

智能系统 Lab1-2 实验文档

本项目是一个基于 pytorch 框架的卷积神经网络，实现了对 12 个手写汉字的多分类任务。

代码架构

项目代码由主要测试代码和一个自定义 CNN 类 Net 组成。

main

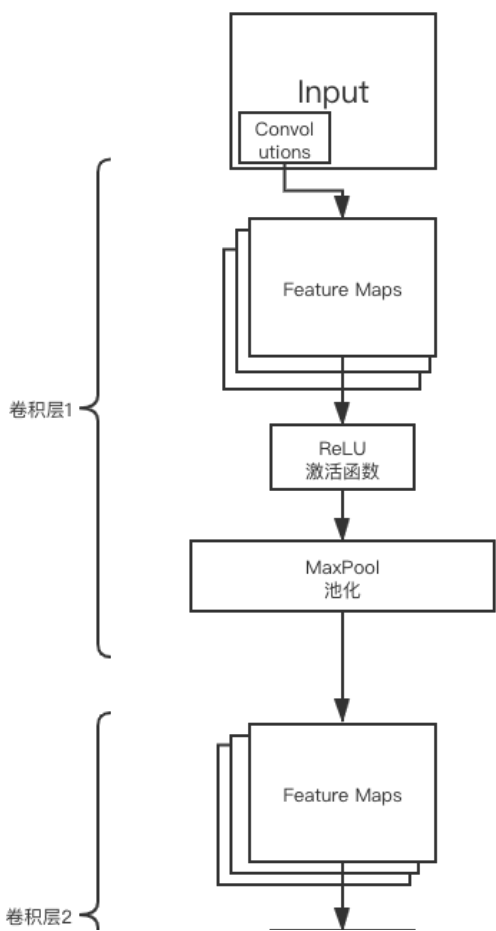
- 生成训练、测试集：基于 torchvision 库，加入了 batch_size 参数
- 训练网络：基于 Adam 优化算法

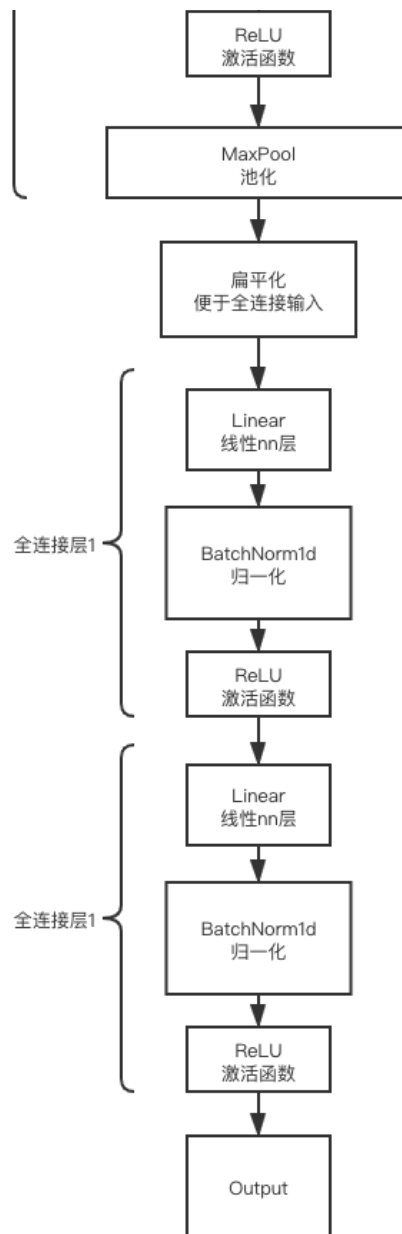
Adam 是一种可以替代传统随机梯度下降过程的一节优化算法，它能基于训练数据动态调整每个参数的学习率，每一次迭代学习率都有一个确定范围，使得参数比较平稳。

