# 计算机视觉实验报告：实现 LeNet-5 在MNIST 数据集上的训练和测试

## 实验目的

使用 Pytorch 来实现 LeNet-5 模型，并用它来解决 MNIST数据集的识别。MNIST 数据集是一个标准的计算机视觉基准数据集。

## 实验环境

编程语言：Python

深度学习框架：PyTorch

硬件：CPU / GPU（如果可用）

数据集：MNIST

## 实验方法

### 模型设计

LeNet-5 模型是包含两个卷积层、两个池化层和三个全连接层的 CNN 模型。模型的输入是大小为 28x28 的灰度图像，输出是一个包含 10 个概率值的向量，每个值代表一个数字类别。

### 数据预处理

使用 torchvision.datasets.mnist 加载 MNIST 数据集，并通过 ToTensor 转换将其转换为 PyTorch 张量。

### 训练过程

使用交叉熵损失函数和随机梯度下降（SGD）优化器。

模型在训练集上进行 100 轮迭代训练。

每个批次大小为 256。

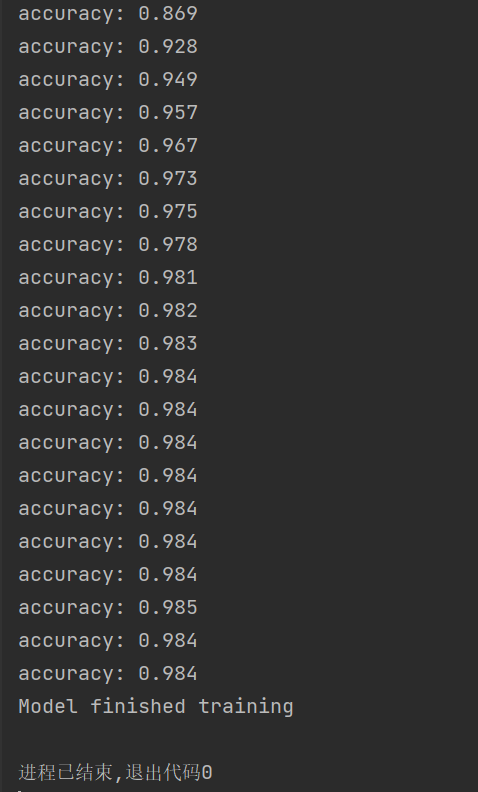
学习率设置为 0.1。

### 评估过程

在每个训练轮次后，我们在测试集上评估模型的性能，使用准确率作为性能指标。

## 实验结果

经过 100 轮的训练，模型在 MNIST 测试集上达到了较高的准确率。实验过程中，准确率逐渐提高，最终稳定在一个值附近。



## 实验分析

CNN 模型通过自动学习图像的局部特征进行分类，这比传统的机器学习方法更为有效。

使用 ReLU 激活函数和最大池化层有助于提取特征并减少计算量。

模型的准确率随着训练轮次的增加而提高，表明模型在学习数据集上的分布。

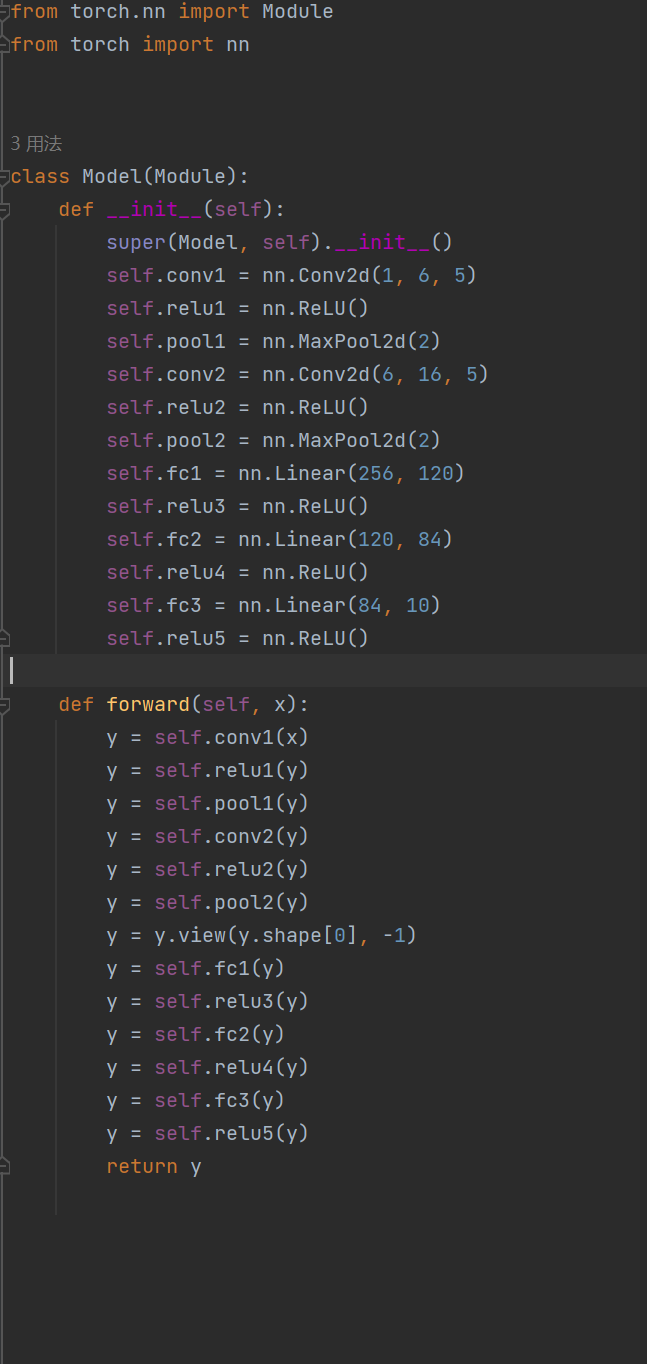
实验中遇到的挑战包括过拟合和欠拟合，通过适当的正则化和调整模型复杂度来解决。

## 实验结论

LeNet-5 模型能够在 MNIST 数据集上实现高准确率的分类。此外，实验还展示了 PyTorch 框架在构建和训练深度学习模型方面的易用性。

## 代码实现

### 模型代码



### 训练代码

