

# 实验一 手写数字识别报告

实验一是使用MNIST手写数字数据集进行训练和预测，实现测试集准确率达到98%及以上。本实验主要有以下目的：

- 掌握卷积神经网络基本原理
- 掌握主流框架的基本用法以及构建卷积神经网络的基本操作
- 了解如何使用GPU

实验报告将从实验环境、网络架构、模型训练、实验结果4个方面进行阐述。

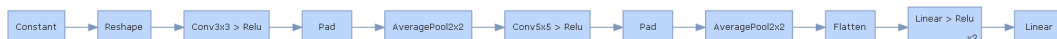
## 实验环境

本实验采用 Anaconda3 + Pycharm 框架进行开发，深度学习框架采用 PyTorch 框架，各类版本号如下：

- python 3.7.10
- pytorch 1.8.1 GPU
- CUDA 10.2
- cudnn 8.2.0
- tensorboard 2.4.1
- tensorboardX 2.2
- hiddenlayer 0.3

## 网络架构

构建卷积神经网络，采用与 LeNet 类似的结构，使用 hiddenlayer 库可视化网络，具体结构如下图所示：



其中：

- CNN1：
  - 卷积：卷积核尺寸为3\*3，步长为1，填充为1；
  - 激活：采用 ReLU 激活函数；
  - 池化：采用平均池化，尺寸是2\*2，步长为2；
- CNN2：
  - 卷积：卷积核尺寸为5\*5，步长为1；
  - 激活：采用 ReLU 激活函数；
  - 池化：采用平均池化，尺寸是2\*2，步长为2；
- FC1：
  - 展平：Flatten 操作
  - 线性层：两层，采用 ReLU 激活函数；

## 模型训练

对所构建的模型使用 GPU 训练，采用梯度下降法优化模型，损失函数使用交叉熵损失。具体模型参数设置如下所示：

- 梯度更新：采用 SGD 随机梯度下降的方法；

- 损失函数：采用 CrossEntropyLoss 交叉熵损失函数；
- 超参设置：batch\_size=256, num\_epochs=100, 学习率 lr=0.01

## 实验结果

使用 tensorboardx 库对实验结果可视化，包括训练的损失、训练准确率、测试准确率，其中测试准确率最终达到了**98%**以上，结果分别如下图所示：

图1 TrainLoss

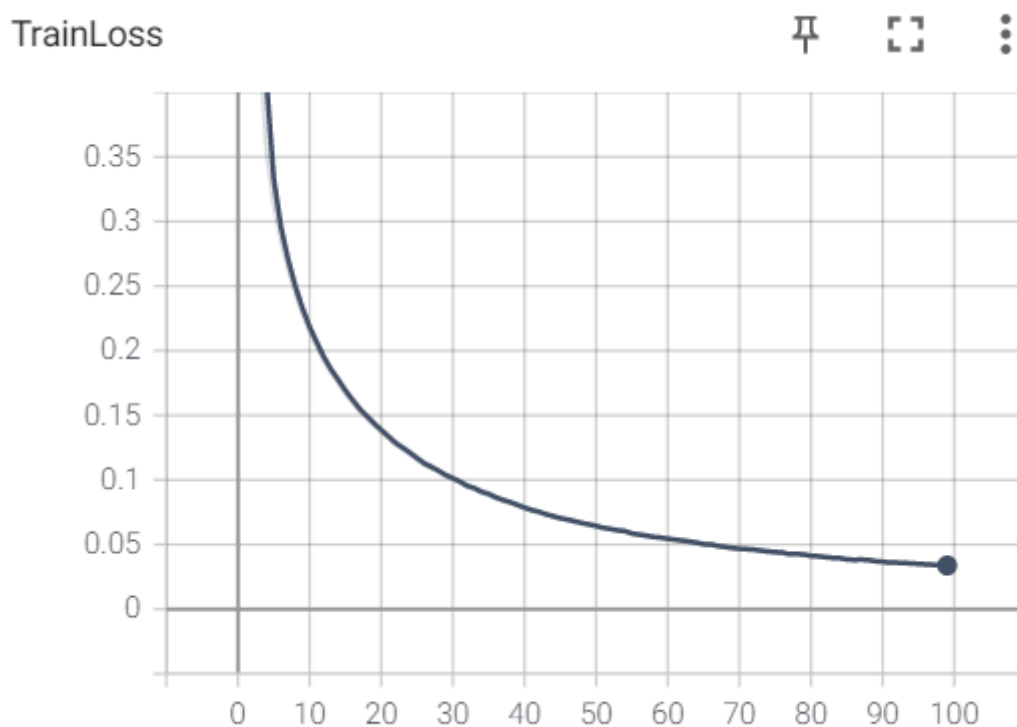


图2 TrainAccuracy

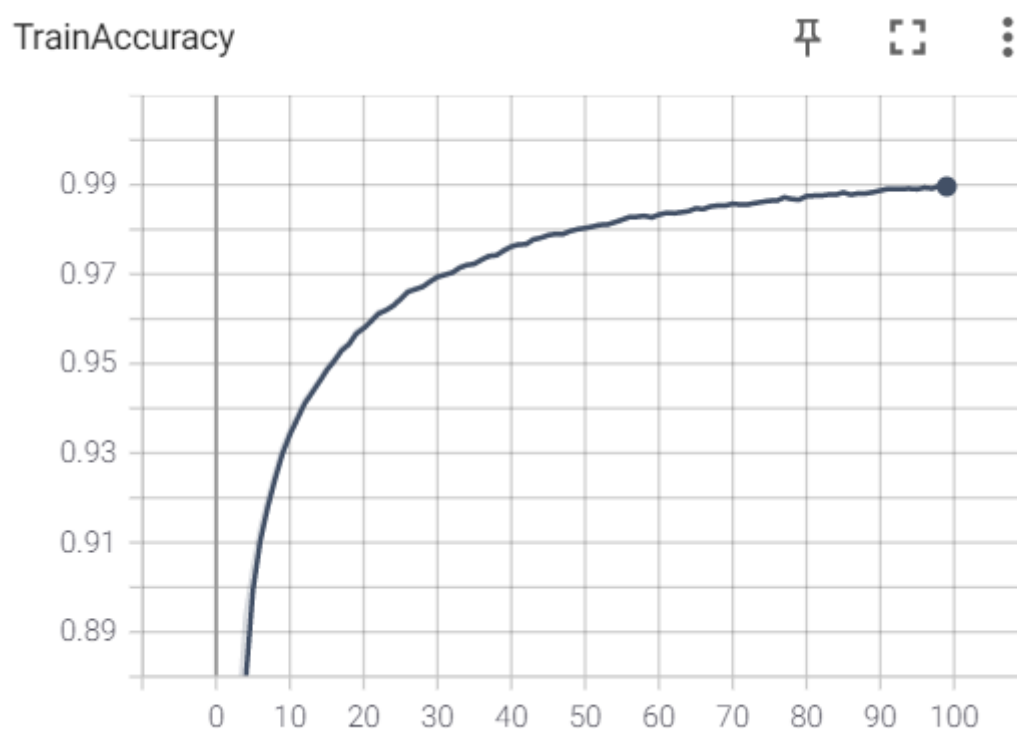


图3 TestAccuracy



## 小结

通过这次实验，我学会了如何利用 Pytorch 深度学习框架实现一段简单的代码，理解了计算图的意义。不仅增加了对于卷积神经网络的理解，还提高了我的动手能力。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，对于后续实验起到了很好的启发作用。最后特别感谢徐老师及助教的培养与付出！