计算机视觉大作业：基于卷积神经网络（CNN）的手写数字分类

22122128，孔馨怡

摘　要：本项目旨在基于卷积神经网络（CNN）实现手写数字分类任务，其中采用了 LeNet-5 模型作为主要架构。通过对 MNIST 数据集进行训练和测试，评估模型在图像分类任务上的性能表现。关键技术包括超参数定义、数据处理流程、LeNet-5 模型构建以及训练和测试方法的实现。

关键词：卷积神经网络 CNN、LeNet、手写数字识别、图像分类、MNIST 数据集

1. 项目背景和意义

当我们面对海量的图像数据时，图像分类是一项至关重要的任务。传统的机器学习方法在处理图像分类时往往面临着维度灾难和特征提取的挑战，而卷积神经网络（CNN）却以其出色的性能和广泛的应用而备受关注。

CNN 作为一种深度学习模型，能够自动学习图像中的特征，无需手动设计特征提取器。这种能力使得 CNN 在图像分类任务上表现出色。它通过一系列卷积层和池化层来提取图像的局部特征，并通过全连接层将这些特征组合起来进行分类。这种分层特征提取的方式使得 CNN 能够有效地捕获图像中的空间结构和层次信息，从而在图像分类任务中取得了巨大的成功。

手写数字识别作为计算机视觉领域的一个重要问题，一直以来备受关注。它不仅涉及到图像处理、模式识别和机器学习等多个领域的交叉，更是许多实际应用的基础，比如自动识别银行支票上的金额、邮件分类等。本项目将使用 LeNet-5 模型来实现手写数字识别任务。

2. LeNet-5 模型原理

2.1 简介

LeNet-5 是由 Yann LeCun 在 1998 年提出的卷积神经网络模型，被认为是深度学习和卷积神经网络发展的先驱之一。也是 LeNet 系列中最为流行的模型之一。

2.2 原理简述

LeNet-5 的整体结构利用了卷积和池化操作的局部连接性和权值共享特性，使得模型对平移、缩放和旋转等图像变换具有一定的鲁棒性，成为了后续卷积神经网络发展的基础。

卷积层（Convolutional Layers）：LeNet-5 包含两个卷积层。这些层在输入图像上应用一系列卷积核（也称为过滤器），以提取图像的特征。每个卷积核在输入图像上滑动，执行卷积操作，生成特征图（Feature Maps）。第一个卷积层产生 6 个特征图，而第二个卷积层产生 16 个特征图。这些特征图捕捉了图像中的不同特征，如边缘、纹理等。

池化层（Pooling Layers）：用于减小特征图的空间尺寸，同时保留重要的特征。LeNet-5 包含两个平均池化层。池化操作通常是在每个特征图上执行的，它将每个区域的像素值聚合成单个值，从而减少计算量并提高模型的鲁棒性。

全连接层（Fully Connected Layers）：位于卷积和池化层之后，用于将卷积层和池化层提取的特征映射到输出类别。LeNet-5 包含两个全连接层，分别包含 120 个和 84 个神经元。这些层通过权重连接将输入特征映射到输出类别的概率分布。

激活函数（Activation Function）：在每个卷积层和全连接层之后，通常会应用非线性激活函数（如 ReLU），以增加网络的表示能力和非线性拟合能力。

Softmax 输出层：在 LeNet-5 的最后，通常使用 Softmax 函数将网络输出转换为类别概率分布，以便进行多类别分类。

2.3 对比与具体层级参数

与 LeNet-4 相比，主要区别在于输入图像的大小和卷积层的数量。LeNet-5 接受 32x32 大小的输入图像，并包含两个卷积层和两个平均池化层。第一个卷积层产生 6 个 28x28 大小的特征图，而第二个卷积层产生 16 个 10x10 大小的特征图。之后是两个全连接层，分别连接到 120 个和 84 个神经元，最终输出 10 个类别的预测结果。这种结构设计保留了 LeNet 系列的经典特征，同时通过增加卷积层的数量和调整输入图像的大小，进一步提高了模型的性能和表现能力。

