**基于LeNet-5的MNIST训练与测试**

## 实验说明

本实验基于PyTorch框架搭建LeNet-5模型，完成了LeNet-5模型在手写数字数据集MNIST上的训练与测试工作。训练共进行20轮，在训练轮次达到10轮以后，loss值和准确率已经基本饱和，验证集上的准确率能够达到99%。后10轮的训练中，学习效果比较有限。

本实验在测试时绘制了混淆矩阵，并进行了模型特征图的可视化，绘制了训练集和验证集上的损失和准确率曲线。

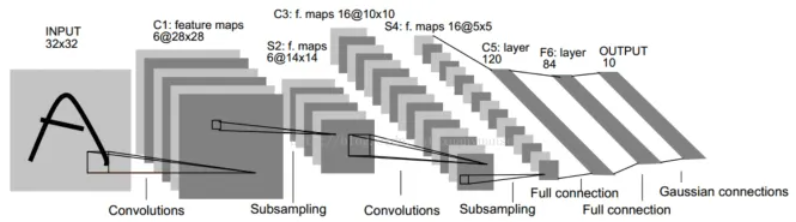
## 数据集准备与介绍

MNIST数据集是一个手写数字数据集，共有7万张手写数字图片，其中6万张是训练集图片、1万张是测试集图片。每张图片是28\*28大小的单个手写数字图片，且是灰度图片。

本次实验使用了torchvision的datasets模块，在第一次时指定download为true即可自动下载数据集到指定路径，本实验将数据集放在训练代码文件的同级data目录下。数据集只需下载一次即可。

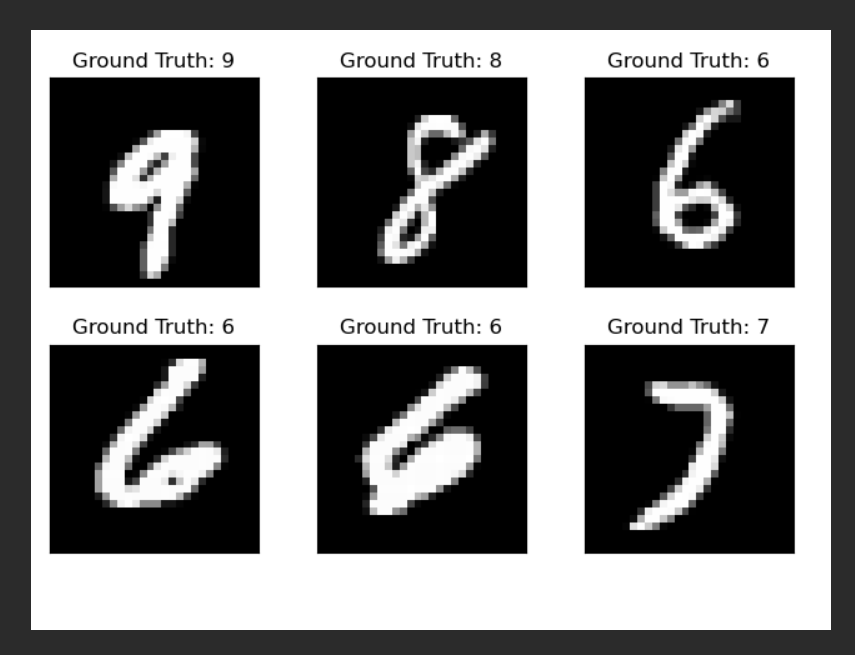
## 模型介绍与搭建

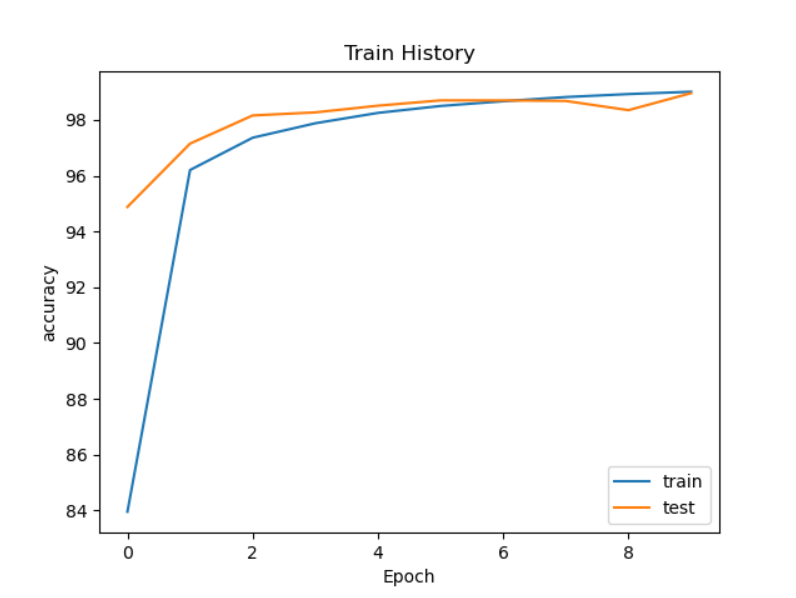
本次实验使用了LeNet-5模型来进行训练，这是卷积神经网络早期的一个经典模型，被认为是卷积神经网络的开创性工作之一。

