神经网络学习学

实验描述

本次实验的主要任务依然是 MNIST 手写数字识别,对照比较三组实验的效果。

- 1. 实验一、先在0~4的数字上训练完模型、再把5~9的数据放进去训练
- 2. 实验二、先在0~4的数字上训练完、再把所有的数据放进去训练
- 3. 实验三,直接使用所有的数据训练

最后我们将分别讨论三个实验中最后模型的表现。

一些参数:

- 1. 主要的评价指标为损失函数和正确率
- 2. 每个实验都跑了五次

模型介绍

整个实验中的CNN模型实际上只有一组结构,描述如下:

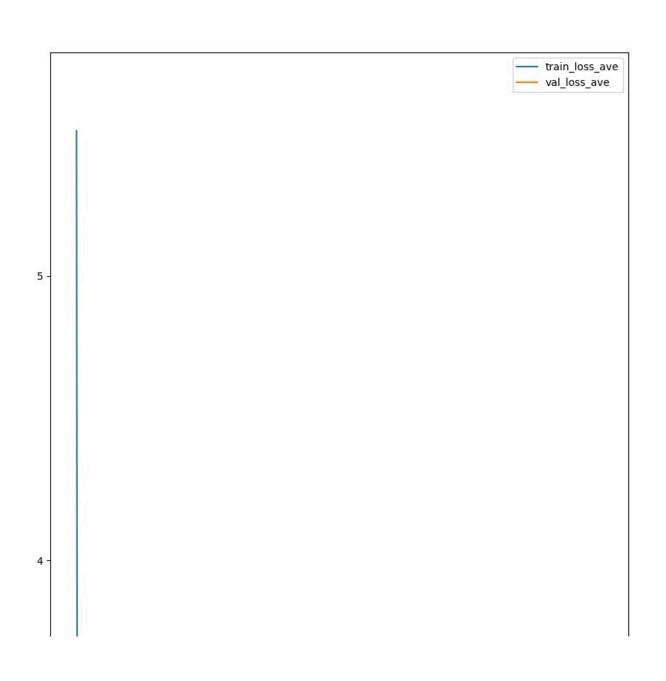
- 1. 输入:尺寸为28x28的矩阵,通道数为1
- 2. 卷积层: kernel size为5x5, padding为2, stride为1, 输出通道为8, 激活函数为普通relu
- 3. 池化层: 使用max pool 2d, kernel size为2
- 4. 卷积层: kernel size为5x5, padding为2, stride为1, 输出通道数为16, 激活函数为普通relu
- 5. 池化层: 使用max pool 2d, kernel size为2
- 6. 平铺操作:将前面CNN池化层的结果平铺
- 7. 全连接层:输出为1024个神经元
- 8. dropout层
- 9. 全连接层:输出为128个神经元
- 10. log_softmax输出为10个概率值

实验结果

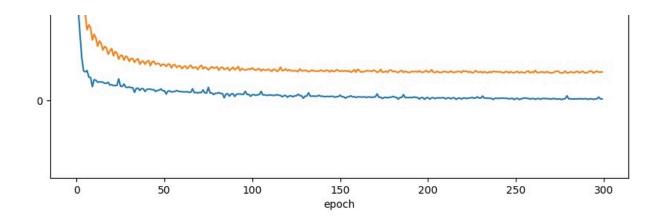
实验一

本实验是现在0~4的数字上训练,结束后再把5~9的数字放进去继续训练。

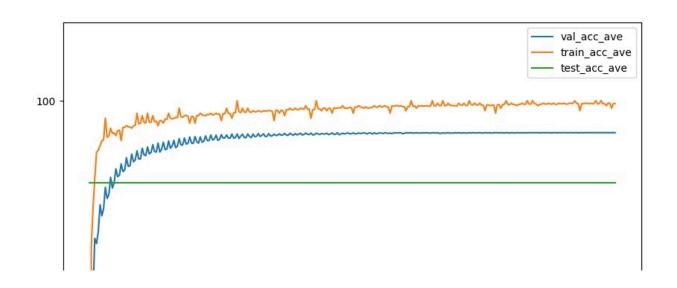
首先是训练过程中的loss情况:

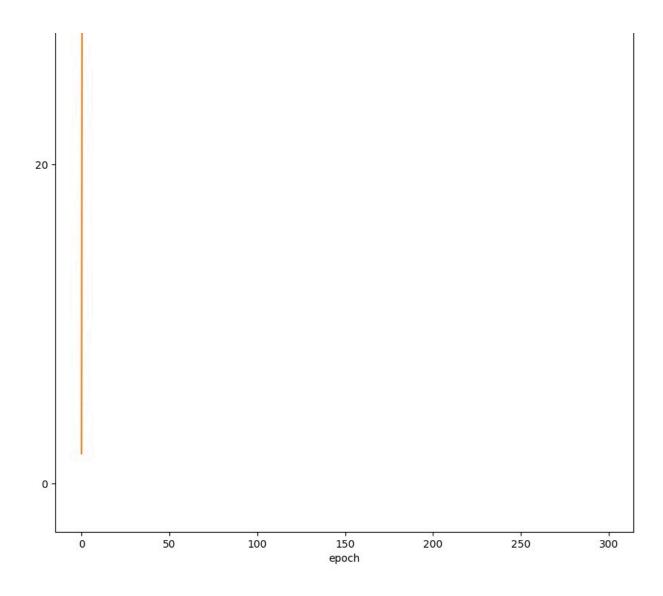


3 ave_loss 2 -1 -

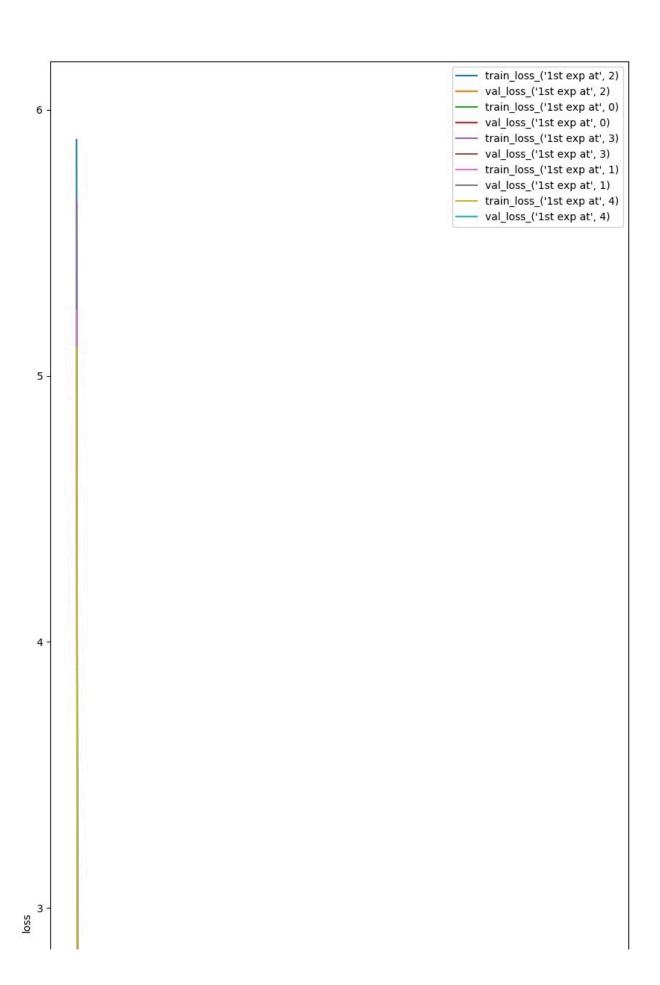


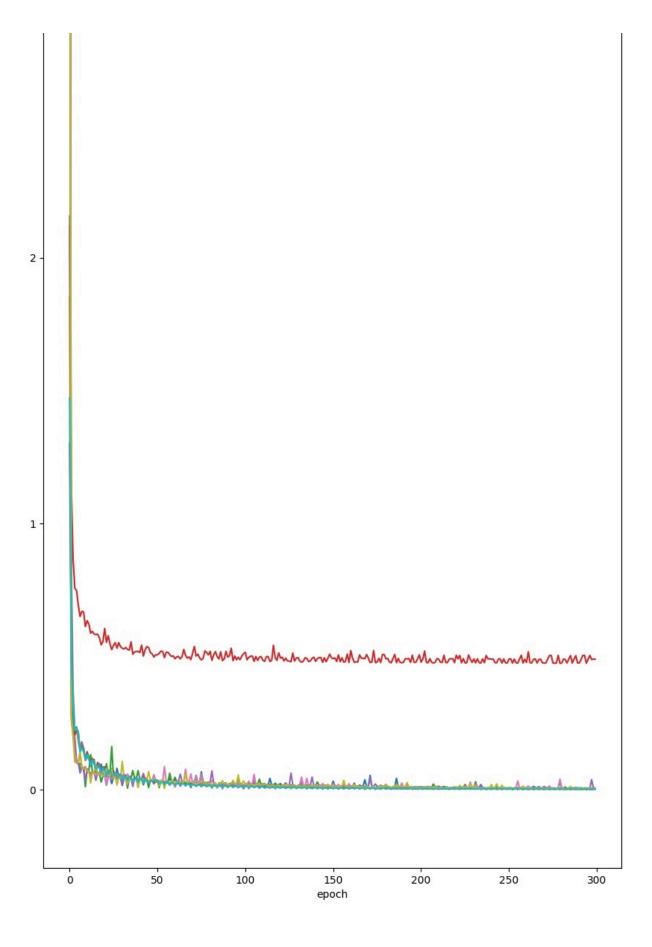
接下来是测试过程中, 准确率的情况。

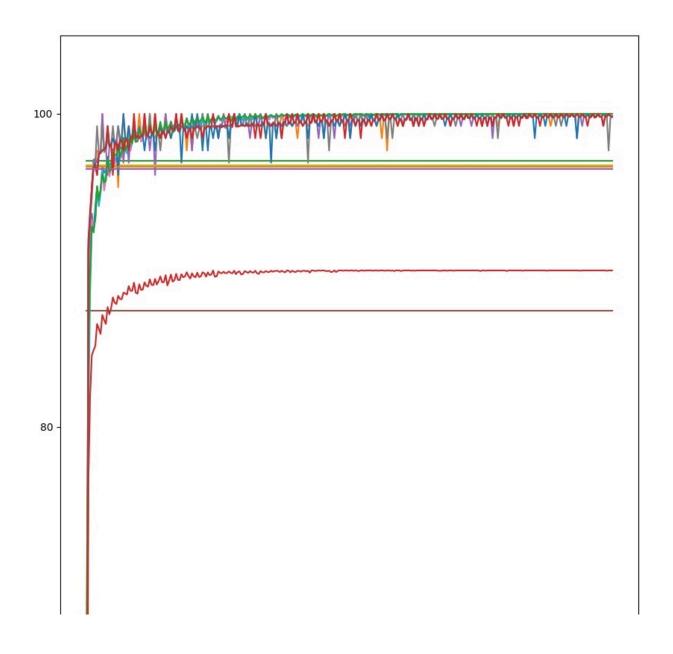




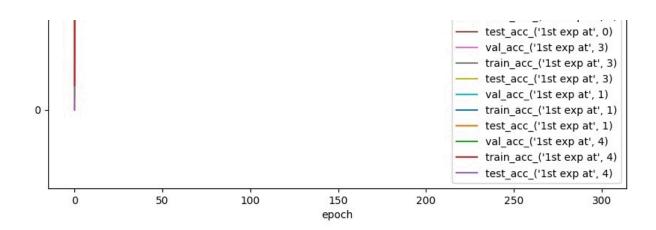
上面是多次试验的均值,下面是原数据的详细情况:







60 40 20 val_acc_('1st exp at', 2) train_acc_('1st exp at', 2) - test_acc_('1st exp at', 2)
- val_acc_('1st exp at', 0)
- train acc ('1st exp at', 0)