下表提供了 LeNet-5 模型的网络层次结构、输入输出情况以及每个层次的主要参数。LeNet-5 共包含 7 个层次，包括 2 个卷积层、2 个池化层和 3 个全连接层。每个层次的输入、输出情况以及卷积/池化窗口大小和可训练参数数量都在表中列出：

网络层\简介	输入	输出	（卷积/池化）窗口	可训练参数
C1（卷积层）	32x32 灰度图像	6 个尺寸为 28x28 的 FeatureMap	6 个 5x5 卷积核	156
S2（池化层）	6 个尺寸为 28x28 的 FeatureMap	6 个尺寸为 14x14 的 FeatureMap	2x2 池化窗口	12
C3（卷积层）	6 个尺寸为 14x14 的 FeatureMap	16 个尺寸为 10x10 的 FeatureMap	16 个 5x5 卷积核	1516
S4（池化层）	16 个尺寸为 10x10 的 FeatureMap	16 个尺寸为 5x5 的 FeatureMap	2x2 池化窗口	32
C5（卷积层）	16 个尺寸为 5x5 的 FeatureMap	120 个尺寸为 1x1 的 FeatureMap	120 个 5x5 卷积核	48120
F6（全连接层）	120 个特征	84 个特征	None	10164
Output（输出层）	84 个特征	10 维向量	None	840

图 1. LeNet-5 模型网络层次结构

3. 核心技术实现

3.1 定义超参数

首先定义了一些关键的超参数。其中，批量大小（BATCH_SIZE）决定了每次模型更新时所处理的数据量，设备类型（DEVICE）用于指定训练时使用的硬件，若可用则选择 GPU，否则使用 CPU。训练轮次（EPOCHS）表示整个训练数据集被模型遍历的次数。这些超参数的选择直接影响了模型的训练效果和速度，是整个训练过程中需要仔细调节和优化的重要因素。

3.2 构建 transform

为了有效处理图像数据并为模型提供良好的输入，采用了 transforms.Compose 构建了一个处理 pipeline，包括了两个主要的预处理步骤：

一是将图像转换为张量（transforms.ToTensor()）：这一步将图像数据转换为 PyTorch 所需的张量格式。张量是 PyTorch 中表示数据的基本形式，它可以被直接用于模型的输入，并且支持自动求导等功能。

二是归一化处理（transforms.Normalize()）：归一化处理是将图像的像素值缩放到一个固定的范围内，以便更好地适应模型的训练过程。我通过传入均值和标准差来进行归一化处理，确保图像数据的分布符合模型的期望。

3.3 构建 LeNet-5 模型

首先定义了一个名为 Digit 的 PyTorch 模型类。在这个类中，使用了 nn.Module 作为基类，并在初始化方法中定义了 LeNet-5 模型的各个层次。

定义两个卷积层（self.conv1 和 self.conv2），分别接收输入图像并通过一系列卷积核提取图像的特征。这些卷积层使用了不同的核大小和输出通道数，以提取不同层次的特征信息。

定义了两个全连接层（self.fc1 和 self.fc2），用于将卷积层提取的特征映射到最终的类别标签上。这些全连接层通过学习特征之间的复杂关系来进行分类，从而实现对手写数字的识别。

在模型的 forward 方法中，对输入数据进行了一系列的处理和计算，包括卷积、激活函数 ReLU 的应用、池化等操作，最终得到了模型的输出结果。

3.4 定义训练方式

train_model 用于训练 LeNet-5 模型。在每个 epoch 中，该方法遍历训练集数据，对每个批次的数据进行处理。首先，将数据和对应的标签部署到指定的设备上，这里使用了 GPU 加速（如果可用）。然后将优化器的梯度初始化为零，以便进行参数更新。通过模型前向传播得到预测结果，并计算预测结果与真实标签之间的交叉熵损失。运用反向传播，将梯度传播回网络，并使用 Adam 优化器更新模型参数，以最小化损失函数的值。在训练过程中，输出周期性当前的训练损失，以便监控模型的训练情况和性能表现。

