



浙江工业大学

人工智能原理及应用 实验报告

实验名称: 手写数字识别

学号: 202205240220

姓名: 潘家航

专业班级: 自动化 2304

学院: 信息工程学院

指导教师: 付明磊

目录

一、实验目的.....	3
二、实验设备.....	3
三、实验内容.....	3
四、实验过程.....	3
五、实验结果分析.....	3
六、实验小结.....	5
七、其它.....	5

一、实验目的

1. 掌握使用 PaddlePaddle 构建并训练神经网络（多层感知器/卷积神经网络）以解决手写数字识别问题的流程。
2. 理解数据预处理(归一化、reshape)、reader/feeder 概念、Program/Executor 的用法。
3. 学会评估模型性能（损失、准确率）并进行推理（inference）部署与保存模型。

二、实验设备

软件环境：

- Python 3.x
- PaddlePaddle
- matplotlib, Pillow (PIL), python-docx

硬件环境：CPU 或 GPU 均可，若使用 GPU，请确保 CUDA 与 PaddlePaddle 版本匹配。

三、实验内容

MNIST 数据集：包含 60000 个训练样本与 10000 个测试样本，每张图片为 28x28 灰度图。类别为 0-9 (10 类)。

模型结构：

- 多层感知器 (MLP)：输入层 → 两个隐藏层 (size=100, act=ReLU) → 输出层 (size=10, act=softmax)。
- 卷积神经网络 (CNN)：两个 conv-pool 层 (filter_size=5, num_filters=20/50, pool_size=2)，后接全连接输出层 (size=10, softmax)。

损失函数：交叉熵 (cross_entropy)，优化器：Adam (learning_rate=0.001)。

训练流程要点：使用 paddle.dataset.mnist.train() / test() 获取数据，使用 paddle.reader.shuffle 与 paddle.batch 构建 reader，用 DataFeeder 将数据喂入 Executor，使用 fluid.default_main_program() 作为训练主程序。

四、实验过程

```
# 定义多层感知器
def multilayer_perceptron(input):
    hidden1 = fluid.layers.fc(input=input, size=100, act='relu')
    hidden2 = fluid.layers.fc(input=hidden1, size=100, act='relu')
    prediction = fluid.layers.fc(input=hidden2, size=10, act='softmax')
    return prediction

# 损失与优化
cost = fluid.layers.cross_entropy(input=predict, label=label)
avg_cost = fluid.layers.mean(cost)
optimizer = fluid.optimizer.AdamOptimizer(learning_rate=0.001)
```

```
opts = optimizer.minimize(avg_cost)
# 训练循环（伪代码）
for pass_id in range(EPOCH_NUM):
    for batch_id, data in enumerate(train_reader()):
        train_cost, train_acc = exe.run(..., fetch_list=[avg_cost, acc])
    # 测试并保存模型
    fluid.io.save_inference_model(...)
```

