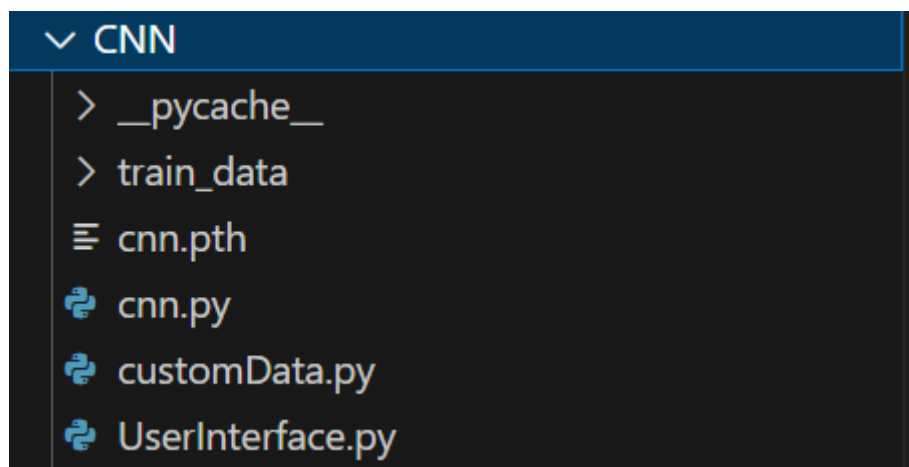


PJ1（CNN）说明文档

22302010019 陈星宇

一、整体结构

1.文件结构



train_data: 数据集，从中划分出训练集和测试集

cnn.pth: 训练好的网络数据存储在文件中，由用户选择是否调用

cnn.py: CNN网络搭建与控制

customData: 构建能够加载训练集和测试集数据的数据加载器，以便用于模型的训练和评估

UserInterface.py: 控制台交互接口，用于读写训练数据时获取用户选择

2.网络结构

整个网络结构包括了两个卷积层、两个 Dropout 层、两个 ReLU 激活函数层和一个全连接输出层。这个网络结构适用于处理输入为 $1 \times 28 \times 28$ 大小的灰度图像，并输出 12 个汉字手写体类别的分类结果。详细说明如下：

输入层：输入图像是单通道的灰度图像，尺寸为 $1 \times 28 \times 28$ ，

第一层卷积：

- 卷积层：输入通道数为 1，输出通道数为 16，卷积核大小为 5×5 ，步长为 1，补零圈数为 2（padding=2）。
- ReLU 激活函数：用于增加网络的非线性性。
- 最大池化层：池化核大小为 2×2 ，步长为 2，对特征图进行降采样，输出尺寸为 $16 \times 14 \times 14$

第二层卷积：

- 卷积层：输入通道数为 16，输出通道数为 32，卷积核大小仍为 5×5 ，步长为 1，补零圈数为 2。
- ReLU 激活函数：增加网络的非线性性。
- 最大池化层：池化核大小为 2×2 ，步长为 2，对特征图进行降采样，输出尺寸为 $32 \times 7 \times 7$ 。

输出层：全连接层，输入特征数为 $32 \times 7 \times 7$ ，输出类别数为 12

二、运行环境

操作系统：Windows 11

Python版本：3.12.2

IDE：VSCode 1.86

依赖项：numpy 1.26.4

pytorch 2.2.2

三、运行结果

使用提交代码中所给的超参数和网络结构进行训练，训练结果如下：

EPOCH	LOSS	ACCURACY
1	0.288579	92.72727%
3	0.103894	96.78030%
5	0.074758	97.84091%
40	0.011287	99.01228%

EPOCH	LOSS	ACCURACY
90	0.005456	99.88636%

继续训练，进步已经不太显著。

四、对防止过拟合所做的改进

使用在全连接层之间放置Dropout来避免过拟合：

```
def forward(self, x):          #x [batch,1,28,28]
    x = self.conv1(x)
    x = self.dropout(x)
    x = self.conv2(x)          # x变成四维张量[batch, 32,7,7]
    x = x.view(x.size(0), -1)  # 将输入特征向量展平  经过线性变换得到
输出
    output = self.output(x)
    return output
```

dropout定义如下：

```
self.dropout = nn.Dropout(dropout_prob) #防止过拟合
```

dropout_prob定为0.2。也即，x中的值有20%的概率被置为0，其余值*1.25（即除以1-0.2）。

理解：在正常的神经网络中，神经元之间会形成强烈的依赖关系，导致某些神经元在训练集上表现得非常好，但在测试集上表现不佳（即过拟合）。Dropout 会随机关闭一些神经元，这样会导致网络的参数在不同的训练迭代中变化，从而减少参数之间的相互依赖。这有助于网络学习更加泛化的特征，而不是过度拟合训练集的特定特征。

实验也证明了放置Dropout来避免过拟合方法的有效性：取每种类型的前550个组成样本集，后70个组成测试集(测试集大小为12*70=840)。不使用Dropout，训练6个epoch，得到的测试集上的正确率为98.45238%，即826/840；使用Dropout后，训练6个epoch，得到的测试集上的正确率为99.76190%，即838/840。可见Dropout大大提升了模型的泛化能力。

五、为防止震荡所做的改进

引入了容忍度`patience`，若有某个`epoch`的测试正确率较之于此前的最大正确率并无提升，`drop`计数器自增1，当`drop>patience`时（或正确率已达预期的理想正确率），中断训练过程，并记录下最佳结果的出现位置，以便后续复现。

```
patience=5
.....
accuracy=test(cnn,test_loader,loss_function)
if(accuracy>maxAccuracy):
    maxAccuracy=accuracy
    maxEpoch=epoch
else :
    drop+=1
    if(drop>patience or maxAccuracy>0.9996):
        print('at epoch ',maxEpoch,' max accuracy:
',maxAccuracy)
        break
```

实验证明，当训练次数足够多时，由于学习率不是动态调整的，这种方法可以在一定程度上减少震荡，确保训练结果不倒退。当训练次数为150时，训练在87次时中断，反馈信息表示最佳结果在81个`epoch`处出现。

六、超参数的选择

（1）批大小

在自定义的`DataLoader`内部，`batch_size=24`，即以24张图片为一组。在`cnn`网络搭建时，又以20个`DataLoader`为一组，进行一次参数更新。实验中，`batch_size`改小会有更佳的训练效果，但是由于本网络收敛速度较快，仍使用了较大的`batch_size`以确保模型更新的稳定性。

（2）学习率

取不同的学习率，训练6个`epoch`，控制其他超参数相同，结果如下：

LEARNING_RATE	ACCURACY
0.1	8.33333%（梯度爆炸）
0.01	97.12121%

LEARNING_RATE	ACCURACY
0.001	97.99242%
0.0001	90.56818%

最后，选择学习率为0.001进行学习。

七、总结与反思

相较于反向传播算法，CNN 的核心是卷积层和池化层。卷积层可以提取图像的局部特征，而池化层则可以减小特征图的尺寸，降低计算复杂度并提取更加抽象的特征。因此，需要合理设置卷积核的大小、步长、填充方式以及池化操作的类型和大小，以便有效地提取图像的特征。BP神经网络对输入数据的特征表示没有明确的局部感知性，而CNN通过卷积操作可以捕获局部区域的特征，在处理具有空间相关性的数据时具有优势。

本PJ在以下方面可能可以做出改进：

- 1.探究不同的卷积核大小、卷积层数对网络训练效果的影响。
- 2.尝试如 L2 正则化的方法，限制模型的复杂度，减少过拟合风险。