3.5 定义测试方法

test_model 方法用于评估模型在测试集上的性能。在评估过程中，将模型切换到评估模式，并禁止计算梯度，以节省计算资源。随后遍历测试数据集，将数据传送到设备上，并通过模型进行前向传播，得到预测结果。计算测试损失，这是模型在测试集上的平均损失。同时，还计算模型在测试集上的准确率，以评估模型的分类性能。最终，打印出测试损失和准确率，以便了解模型在实际数据上的表现情况。在接下来的实验结果和分析处详细介绍。

4. 实验结果及分析

4.1 实验平台与工具选择

本文实验环境参数如下：

主机配置： MacBook Pro 2022，芯片 Apple M2

编译环境： Anaconda Jupyter Notebook

本文的代码实现主要调用了 torch 库。

4.2 实验数据集

本次项目选用的是 MNIST 数据集，这是一个经典的、用于训练机器学习算法的数据集。它包含了 60000 个用于训练的示例和 10000 个用于测试的示例。每个示例都是一个 28x28 大小的灰度图像，代表了 0 到 9 之间的手写数字。MNIST 数据集已经成为了测试新算法性能的标准基准。

4.3 实验结果及分析

```
Train Epoch:1    Loss:2.305202
Train Epoch:1    Loss:0.067067
Test -- Average loss:0.0032,Accuracy :98.180

Train Epoch:2    Loss:0.011282
Train Epoch:2    Loss:0.000112
Test -- Average loss:0.0025,Accuracy :98.800

Train Epoch:3    Loss:0.031875
Train Epoch:3    Loss:0.055311
Test -- Average loss:0.0025,Accuracy :98.770

Train Epoch:4    Loss:0.001135
Train Epoch:4    Loss:0.006348
Test -- Average loss:0.0030,Accuracy :98.540

Train Epoch:5    Loss:0.000042
Train Epoch:5    Loss:0.000007
Test -- Average loss:0.0029,Accuracy :98.920

Train Epoch:6    Loss:0.000089
Train Epoch:6    Loss:0.000012
Test -- Average loss:0.0025,Accuracy :99.030

Train Epoch:7    Loss:0.001933
...
Train Epoch:10   Loss:0.000000
Train Epoch:10   Loss:0.000000
Test -- Average loss:0.0048,Accuracy :98.790
```

图 2. 代码打印结果呈现

在第一个 epoch，训练损失从 2.305202 降低到 0.067067，测试集的平均损失为 0.0032，准确率为 98.18%。随着训练的进行，训练损失逐渐减小，到第四个 epoch 时甚至下降到 0.001135，而测试集的平均损失也保持在一个较低的水平，准确率在 98.54%左右。在后续的训练过程中，虽然训练损失和测试损失有轻微波动，但总体上保持在一个较稳定的水平。最终在第十个 epoch，训练损失和测试损失都非常接近零，而测试集的准确率达到了 98.79%。

在经过 10 个 epoch 的训练后，表现出了较好的性能，训练损失和测试损失均较低，而测试集的准确率也达到了较高的水平，这表明模型在训练集和测试集上都取得了良好的性能和泛化能力。

参考文献（格式）

[1] 杜圣杰,贾晓芬,黄友税.面向 CNN 模型图像分类任务的高效激活函数设计 U.红外与激光工程,2021,12(8):11-12.

[2] 彭斌，白静，李文静，郑虎，马向宇. 面向图像分类的视觉 Transformer 研究进展. 计算机科学与探索 2024,18(02),320-344

[3] 昝楠楠（1984-）. 基于全局 CNN 与局部 LSTM 的国画图像分类算法. DOI：10.20033/j.1003-7241.(2024)04-0115-03

[4] 李伟,孙云娟. 基于深度学习的 CNN 手写体数字识别. 洛阳理工学院学报(自然科学版) 2024,34(01),56-60+66

[5] Yann LeCun, Leon Bottou, Yoshua Bengio, and Partrick Haffner. Gradient-Based Learning Applied to Document Recognition