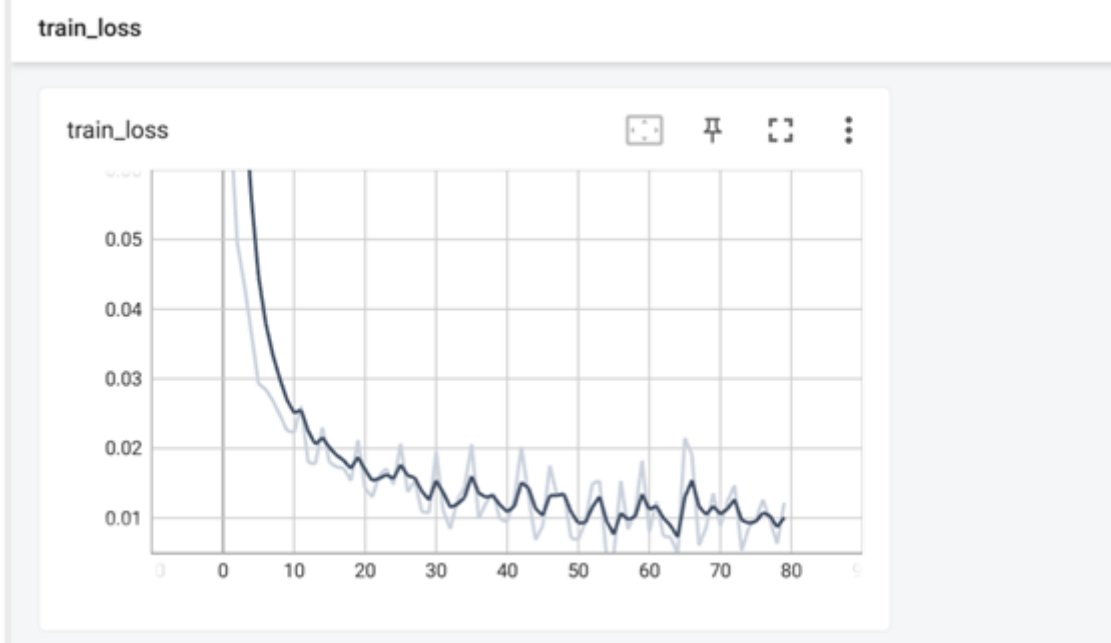


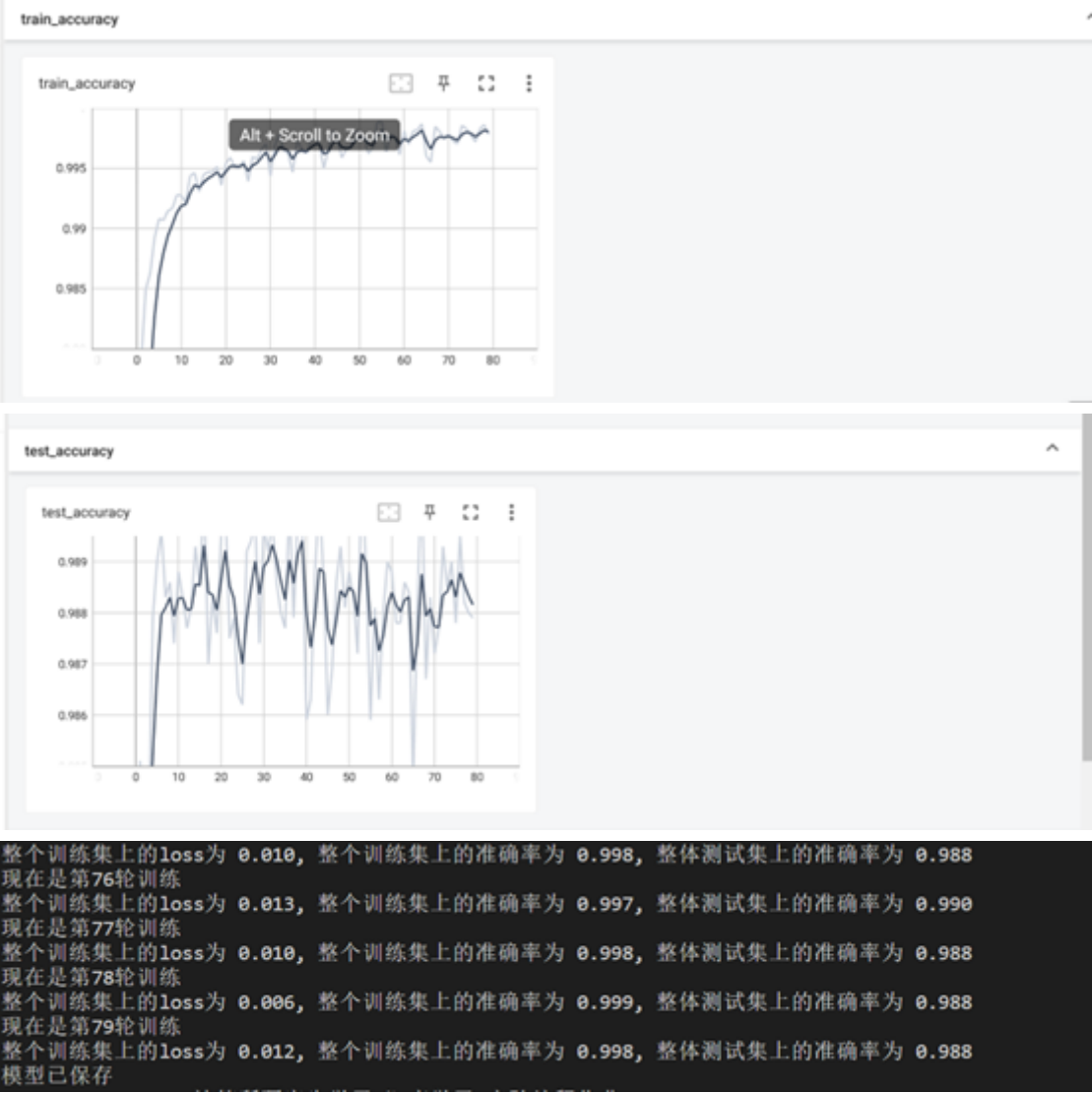
测试准确率达到98%以上要求，但尚未能达到99%。

实验三：也即最终实验，参数设置同实验二，即：

- 优化器：采用Adam梯度下降优化器
- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数
- 超参设置： batch_size=256, num_epochs=30, lr=0.01

仍然采用LeNet模型，得到的训练损失，训练准确率，测试准确率如下图所示：





由此可见，测试准确率达到98%以上要求，并稳定在98.8%左右，并能达到99%准确率，所以我们最终选择 LeNet网络模型和超参数配置作为最终结果。

小结 通过这次实验，自己学会了利用 Pytorch 深度学习框架实现手写数字识别，认识了LeNet网络模型，增加了对于卷积神经网络的理解。对于深度学习的整个流程，包括数据预 处理、网络模型构建、模型训练、结果可视化等方面都有了更深层次的理解，代码的组织编写水平也有了很大的提高，对于后续实验起到了很好 的启发作用