- 测试并计算正确率

Net

本项目的 CNN 网络整体由两层卷积层、两层全连接层组成，具体结构如下：

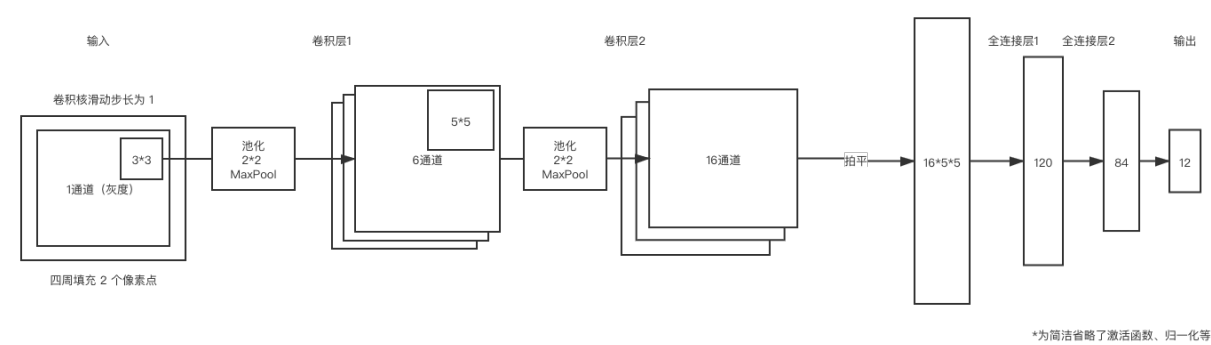




多分类任务：对 12 个手写汉字分类

- 生成训练、测试集：利用 torchvision 库，对图像进行以下处理：
 - 居中裁切 28*28 的图像
 - 随机旋转图像，防止对一种方向上的过拟合
 - 调整为灰阶，保证图像只有一个通道
 - 将类型转为 torch.FloatTensor
- 装载训练、测试集：加入 batch_size 参数，使训练震荡减小；打乱顺序，防止过拟合。
- 训练、测试集选取方法与 Lab1-1 相同，即每个汉字前 600 个图像作为训练集，后 20 个图像作为测试集。
- ReLU 激活函数： $y = x (x > 0)$ 当 $x > 0$ 时，ReLU 导数为 1，当 $x \leq 0$ 时，ReLU 导数为 0，导数为 1 可以保证激活函数不对链式求导产生缩放影响，避免梯度消失。同时提高训练性能。
- 最大池化：提取重要特征。

网络结构参数：



对训练次数的测试

在参数如上，batch_size=60 的情况下对训练次数的效果进行测试。由于存在随机性，每种情况测试 3 次取平均值。

次数	准确率1	准确率2	准确率3	平均准确率
1	85.83%	86.67%	93.75%	88.75%
3	94.58%	94.17%	87.08%	91.94%
5	97.08%	96.25%	96.67%	96.67%
7	96.67%	97.08%	98.75%	97.50%
9（可能过拟合）	95.42%	96.25%	96.67%	96.11%
8	90.42%	95.00%	96.25%	93.89%
6	97.50%	92.08%	95.83%	95.14%

训练 7 次的情况下训练效果最好，虽然整体差距不大。注意到每种情况下的准确率震荡都比较厉害，尝试对不同的 batch_size 比较测试。

对 batch_size 的测试

考虑到初始的 batch_size=60 对于容量为每字 600 的训练集可能过大，导致准确率震荡比较厉害。尝试固定次数为 7，减小 batch_size 训练，减小震荡。

batch_size	准确率1	准确率2	准确率3	平均准确率
6	97.92%	97.92%	97.08%	97.64%
3	96.25%	94.16%	95.41%	95.27%
10	97.08%	97.08%	97.91%	97.35%
8	97.08%	98.75%	94.17%	96.67%
7	97.92%	97.08%	96.25%	97.08%
20	77.08%	97.50%	95.00%	89.86%
30	91.25%	96.67%	97.50%	95.14%

总体而言，batch_size 在 10 以下的范围内，准确率震荡差别不大。batch_size 越大，虽然每次训练的 loss 震荡减小了，但最终输出的准确率却更容易产生波动。

batch_size 过大时，参数容易被调整到一个错误的方向而来不及修正；batch_size 过小时，由于训练集容量可能较小，参数调整时的震荡太大，也会影响训练效果。

最终，在 batch_size 为 6 时效果最好。

小结

相比于 Lab1-1 不依赖框架的 BP 网络，本次实验的 CNN 网络能更有效地对图像的局部特征建模（而不是简单地拍平），更加适合图像处理任务。

本项目也在训练效率上有了很明显的提升。考虑有以下几点原因：

1. 框架使用了性能更好的矩阵算法
2. CNN 框架在训练性能上整体优于 BP 网络
3. 增加了 batch_size 参数，减少了实际求导、调整参数的次数

以下是本项目可以进一步改进的点：

1. 可以尝试更多的调优方法，例如针对不同的网络结构比较测试
2. 可以进一步尝试提高准确率的稳定性