《大数据分析方法》课程实验报告

学号：2020204246

姓名：王浩

专业：计算机科学与技术

班级：20 图灵

实验六 卷积神经网络 CNN 框架的实现与应用

实验目的：

1. 掌握卷积神经网络 CNN 的基本原理

2. 利用 CNN 实现手写数字识别

实验内容：

1. 数据集：

MNIST 数据集，6000 张训练图像，10000 张测试图像，每张图像 size 为 28*28

2. 利用 LeNet-5 CNN 框架，实现手写数字识别。

3. 利用更先进的 CNN 网络模型（自选），实现手写数字识别

实验原理：

利用 LeNet-5 CNN 框架，实现手写数字识别。

1. 网络层级结构概述如图 1：7 层神经网络

Input layer: 输入数据为原始训练图像

Conv1：6 个 5*5 的卷积核，步长 Stride 为 1

Pooling1：卷积核 size 为 2*2，步长 Stride 为 2

Conv2：12 个 5*5 的卷积核，步长 Stride 为 1

Pooling2：卷积核 size 为 2*2，步长 Stride 为 2

Output layer：输出为 10 维向量

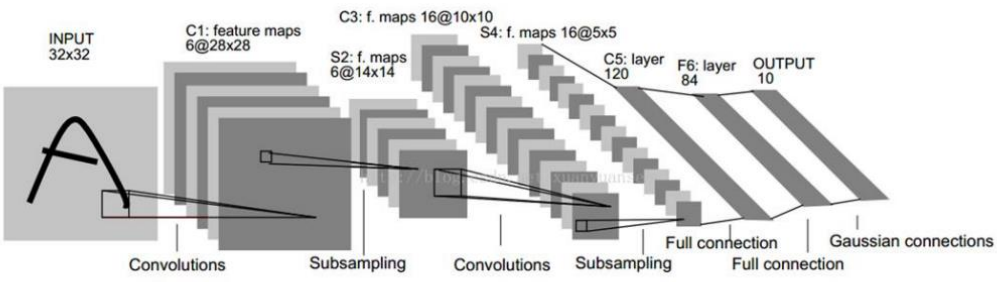

图 1：CNN 模型基本框架图

实验基本流程：

（1）获取训练数据和测试数据；

（2）定义网络层级结构；

（3）初始设置网络参数（权重 W，偏向 b）cnnsetup(cnn, train_x, train_y)

（4）训练超参数 opts 定义（学习率，batchsize，epoch）

（5）网络训练之前向运算 cnnff(net, batch_x)

（6）网络训练之反向传播 cnnbp(net, batch_y)

（7）网络训练之参数更新 cnnapplygrads(net, opts)

（8）重复（5）（6）（7），直至满足 epoch

（9）网络测试 cnntest(cnn, test_x, test_y)

实验要求：

1. 利用交叉验证方法，分析识别结果
2. 分析网络参数 opts 设置对最终识别结果的影响，画出相应的结果分析图

一．问题描述

实验流程，根据 lenet5 网络结构，构建神经网络模型，采用交叉熵计算损失，通过 Adam 算法进行优化。

开始训练模型，计算每 epoch 的损失和准确率。

对测试集进行预测，并对预测结果进行可视化

二.　解决思路

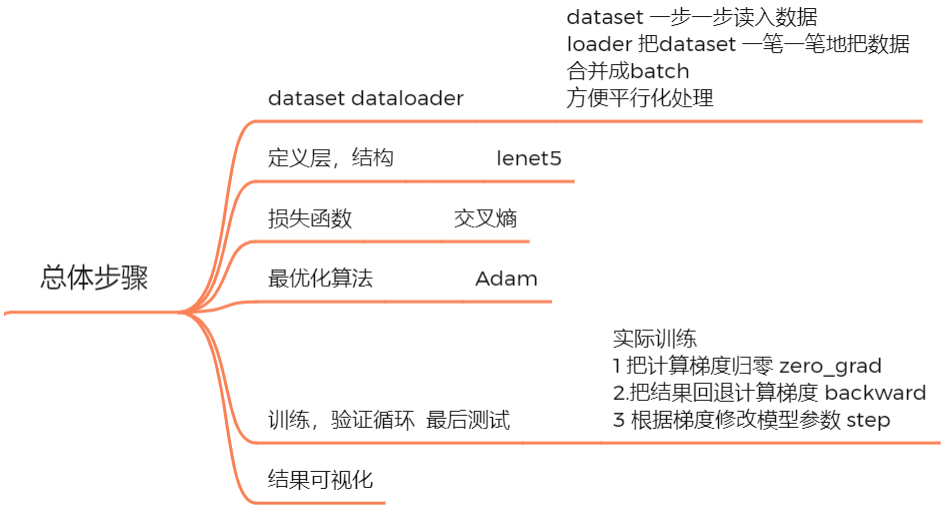

图 1 问题解决流程图

2.1　数据集处理

2.1.1　生成数据

采用datasets中的MNIST下载并导入数据，同时进行可视化

```
train_dataset = datasets.MNIST(root='mnist_data',
                               train=True,
                               transform=transforms,
                               download=True)
```

```
for index in range(1, 10):
    plt.subplot(3, 3, index)
    plt.axis('off')
    plt.imshow(valid_dataset.data[index], cmap='gray_r')
```

