

CNN-mnist实验报告

概述

CNN使用了卷积结构，更充分的利用了图像之中的位置信息，卷积操作使得神经网络对图像的形状更加敏感，例如，在手写数字中，我们可以通过数“圈”识别一些数字，数字8有两个圈，数字0有一个圈，而其他数字没有圈，因此，“圈”这个形状特征可以帮助我们识别2个数字。CNN非常适合图像的分类任务。

本次实验报告有：

使用pytorch实现了CNN，并在mnist手写数字识别任务上获得较好效果

实现了dropout 正则化

实现了batch Normalization技巧

尝试了类似于GoogleNet那种跨层传输的结构，做了简单的实验

任务定义与输入输出

mnist的任务定义以及输入输出已在MLP实验中介绍，此处不再赘述。

方法描述

batch Normalization

batch Normalization是谷歌于2015年提出的一种非常有效的解决梯度消失问题的方法。

梯度消失问题一部分原因是由于激活函数的饱和性导致的，如softmax，在输入数字较大时，导数非常小，因此，当输入为-1~1时，激活函数才能获得较好的表现。因此，在训练过程中，我们可以试着把输入数字放缩到-1~1的范围内，就可以缓解梯度消失问题。

由于神经网络的目的是为了拟合数据的分布，只要我们在放缩过程中，不改变数据的分布，就不会影响神经网络的结果。

因此batch Normalization第一步先统计数据的分布，第二步把数据进行放缩。

残差学习

通常神经网络是要学习一个 f 使得 $y=f(x)$ ，存在的一个问题就是当 f 过于复杂，在神经网络层数增加，表达能力变强时，很容易过拟合，但是如果我们学习残差即 $y = g(x) + x$ 中的 $g(x)$ ，在深层神经网络中，表达能力没有下降，同时保留了 x 的很多信息，就缓解了过拟合问题。

dropout

dropout通过随机抛弃，即随机减少参与训练的参数量，缓解了过拟合问题。

结果分析

本次实现的cnn基本结构为

Conv 5*5,32

relu

maxpool 2

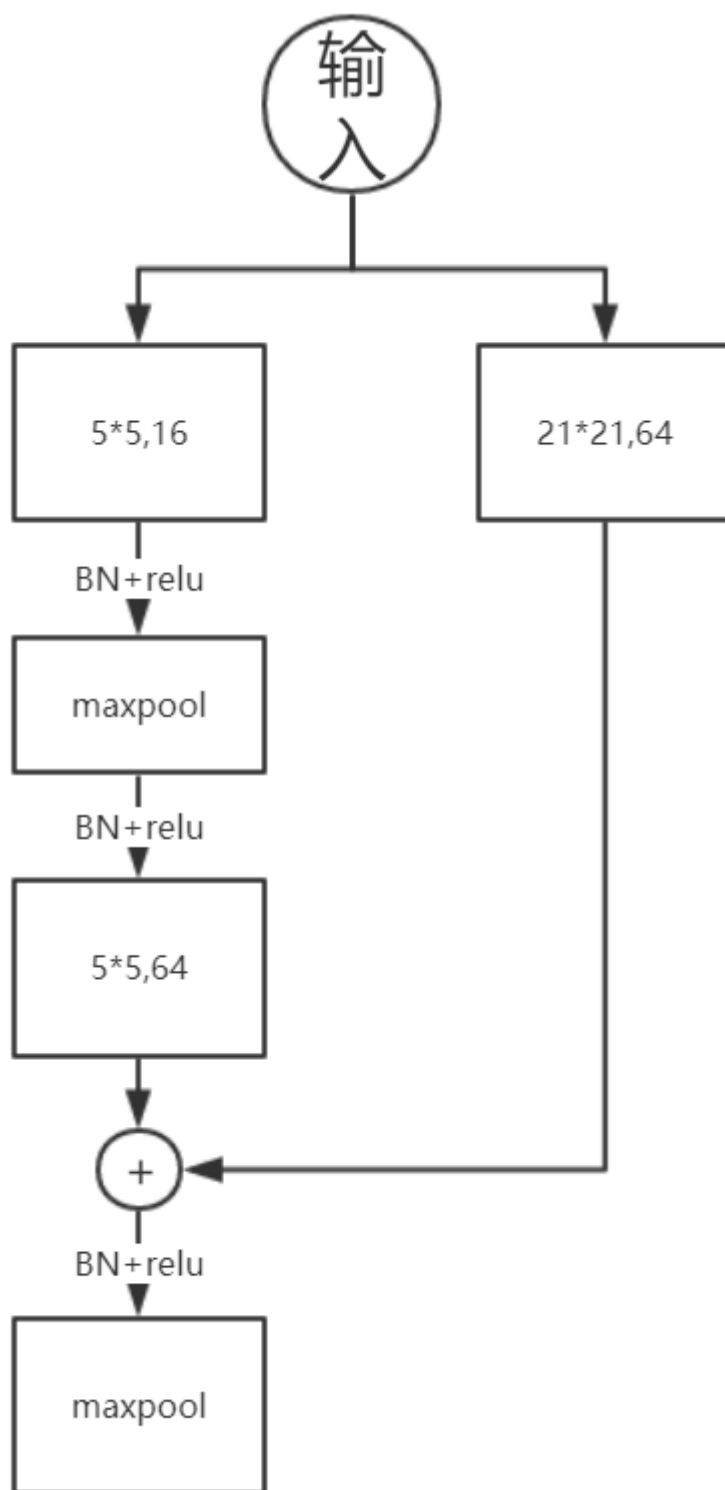
Conv 5*5, 64

relu

maxpool 2

Linear 1024 -> 10

实验说明	测试集准确率	实验说明
上述的cnn基本结构	98.66	第一个epoch结果为90
cnn+batch normal	99.17	第一个epoch结果为96
cnn+batch normal+dropout	98.94	
cnn特殊结构	98.93	结果如下图



发现batch Normalization非常有效，不仅提升了模型效果，还使得梯度下降的速度更快。

dropout和跨层传输的效果不够好，可能是因为本次实验的神经网络不深，过拟合现象不是特别严重，加入正则化技巧没有得到显著的效果

源码运行环境

运行源码需要安装

python

pytorch