LeNet-5的基本结构包含7层网络结构，包括2个卷积层、2个池化层、2个全连接层和输出层，模型使用的卷积核大小为5\*5。下图为该模型的基本结构：

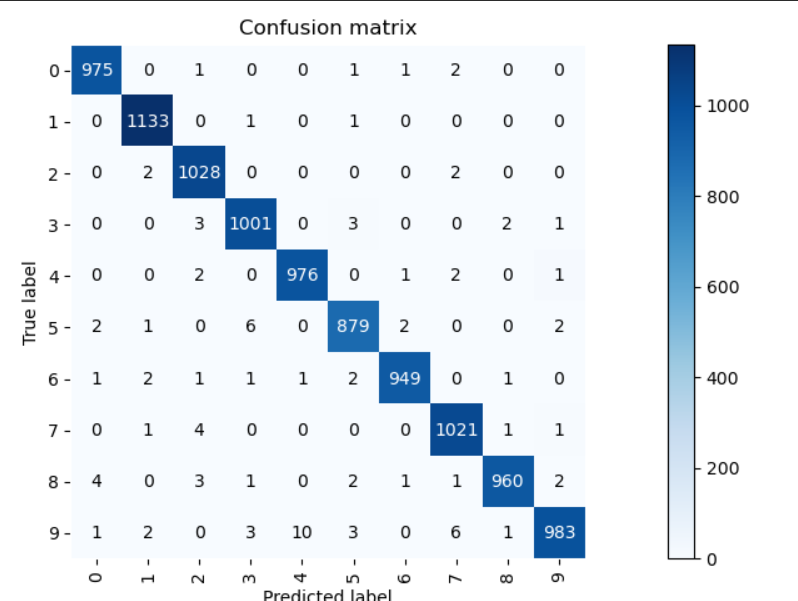
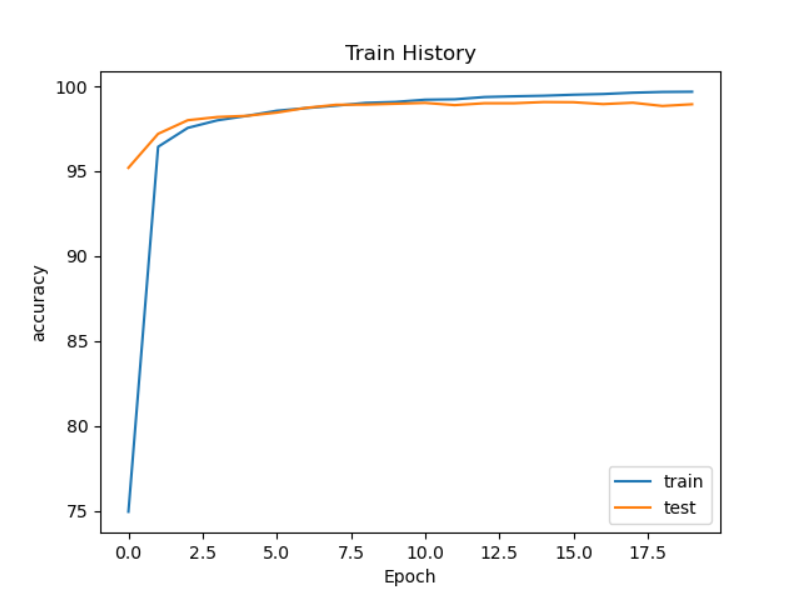
该模型原本设计的输入图片大小是32\*32的，但本实验中用到的数据集图片大小是28\*28的，因此不同于原本模型在第一个卷积层使用的填充为0，本实验将第一个卷积层的填充设为2。这样即可与原模型在第一次卷积后保持相同大小的输出，即6\*28\*28。

