

**深度学习课程**

**实验一：手写数字识别实验报告**

**学生姓名： 李增圣**

**学生学号： 2024E8009282002**

**指导教师： 徐俊刚 教授 中国科学院大学计算机科学与技术学院**

**培养单位： 中国科学院大学材料科学与光电技术学院**

**2025年 4 月**

**摘 要**

本实验使用 PyTorch 框架实现了一个卷积神经网络（CNN）模型，进行 MNIST 手写数字数据集的分类任务。通过训练该模型，旨在达到测试集准确率超过98%的目标。在实验中，首先下载并加载了 MNIST 数据集，定义了一个包含两个卷积层、两个池化层和一个全连接层的卷积神经网络。优化器采用 Adam，损失函数选择交叉熵损失函数。训练过程中，我们在每个 epoch 中评估训练和测试集的损失及准确率，并可视化了每100个批次的损失和准确率变化以及每个训练周期（epoch）的损失和准确率变化。实验结果表明，模型在训练初期能够快速收敛，损失快速下降，准确率迅速提升，最终在测试集上取得了超过98%的准确率，验证了卷积神经网络在图像分类任务中的优越性能。

**关键词：**卷积神经网络，PyTorch，MNIST，CNN，CN手写数字识别

**目 录**

[第1章 实验背景 3](#_Toc29812)

[第2章 实验过程 4](#_Toc14054)

[2.1 导入相关库 4](#_Toc29073)

[2.2 下载并加载数据集 4](#_Toc21950)

[2.3 定义模型 4](#_Toc24092)

[2.4 定义训练和测试函数 5](#_Toc4606)

[第3章 实验结果 7](#_Toc3161)

[3.1 训练参数 7](#_Toc9332)

[3.2 实验结果 7](#_Toc11017)

[3.2.1 每100个批次的损失和准确率 7](#_Toc15692)

[3.2.2 每个训练周期（Epoch）的损失和准确率 8](#_Toc7279)

[3.3 结果分析 8](#_Toc25969)

# 实验背景

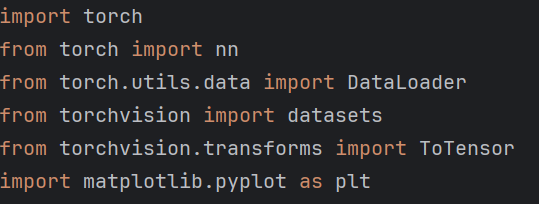
本实验使用 PyTorch 框架实现一个卷积神经网络，进行 MNIST 手写数字数据集的分类任务。训练并评估其在 MNIST 数据集上的分类性能，目标是在测试集上达到98%以上的准确率。

MNIST 数据集包含了手写的数字（0-9），并被广泛用于图像分类任务的研究中。MNIST数据集由60000张训练集图片与10000张测试集图片构成，图片为28\*28的灰度图像。