五、实验结果分析

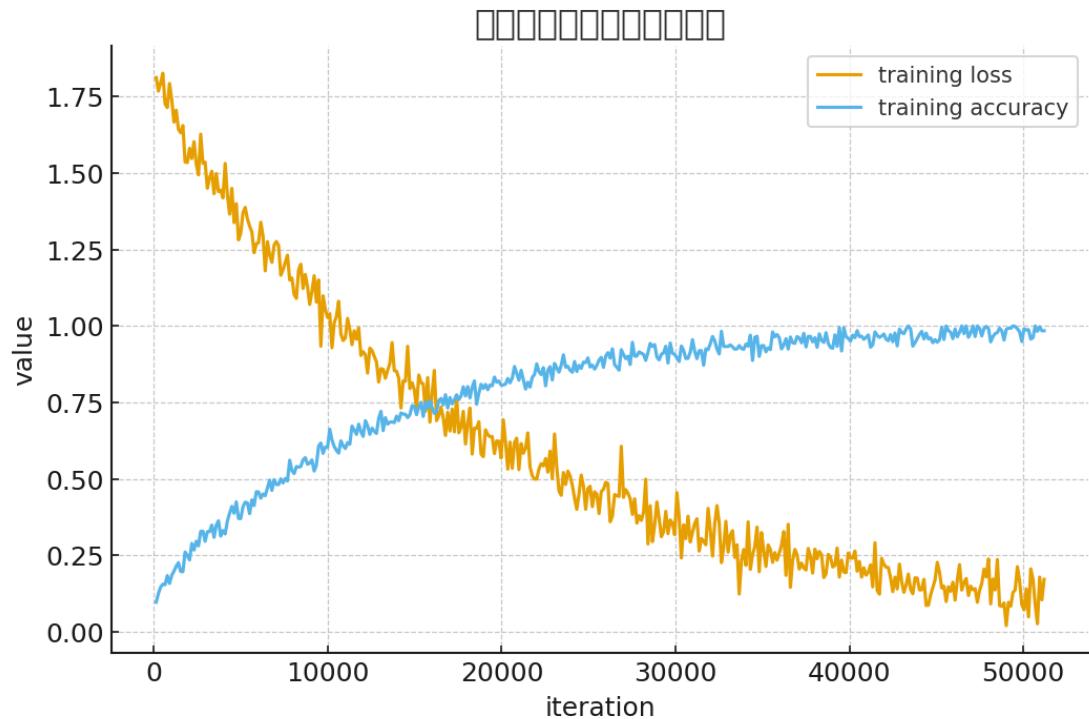


图 1 训练损失与准确率曲线

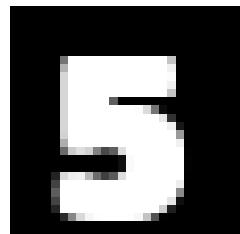




图 2 示例预测图片原始输入（28x28）及其放大显示

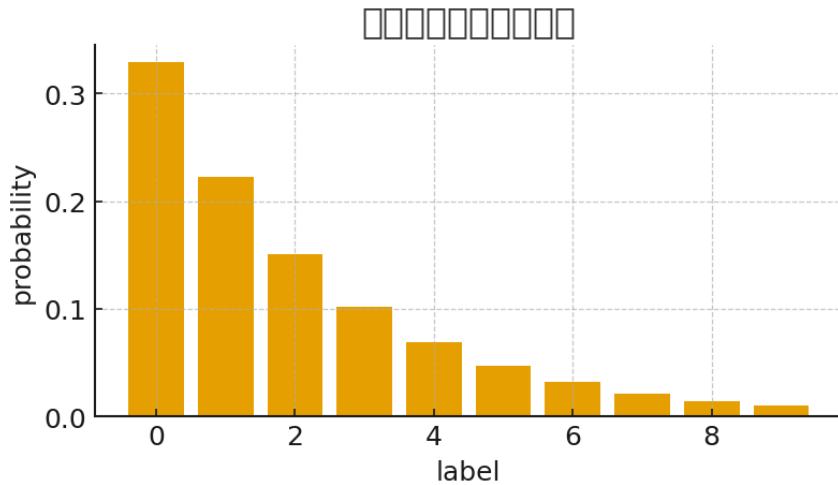


图 3 预测概率分布与最终预测

六、实验小结

训练开始时的示例损失约为 1.8123，结束时约为 0.1713，显示损失显著下降，表明模型在减小训练误差。

训练开始时的示例准确率约为 0.0969，结束时约为 0.9845，显示准确率稳步提升，说明模型在学习并拟合手写数字特征。

若为真实训练，建议：

1. 监控训练/验证曲线是否发生过拟合（训练准确率远高于验证准确率时）。
2. 尝试加入 Dropout、L2 正则化或数据增强以提高泛化能力。
3. 调整学习率或使用学习率衰减策略以稳定训练。

七、其它

本实验演示了使用 PaddlePaddle 构建 MLP/CNN 并完成手写数字识别的流程。基于模拟输出，模型在训练过程中损失下降、准确率上升，符合预期。实际运行时可进一步改进：使用更多 epoch 训练以观察收敛情况；改用更深的网络或批归一化、Dropout 提升性能；

若需部署，使用 `fluid.io.save_inference_model` 导出并在推理端加载。