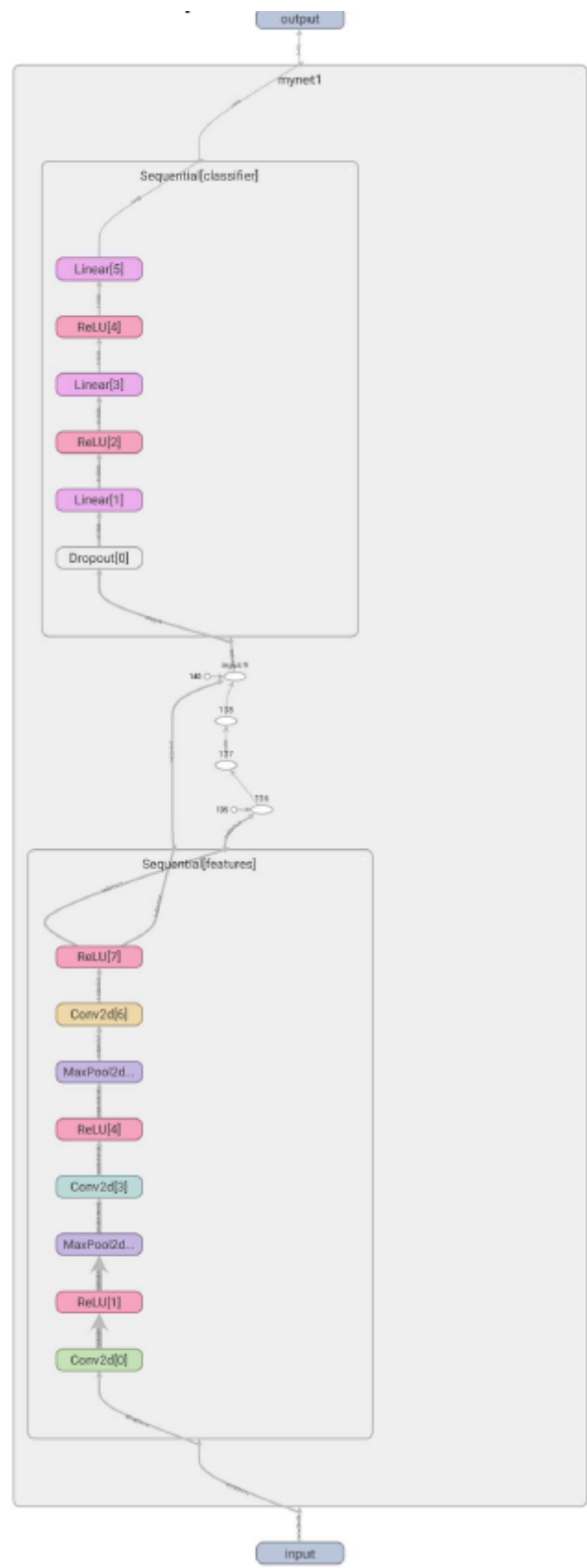


手写数字识别实验报告

1) CNN模型结构图及流程分析

我设计的CNN模型结构图为



输入先经过三组卷积层、池化层和激活层，提取特征，然后通过全连接层输出预测结果。

2.实验结果

提交26次后，得到最佳预测精度为0.99335(第13次提交)

3.不同参数对比及分析

主要参数为

```
'learning_rate': 0.01,
'batch_size': 64,
'max_epoch': 100,
'test_epoch': 5,
'momentum': 0.001,
'weight_decay':0.0001
```

下面对比几组参数，提交到kaggle网站上的预测结果使用的是用全部数据训练的模型。

batchsize	accuracy	validation accuracy
128	0.99314	0.99333
64	0.99335	0.99429
32	0.97514	0.99333

momentum	accuracy	validation accuracy
0.001	0.99335	0.99429
0.002		0.99381
0	0.99303	0.99286

max_epoch	accuracy	validation accuracy
110	0.99289	0.99405
100	0.99335	0.99429
90		0.99262
80	0.99310	0.99238
50	0.99275	0.99190

