基于深度学习的 MNIST 手写数字识别系统

22121630 汪江豪

摘　要：本项目基于 Pytorch 框架，主要使用 CNN 卷积神经网络，构建了一个完整的 MNIST 手写数字识别系统，包含数据收集、模型训练、模型测试、特征分析、模型分析等内容和自定义数据集识别等功能。旨在完成手写数字识别系统的构建，同时分析深度学习过程中模型的性能差异，以指导后续其他基于深度学习的 CV 项目构建。

关键词：Pytorch, MNIST, CNN, 模型训练

1. 项目背景和意义

1.1 项目背景

手写数字识别是计算机视觉领域的经典问题，其目标是将手写的数字图像自动分类为对应的数字类别（0-9）。该问题的研究始于 20 世纪 60 年代，随着深度学习技术的发展，基于神经网络的解决方案已经成为主流。

MNIST 数据集是手写数字识别领域最常用的基准数据集之一，由美国国家标准与技术研究院(NIST)提供，包含了 60000 张训练图像和 10000 张测试图像。每张图像是 28*28 像素的灰度图，代表手写的单个数字。由于其数据规模适中、标注清晰、任务简单，MNIST 成为验证机器学习算法性能的“Hello World”级标准。

1.2 项目意义

在 CV 领域，本项目旨在通过对比 CNN 算法在 MNIST 数据集上不同数据规模下的性能与效果分析，探索深度学习在计算机视觉领域中应用的巨大潜力。

该项目研究成果可推广到更复杂的图像分类、目标检测等任务中，亦可为后续研究（如手写文字识别、文档分析）提供技术积累与参考。

作为深度学习与计算机视觉两个领域的入门项目，本项目为初学者提供了完整的实验范本（数据分析、模型训练、测试分析），也为学术研究提供了基准实验结果，支持后续改进算法的对比分析。[4]

2. 实验原理

2.1 CNN 核心思想

卷积神经网络(Convolutional Neural Network, CNN)是一种专门用于处理具有网格结构数据的深度学习模型。其核心思想是通过局部感受野和参数共享，高效提取数据的局部特征。CNN 的关键组件包括：

- 卷积层(Convolutional Layer)：提取局部特征（如边缘、纹理）。[1]
- 池化层(Pooling Layer)：下采样，压缩特征图尺寸，增强鲁棒性。

• 全连接层(Fully Connected Layer)：将高维个证映射到分类空间。

• 激活函数(Activation Function)：引入非线性（如 Relu）。

卷积是 CNN 中核心操作，数学原理公式为：

$$O(i,j) = \sum_{c=1}^{C} \sum_{m} \sum_{n} I_c(i+m, j+n) \cdot K_c(m,n) + b$$

其中，$O(i,j)$是输出特征图的像素值，$I_c(i+m, j+n)$是输入特征图的局部区域，$K_c(m,n)$是卷积核的权重，b是偏置项。[2]通过该函数，能较好地提取数据特征。[3]

2.2 损失函数

损失函数(Loss Function)是深度学习模型训练的核心租价，用于衡量模型预测值与真实值之间的差异。其核心作用是量化误差、指导优化。

损失函数通常用交叉熵表示，本项目的最终分类标签是 0~9，属于多分类问题，对于分类问题，公式如下：

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

其中，$\hat{y}_i$是预测概率，y_i是真实标签编码，通过该函数，能较好地衡量模型效果。[4]

2.3 CNN 在 MNIST 中的优势

相比 MLP 多层感知机，CNN 无需手动展平图像，直接利用空间结构信息，准确率更高。且可通过中间层激活图观察模型如何提取数字特征。

3. 系统实现

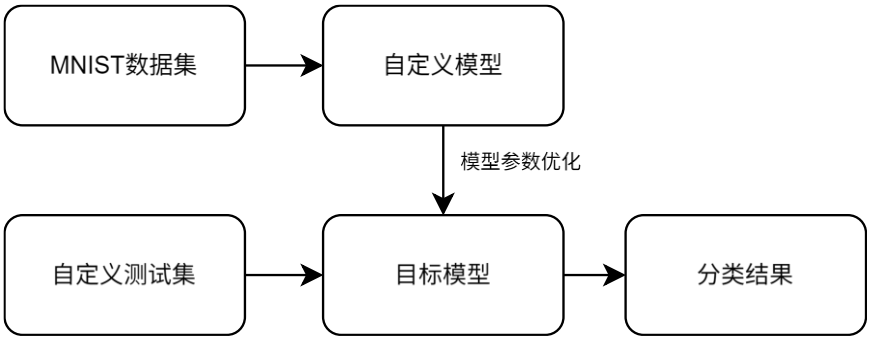

图 1 系统实现流程图

如上图，展示了整个手写数字识别系统的实现逻辑，通过 MNIST 数据集构建自定义模型，然后通过迭代，优化模型参数，构建理想目标模型。最后通过自定义手写数字测试集，测试系统效果。