# 实验过程

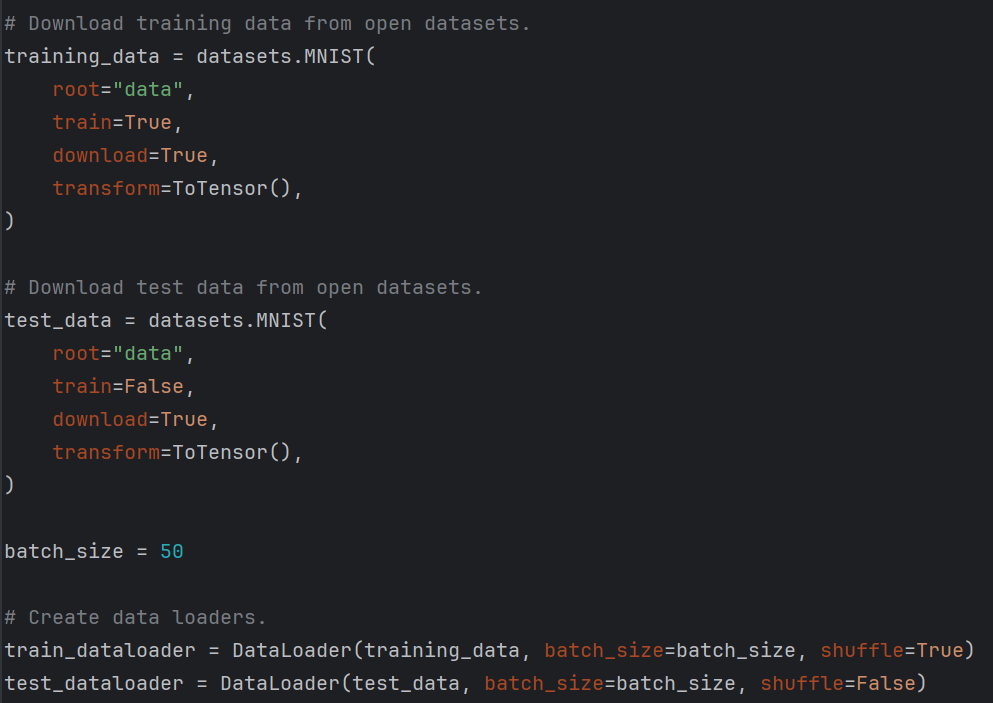
## 导入相关库

实验使用pytorch框架，torchvision用于下载与加载数据集。



## 下载并加载数据集

DataLoader 是 PyTorch 提供的用于批量加载数据的工具，可以自动将数据集划分为多个小批次



## 定义模型

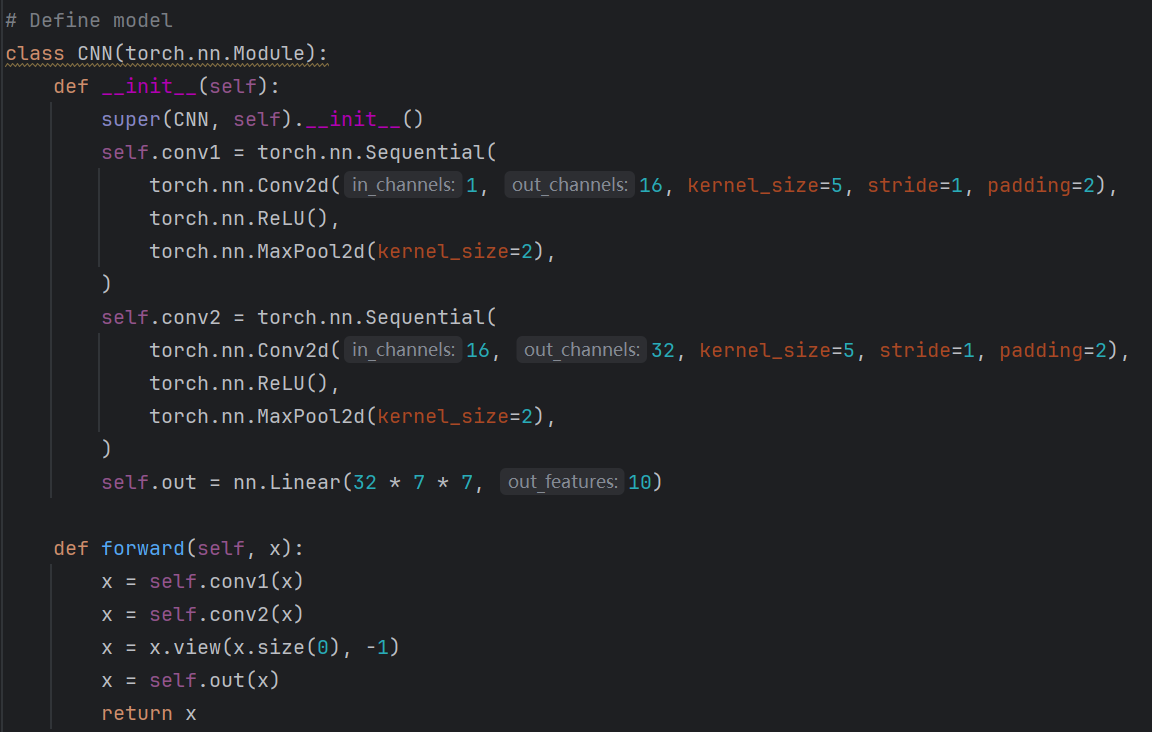
模型继承torch.nn.Module,需要调用父类的初始化函数。这里创建了两个卷积层，两个池化层，一个全连接层。卷积层的5个参数分别为输入通道，输出通道，卷积核大小，步长，填充。填充的作用是控制卷积后图形的大小。池化层选择的是最大池化。全连接层的两个参数分别是将卷积输出图像展平后的节点数乘输出通道数与分类的类别数。