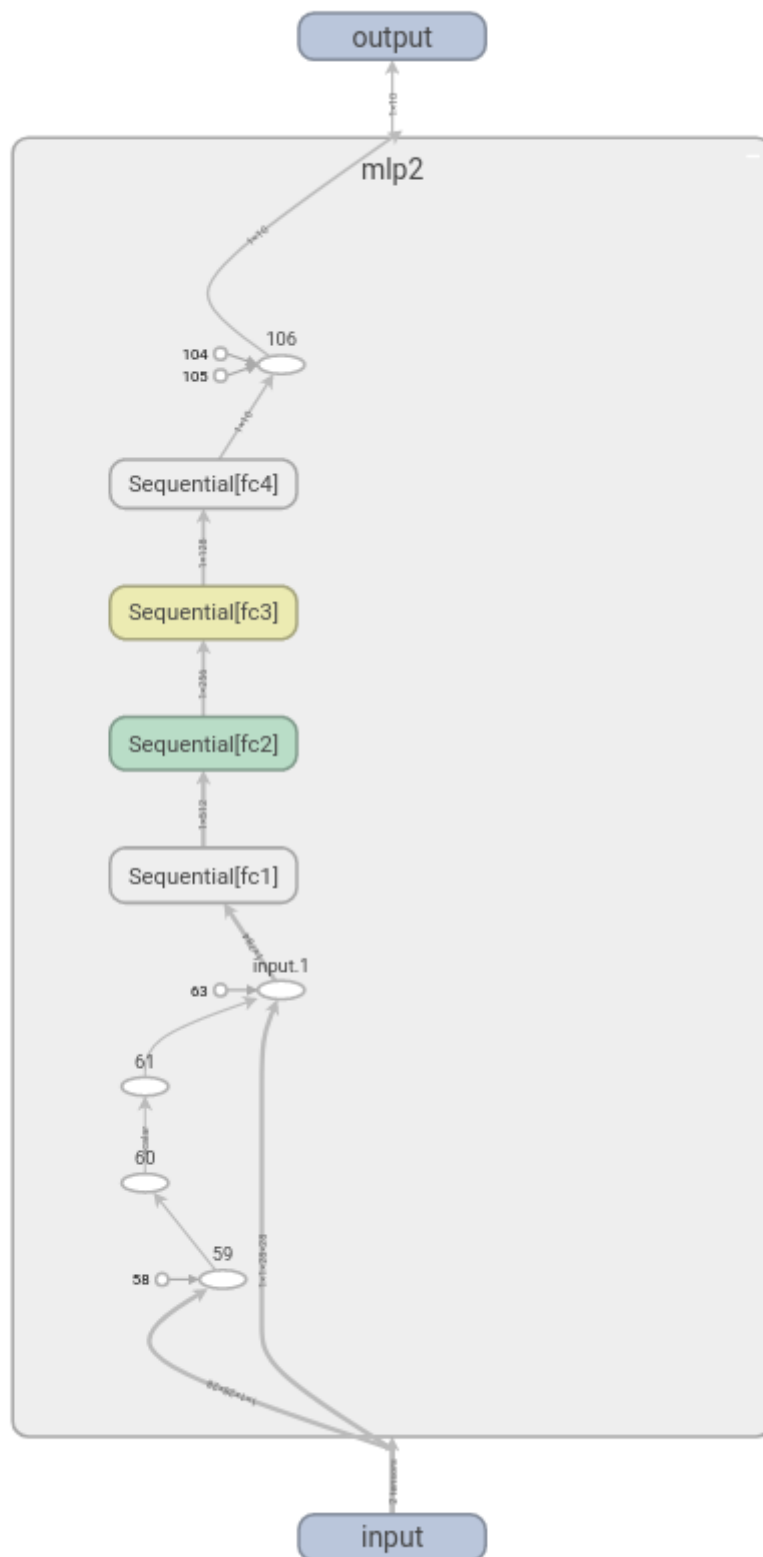
weight decay	accuracy	validation accuracy
0.00005		0.99333
0.0001	0.99335	0.99429
0.0002		0.99238
0.0003		0.99238
0.0004	0.99296	0.99452
0.0005	0.99310	0.99333

Dropout	accuracy	validation accuracy
0.6	0.99325	0.99548
0.5	0.99335	0.99429
0.65		0.99333