4. 核心步骤

4.1 模型定义与训练

通过继承 torch.nn 中基类模型 Module 来构建简单的 CNN 神经网络模型，并自定义前向传播方法。

```
CUDA是否可用： True
CUDA设备数量： 1
当前CUDA设备索引： 0
当前CUDA设备名称： NVIDIA GeForce RTX 3060 Laptop GPU
CUDA版本： 12.1
cuDNN版本： 90100
```

图 2 设备选择

模型训练设备选择 NVIDIA GeFore RTX 3060 而不是 CPU，可显著提升卷积计算速度。模型训练设置 3 个轮次，通过迭代方式提升模型效果。可以多次训练，保存最佳效果下的模型数据字典。[6]

4.2 数据规模分析

为了比较不同数据规模、数据轮次训练带来的对实验准确率的影响，我分析绘制了综合的训练过程分析图，旨在揭示深度学习模型对 CV 项目的训练效果。[6]

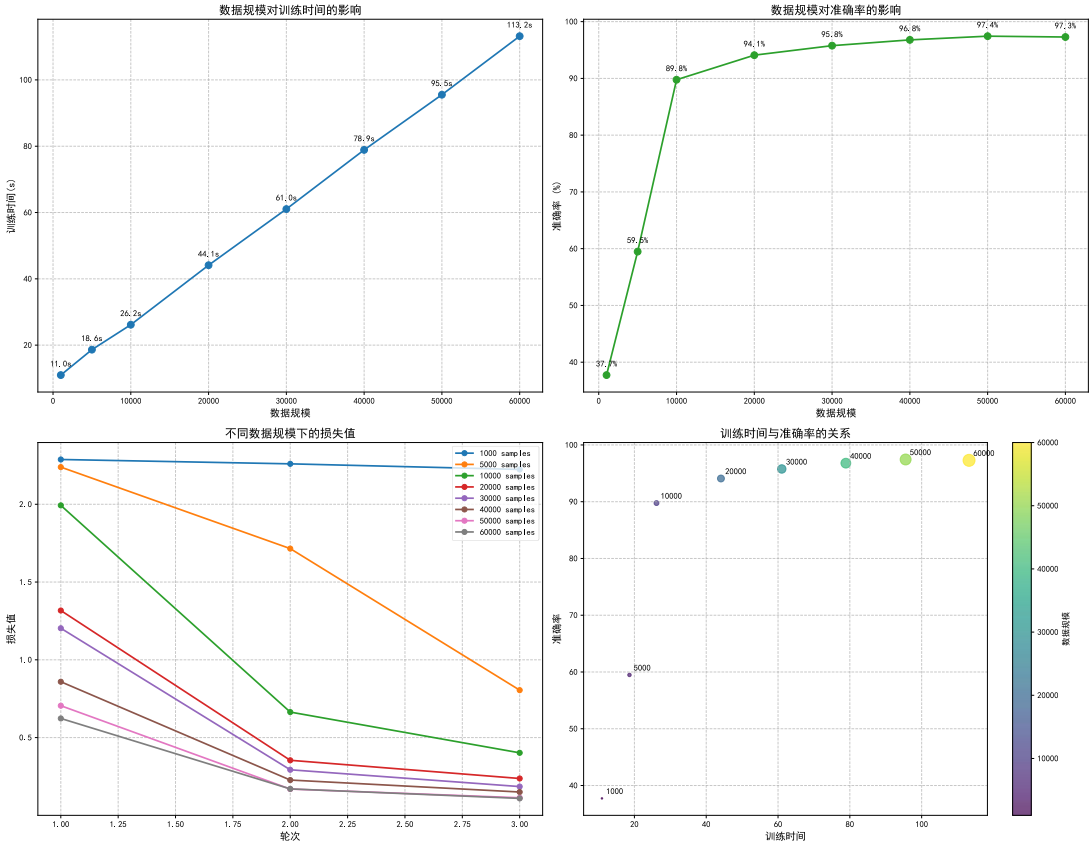

图 3 数据规模对训练的影响

如上图，由于 MNIST 数据集公有近 60000 张图片，因此我将数据分成了不同规模大小，分别为 5000，10000，20000，30000，40000，50000，60000，以全面分析不同数据规模对实验结果的影响。

从左上图，数据规模对训练时间的影响，可以看出，对于 CNN 模型，训练时间随着数据规模基本是线性增加，在硬件条件允许的情况下，可适当增加数据规模，时间不会显著增加。