定义第一个卷积层：输入通道为1（灰度图像），输出通道为16，卷积核大小为5x5，步幅为1，填充为2。

定义第二个卷积层：输入通道为16，输出通道为32，卷积核大小为5x5，步幅为1，填充为2。

全连接层，将卷积层输出的特征图展平并送入全连接层进行分类。

前向传播过程：输入通过第一个卷积层、输入通过第二个卷积层、将卷积层输出的特征图展平为一维向量、输入到全连接层进行分类。

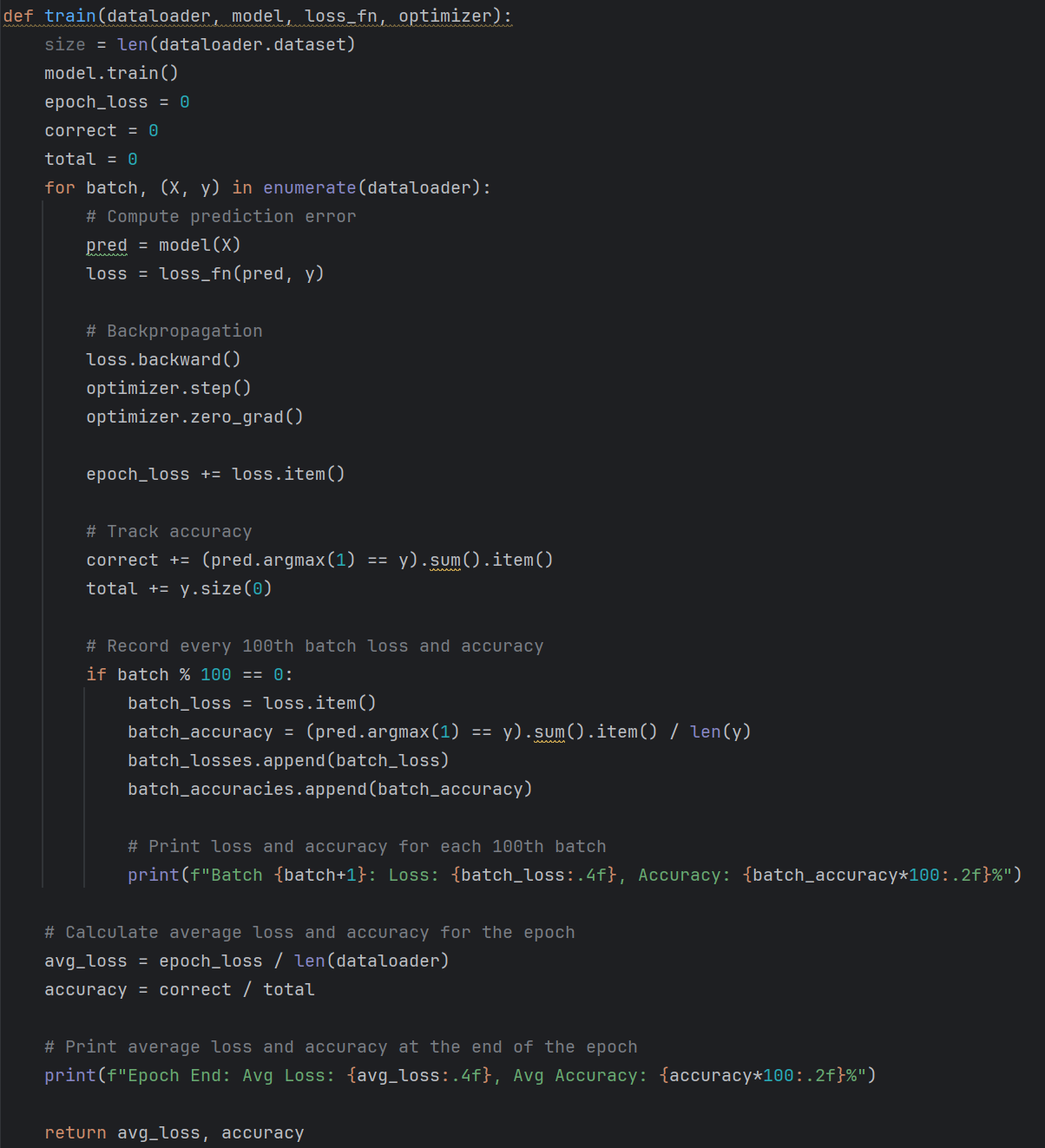


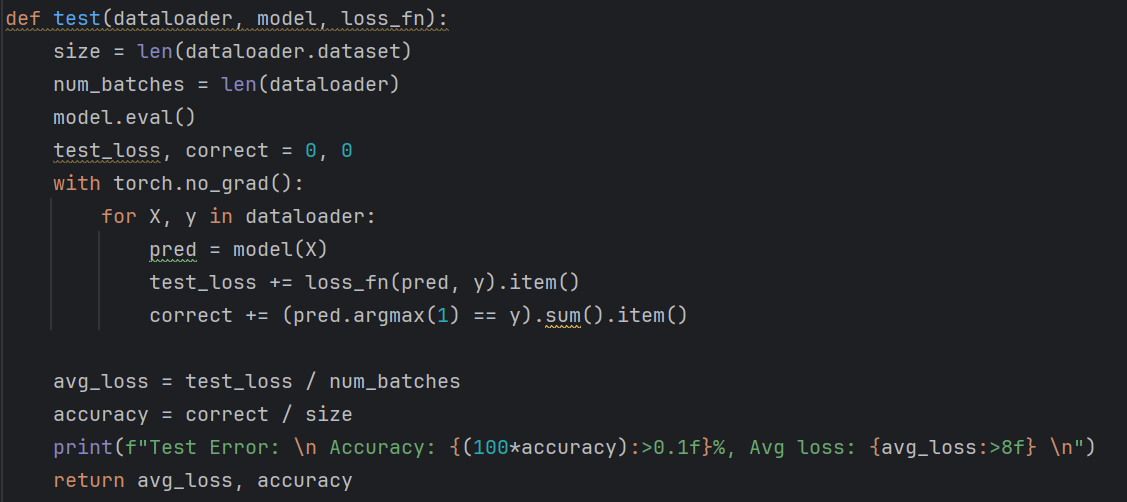
## 定义训练和测试函数

损失函数使用交叉熵损失，优化器使用Adam，batch\_size大小设置为50，训练5个epoch，学习率0.001。

创建损失函数，模型，优化器，获取训练集与测试集的Dataloader，将模型。训练函数：遍历训练集，计算每个批次的损失，并根据损失进行反向传播和优化更新模型参数。同时计算训练准确率。测试函数：计算每个测试批次的损失和准确率，并返回平均损失和准确率。