4.CNN与baseline对比

mlp

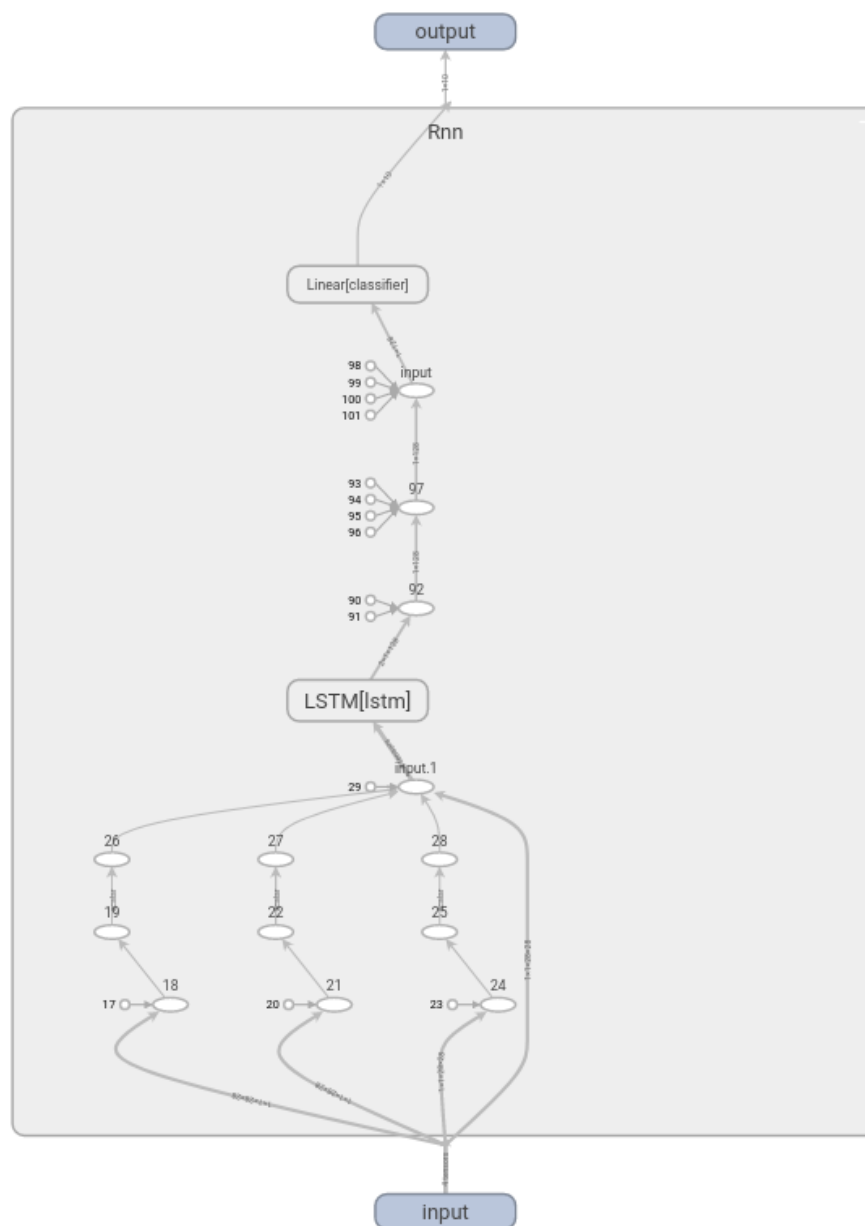
模型结构图



batchsize	accuracy	validation accuracy
32	0.97492	0.99500
128		0.97286
64		0.97429

Istm

模型结构图



batchsize	accuracy	validation accuracy
8	0.98396	0.98619
16		0.98071
32		0.97810
64		0.97476
128	0.96564	0.96714

发现和mlp以及lstm相比,发现cnn的泛化能力明显更强,不容易出现过拟合,所以在测试集上准确率更高。但是cnn缺少对全局的感知,所以通常需要加上mlp来获取全局信息。

5.问题思考

1)实验训练什么时候停止是最合适的?简要陈述你的实现方式,并试分析固定迭代次数与通过验证集调整等方法的优缺点。

当loss不再下降或者准确率不再提高时停止最合适。我的实现方式是固定迭代次数，因为这样方便比较不同模型之间的效果，而通过验证集调整则可以防止过拟合，保留最好的模型。

2)实验参数的初始化是怎么做的?不同的方法适合哪些地方?(现有的初始化方法为零均值初始化,高斯分布初始化,正交初始化等)

实验参数的初始化采用的是高斯分布初始化和pytorch默认的初始化方式。

一般用均匀分布、正态分布来初始化线性层，卷积的权重和偏重，根据具体对象又分为 xavier 的均匀、正态分布，kaiming 的均匀、正态分布。xavier 的分布适用于激活函数是 tanh 的初始化，不太适用于 ReLU。而 kaiming 则非常适用于带 ReLU 激活函数的地方。PyTorch 里的线性层的初始化为均匀分布 $U(-\sqrt{1/in}, \sqrt{1/in})$ ($a = \sqrt{5}$ 的 kaiming 分布，in 为矩阵的第二维大小)

3)过拟合是深度学习常见的问题,有什么方法可以方式训练过程陷入过拟合。

主要方法有

- 数据增强
- 提前终止
- dropout
- weight decay
- batchnorm

4)试分析 CNN(卷积神经网络)相对于全连接神经网络的优点。

- CNN相对于全连接层需要训练的参数更少，减少过拟合，需要的训练集更少
- 当CNN学习到检测某个特征的核（kernel）的时候，它可以在图片的任何位置检测到这个特征。而DNN在某个位置学习到某个特征，它只能在这个位置识别它。由于图片一般有很多重复的特征，CNN在分类等领域可以用更少的训练集比DNN泛化地好得多

6.心得体会

深度学习调参确实是一个体力活，需要大量实验，经常会搞混乱，所以一定要先设计好实验，并且养成良好的参数记录习惯，不然很容易白费功夫。其实很早就得到最佳结果了，但还是继续试了很多参数。想起凸优化老师说的一句话，最可怕的是我们一开始就知道了全局最优解，但我们自己却不知道。这次实验到底有没有得到全局最优，最终我还是不知道。

另外这次作业让我熟悉了pytorch的使用，pytorch相比于tensorflow确实很好用，果然是第一生产力！