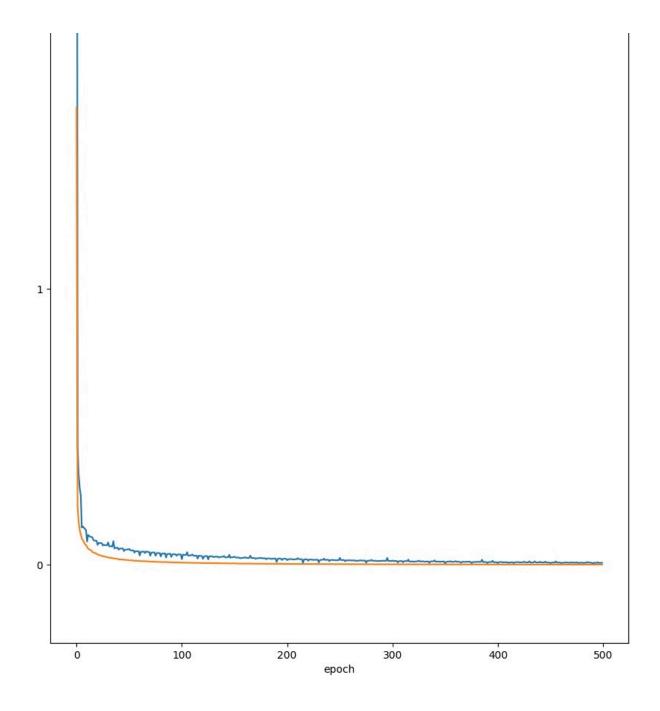
实验二

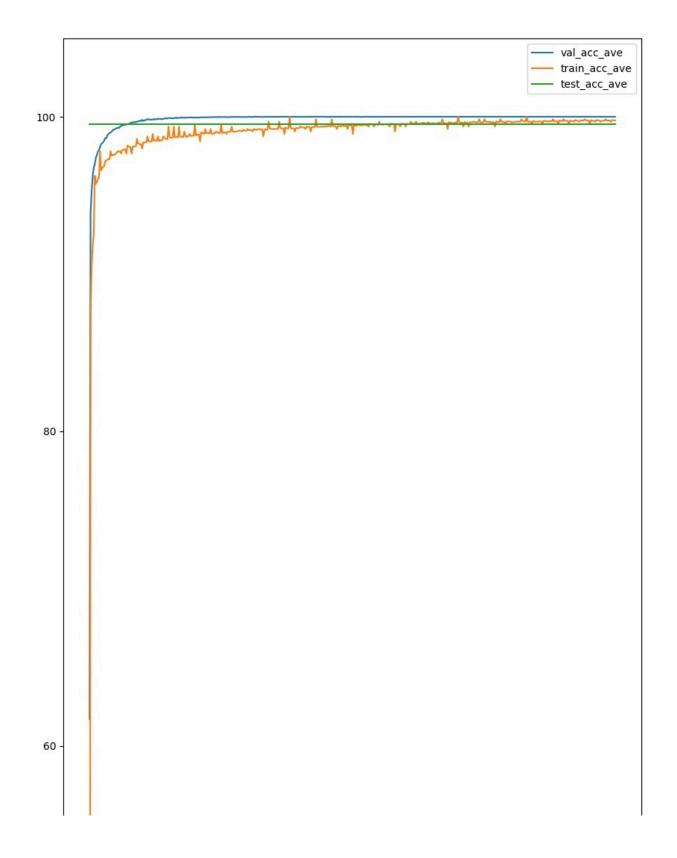
本实验是现在0~4的数字上训练,结束后再把5~9的数字放进去继续训练。

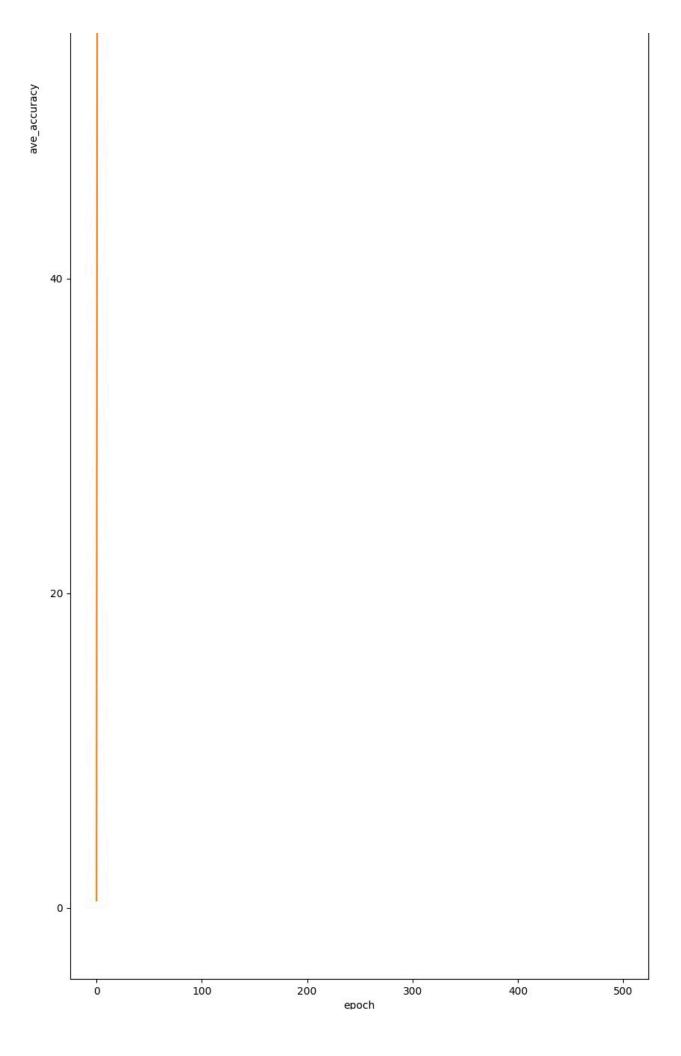
首先是训练过程中的loss情况:



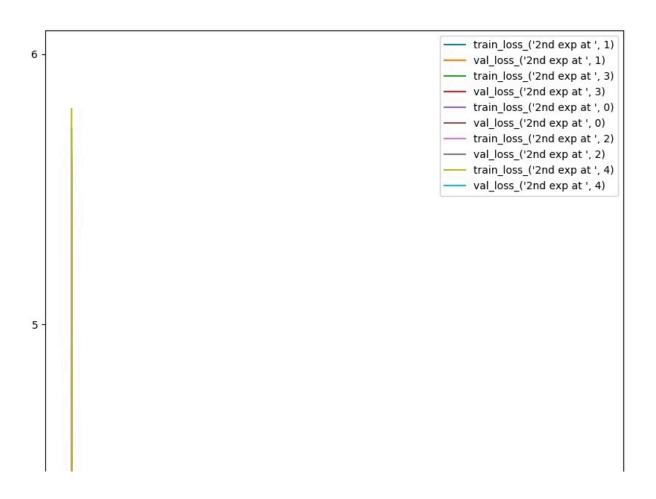
5 -4 -3 ave_loss 2 -







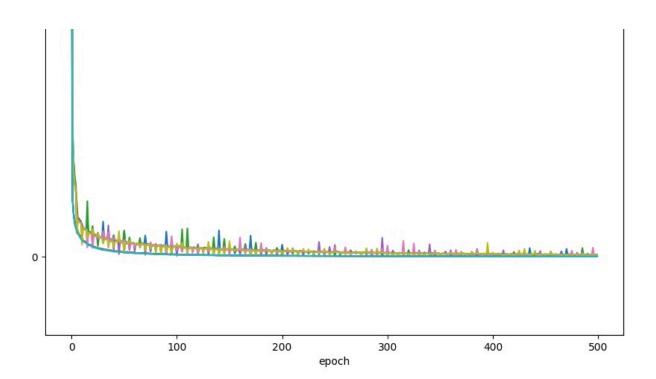
上面是多次实验的均值,下面是详细的原始数据:

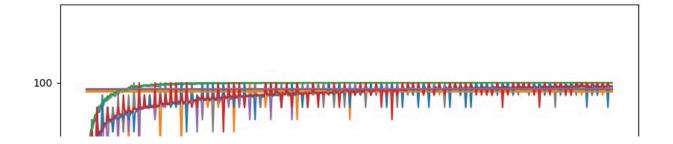


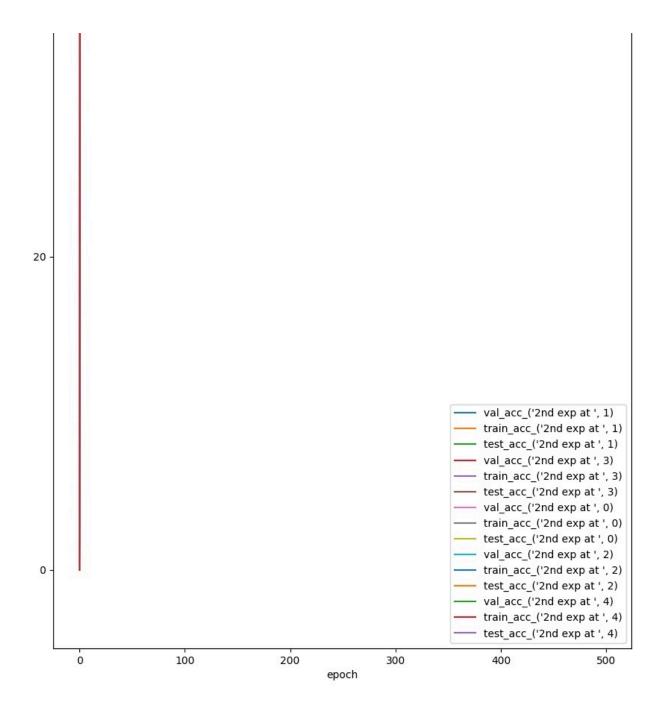
4 -3 loss

2 -

1 -



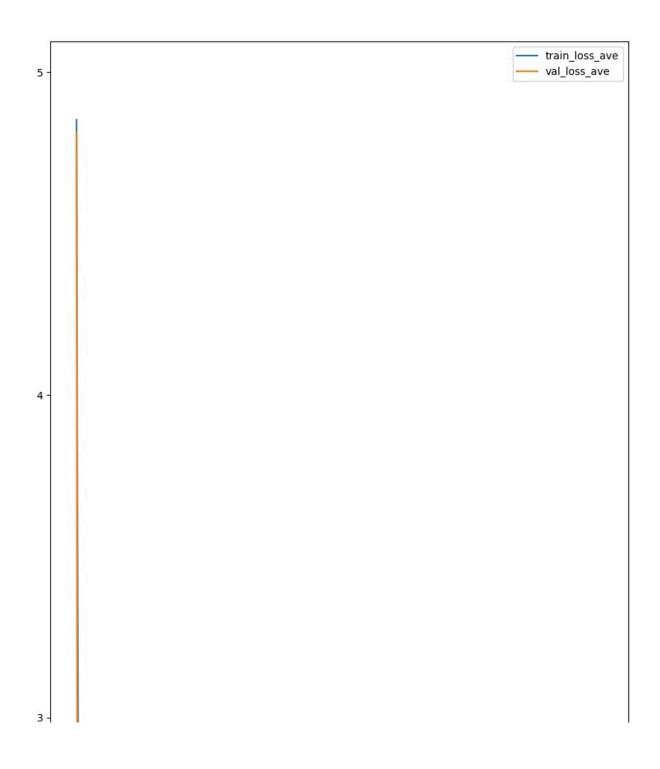




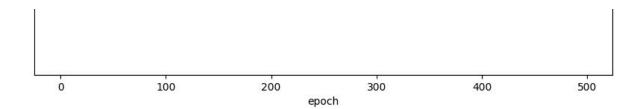
实验三

本实验是现在0~4的数字上训练,结束后再把5~9的数字放进去继续训练。

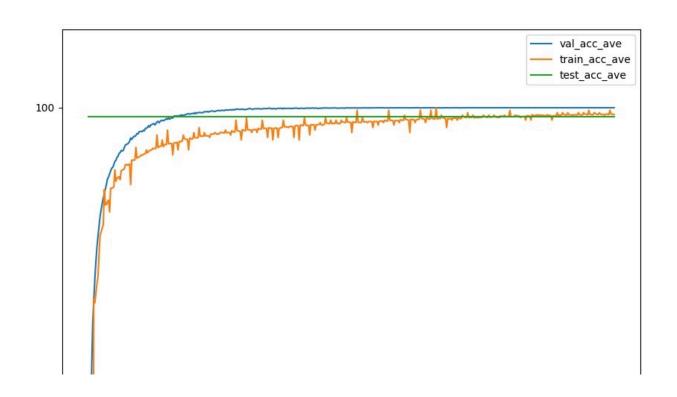
首先是训练过程中的loss情况:

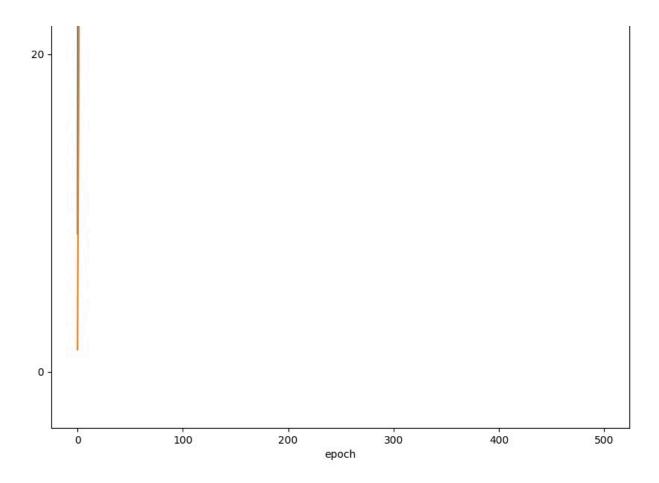


ave_loss 2 -1 -0 -

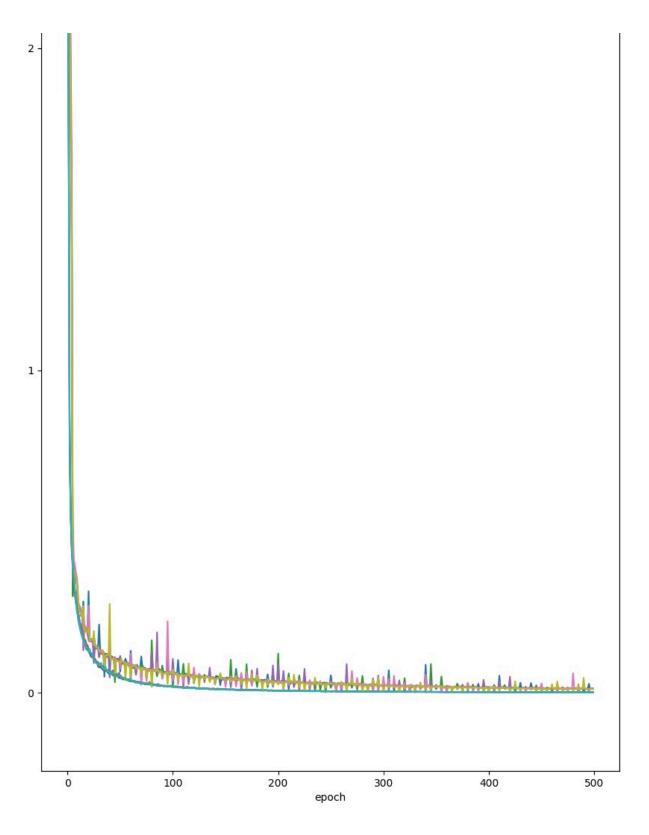


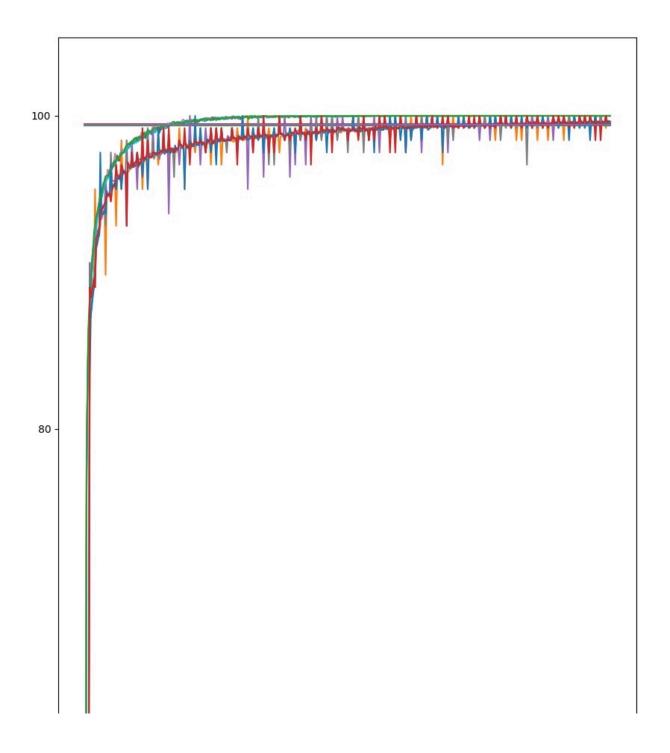
接下来是在测试过程中的准确率的情况:





上面是多次试验的均值情况,下面是详细的原始数据:





60 40 20 val_acc_('3rd exp at ', 0) train_acc_('3rd exp at ', 0) test_acc_('3rd exp at ', 0) val_acc_('3rd exp at ', 2) train_acc_('3rd exp at ', 2) test_acc_('3rd exp at ', 2) val_acc_('3rd exp at ', 1) train_acc_('3rd exp at ', 1) test_acc_('3rd exp at ', 1) val_acc_('3rd exp at ', 4) train_acc_('3rd exp at ', 4)