在每个 epoch 中，模型首先在训练集上进行训练，接着在测试集上进行评估。每个 epoch 的损失和准确率被存储，用于后续绘制图表。





# 实验结果

## 训练参数

优化器：Adam

损失函数：交叉熵

学习率：0.001

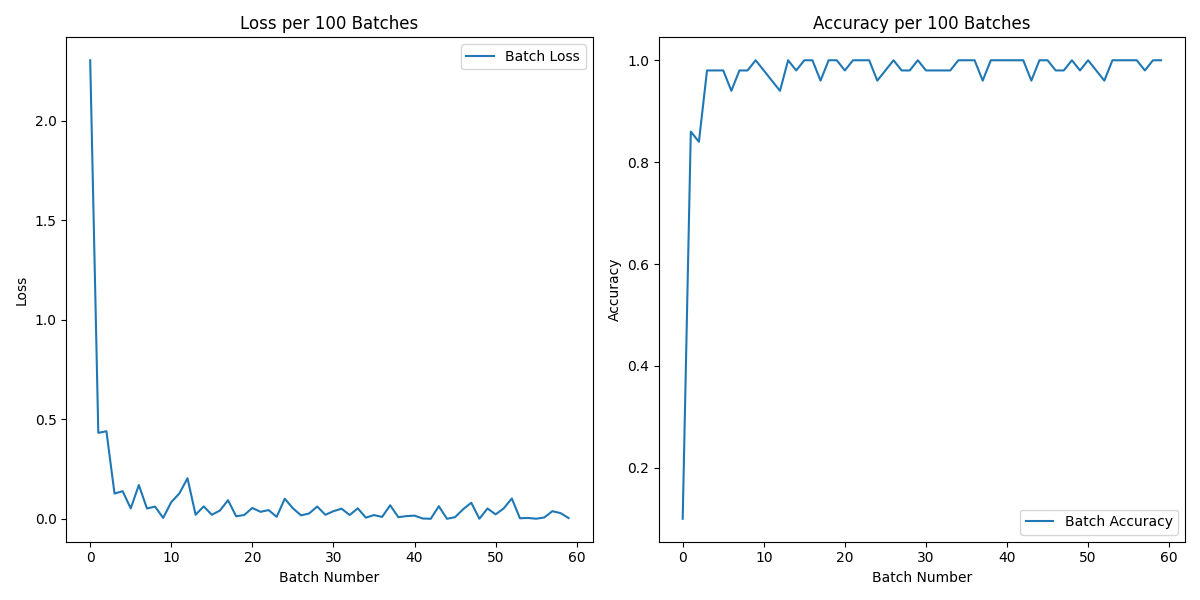
batch\_size:50

epochs:5

训练设备:GPU

## 实验结果

### 每100个Batch的损失和准确率

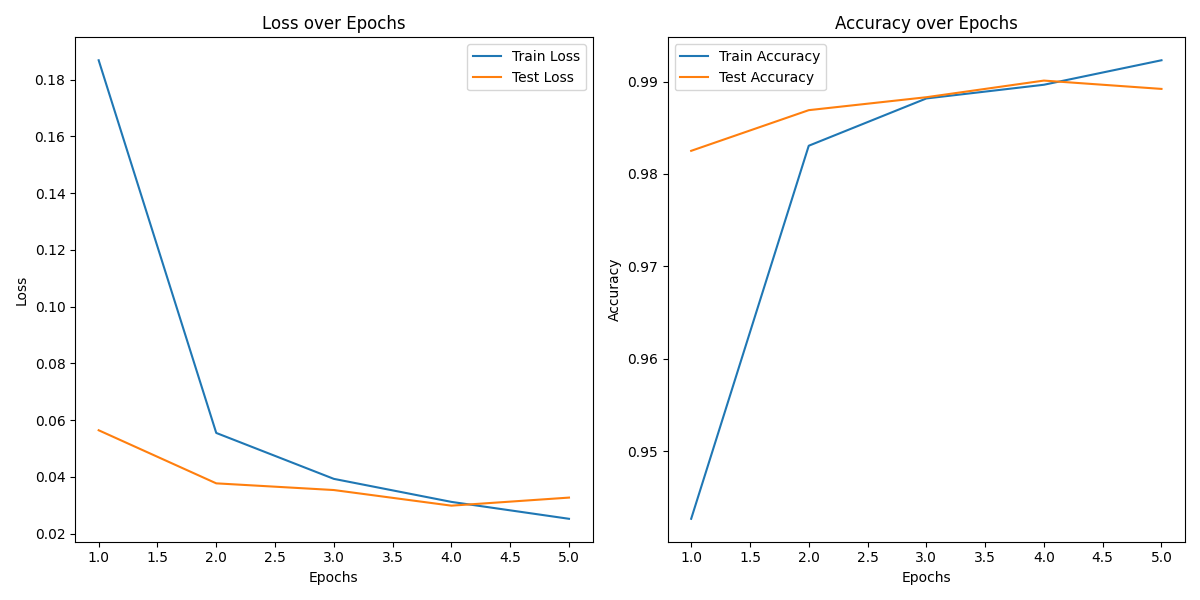


损失（Loss）：在训练的初期（前几个批次），损失显著下降，之后趋于稳定。最初的损失较高，主要是因为模型还没有学习到有效的特征，但随着训练的进行，损失值逐渐减小，表示模型逐步学会了如何有效地进行分类。

准确率（Accuracy）：从图中的右侧可以看到，准确率在前几个批次时有明显的提升，尤其是在前20个批次之后，准确率迅速达到较高水平（接近100%）。这表明，尽管训练批次较少，模型在这些批次内就能够很好地拟合训练数据。