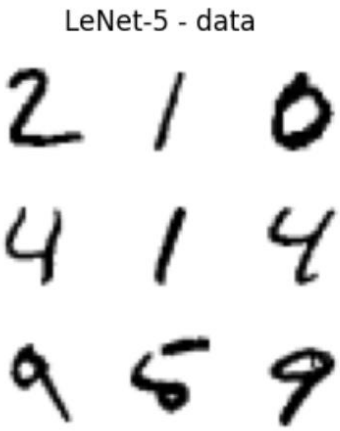

图 2 提取数据图

2.1.2 定义模型结构

在LenNet5类中定义两个Sequential容器将神经网络分为卷积池化层和全连接层，即提取特征和分类两个功能。

```
self.feature_extractor = nn.Sequential(
    nn.Conv2d(1, 6, 5),
    nn.Tanh(),
    nn.AvgPool2d(kernel_size=2),
    nn.Conv2d(6, 16, 5),
    nn.Tanh(),
    nn.AvgPool2d(kernel_size=2),
    nn.Conv2d(16, 120, 5),
    nn.Tanh()
)

# two affine operations: y = Wx + b
self.classifier = nn.Sequential(
    nn.Linear(in_features=120, out_features=84),
    nn.Tanh(),
    nn.Linear(in_features=84, out_features=n_classes),
)
```

图 3 模型结构图

2.1.2 定义损失函数和优化器

采用交叉熵计算损失，通过 Adam 算法进行优化。

```
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE)
criterion = nn.CrossEntropyLoss()
```

图 4 损失函数和优化器图

2.1.3 train_loop

将训练过程封装为两部分

一个是训练集训练，记录每epoch的loss并迭代更新相应参数

一个是测试集训练，记录每epoch的loss。

输出每epoch的训练集，测试集误差以及准确率。

```
16:44:06 --- Epoch: 0   Train loss: 0.0569      Valid loss: 0.0607      Train accuracy: 98.64   Valid accuracy:
98.09
16:44:49 --- Epoch: 1   Train loss: 0.0440      Valid loss: 0.0513      Train accuracy: 99.05   Valid accuracy:
98.40
16:45:34 --- Epoch: 2   Train loss: 0.0370      Valid loss: 0.0518      Train accuracy: 99.15   Valid accuracy:
98.54
16:46:18 --- Epoch: 3   Train loss: 0.0299      Valid loss: 0.0487      Train accuracy: 99.28   Valid accuracy:
98.60
16:47:01 --- Epoch: 4   Train loss: 0.0257      Valid loss: 0.0498      Train accuracy: 99.32   Valid accuracy:
98.55
16:47:43 --- Epoch: 5   Train loss: 0.0227      Valid loss: 0.0459      Train accuracy: 99.44   Valid accuracy:
98.58
16:48:25 --- Epoch: 6   Train loss: 0.0198      Valid loss: 0.0436      Train accuracy: 99.44   Valid accuracy:
98.82
16:49:07 --- Epoch: 7   Train loss: 0.0188      Valid loss: 0.0498      Train accuracy: 99.44   Valid accuracy:
98.60
```

图 5 实验中间结果

图 6 训练集和测试集误差随 epoch 变化图

三. 模型构建及结果分析

3.1 模型结果

从测试集中取出60个图片，用刚刚训练好的模型进行预测，输出模型结果，并可视化

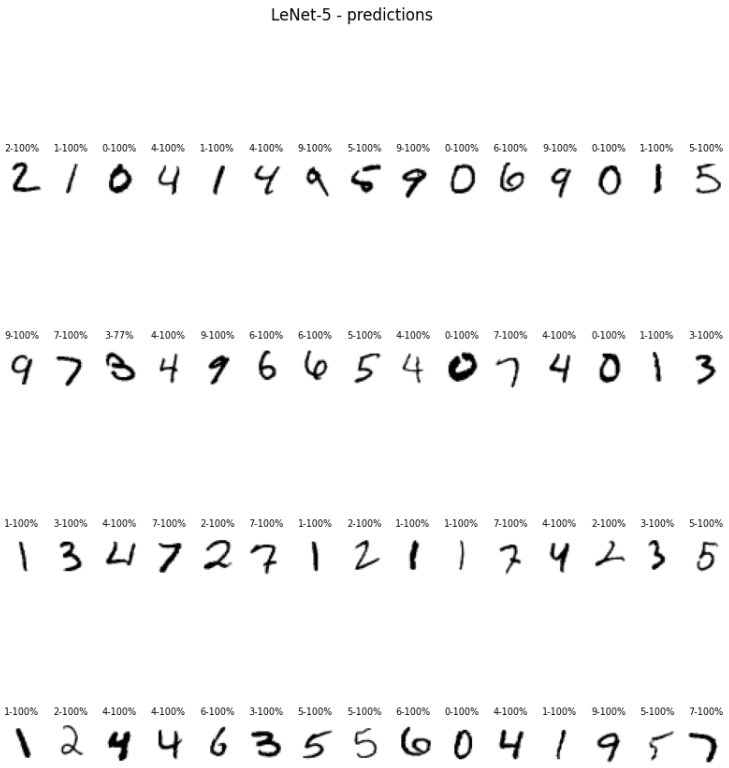

图 5 准确率-PCA 维数图

上图每个图片上方第一个数字是预测的结果，段横杠后面是预测的概率。

训练了20epoch后的模型误差和准确率

16:57:24 --- Epoch: 19 Train loss: 0.0088 Valid loss: 0.0520 Train accuracy: 99.76 Valid accuracy: 98.65

图 6 模型准确率和误差图

3.2 实验中存在的问题及解决方法

在实验过程中有很多pytorch的语法，并不太懂，通过上网查阅相关资料，查看官方文档，看相关视频等方式解决。比如，对模型保存和调用，相关函数的返回值等。