从右上图，数据规模对准确率的影响中，可以看出当数据规模小于等于 10000 时，在提升数据量时能够显著增加模型识别的准确率，但当数据规模超过 10000 后，继续增大数据量，模型识别的准确率不再明显上升，这表明，我们为了追求训练效率和效果时，最好将数据规模设置在 10000 以上。[7]

左下图，表明了在不同轮次中，不同的数据规模下损失值的变化情况。可见，数据规模越大，损失值越小，训练效果越佳，权衡考虑，建议将数据规模设置 10000 以上。

右下图，反映了训练时间对模型准确率的影响，为了获得较好的模型训练效果，数据规模应设置 20000 以上，保留 40 秒以上的训练时间。

4.3 训练轮次分析

为了探究训练轮次对模型效果影响，我通过迭代不同训练轮次，计算了对应的模型准确率。结果如下图：

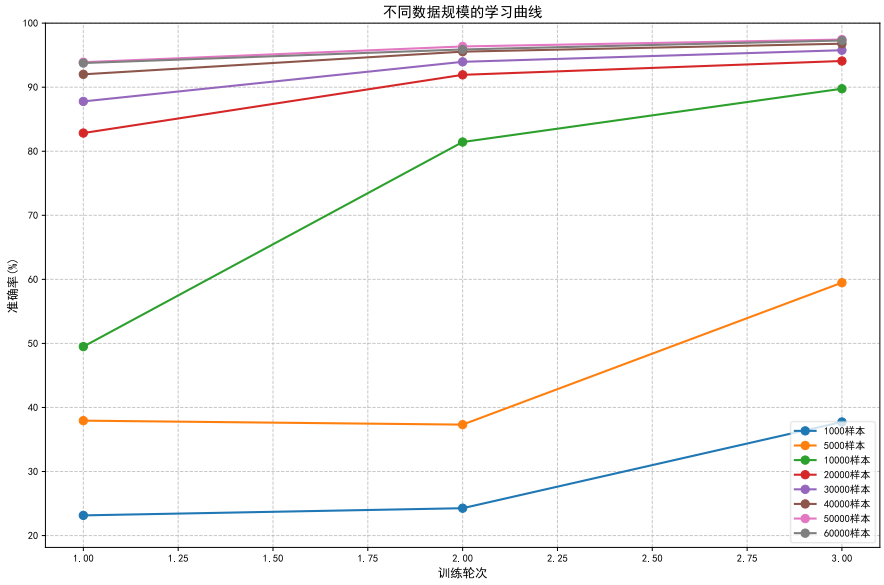

图 4 不同轮次和数据规模的学习曲线

可见，为了保证质量，轮次至少 2 轮以上，数据规模保持在 20000 以上。

5. 效果展示

图 5 自建测试数据

通过一个 demo 程序构建了自定义测试集，用户可自行输入数字，进行识别。

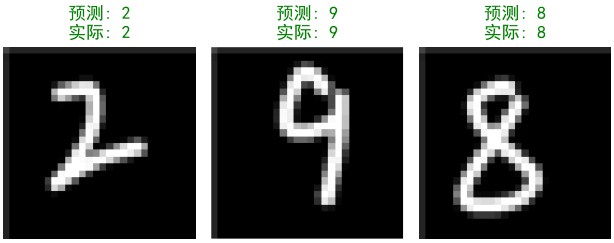

图 6 识别结果

如上图，结果识别正确，但若手写体连笔过多，可能错误率较高，后续可通过数据增强，优化模型参数以提高系统鲁棒性。

6. 总结

MNIST 手写数字识别是计算机视觉领域的经典项目，此次通过结合深度学习 CNN 算法，让我深入了解了 CV 项目的丰富性和深度学习算法的强大之处，当下 AI 火热，我们应多接触前沿 AI 知识，并横向运用于 CV 等其他领域，以提升自己的计算机学科素养能力。

参考文献

[1] Shannon C E. *A Mathematical Theory of Communication*. Bell System Technical Journal, 1948, 27(3): 379-423.

[2] LeCun Y, Bottou L, Bengio Y, Haffner P. *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[3] 陈云霁，智能计算系统——从深度学习到大模型，机械工业出版社, 2024: 49-60

[4] Goodfellow I, Bengio Y, Courville A. Deep Learning[M]. MIT Press, 2016: Chapter 9.

[5] Chollet F. Deep Learning with Python[M]. Manning Publications Co., 2017: Chapter 3.

[6] 罗婧, 叶志晟, 杨泽华, 傅天豪, 魏雄, 汪小林, 罗英伟, . 研发类 GPU 集群任务数据集的构建及分析[J]. 计算机工程与科学, 2024, 46(12): 2128-2137

[7] Bishop C M. *Pattern Recognition and Machine Learning*. Springer, 2006: 209-210.