这两个图表显示了训练过程中每100个批次的损失和准确率变化，说明模型在初期的学习速度较快，随着训练的深入，损失逐渐减小，而准确率则逐渐提高，接近其极限。

### 每个训练Epoch的损失和准确率



损失（Loss）：训练损失和测试损失均表现出了较快的下降趋势。特别是在第1个和第2个周期，损失大幅下降，说明模型在初期就能够有效地学习到数据特征。训练损失在后期趋于平稳，而测试损失保持相对较低的水平，显示出模型在测试集上的泛化能力。

准确率（Accuracy）：训练准确率和测试准确率的变化趋势类似，都表现出快速上升的趋势。训练准确率迅速达到接近100%，测试准确率在第2个周期后也接近98%以上，符合实验的目标（测试集准确率要达到98%及以上）。

这两个图表说明了模型在训练过程中逐步收敛，且在训练集和测试集上的表现都非常良好。测试准确率的稳定上升表明模型没有过拟合，并且具有良好的泛化能力。

## 结果分析

本次实验成功实现了基于卷积神经网络（CNN）对MNIST手写数字数据集的分类，且模型在测试集上的准确率达到98%以上，符合实验目标。实验中使用的模型能够快速学习数据特征，并在训练和测试数据上表现出色，展示了卷积神经网络在图像分类任务中的强大性能。