## 训练与测试

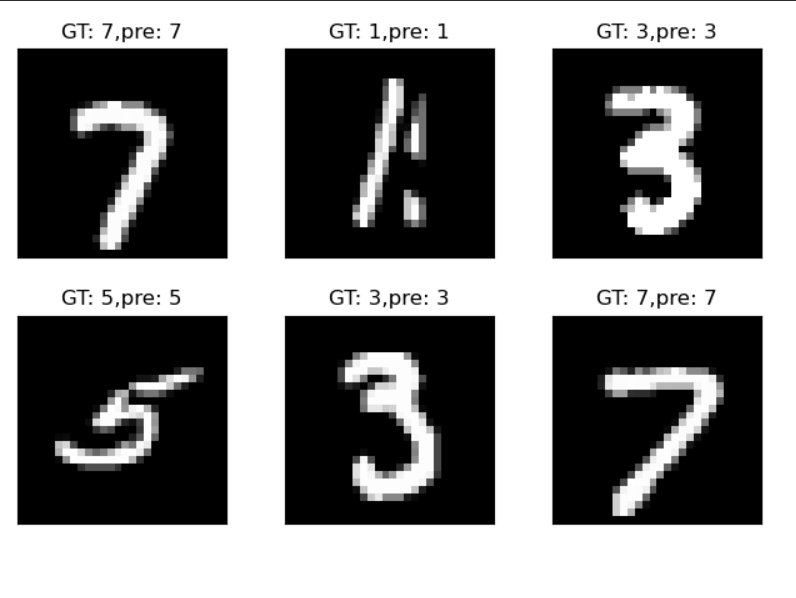
本实验在读取数据集后进行了数据集的可视化，展示了读取的图像和对应的标签，对应于代码中的show\_pic函数，结果如下图所示：

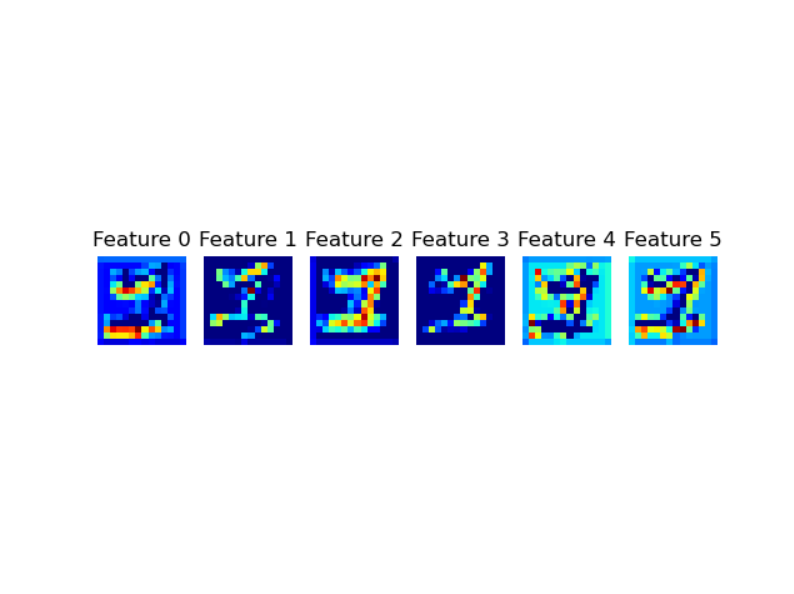
在训练过程中，除了控制台的进度输出外，训练结束后还绘制了模型的准确率曲线，对应于代码中的show\_train\_history函数，结果如下图所示：

本次训练共10轮，训练集上的精度稳步增长，测试集上的表现与训练集相差不大，看上去模型还没有过拟合，并且可能还有提升的空间。于是加大训练轮次，进行20轮训练。在10-20个轮次之间，训练集上的准确率还能提高，但测试集上的准确率却在原地打转，逐渐被训练集拉开距离。可能是模型开始过拟合了。下图是20轮训练的准确率曲线：

此外，在实验中绘制了测试集上的混淆矩阵，对应于代码中的confusion\_matrix函数和plot\_confusion\_matrix函数，结果如下图所示：

从混淆矩阵中可以清楚的看到模型在作预测时的情况。并且做了模型预测结果的可视化，其中gt代表真值，pre代表预测值。如下图所示：

最后，准备28\*28大小的测试图片进行测试，将训练好的模型载入并进行特征图可视化来观察每一层特征，测试图片和特征图可视化如下所示：



## 运行说明

本实验依托于PyTorch框架完成，运行前应确保完成PyTorch的安装。在运行前应确认同级文件中的data目录下是否有MNIST数据集，若没有，则可以将download参数置为True以进行下载。此外，本实验中用到的可视化函数和辅助函数在使用后均进行了注释，若要再次使用需在对应位置解开注释。