

人工神经网络第二次作业报告

计52 2015011245 周京汉

本次作业的主要要求是自己完成一个CNN网络的卷积和pooling的功能，然后进行loss和accuracy的实验对比。因此，本次实验报告主要分为3个部分，分别讨论CNN网络的loss对比，CNN网络与MLP各项指标的对比和实验过程可视化的问题。

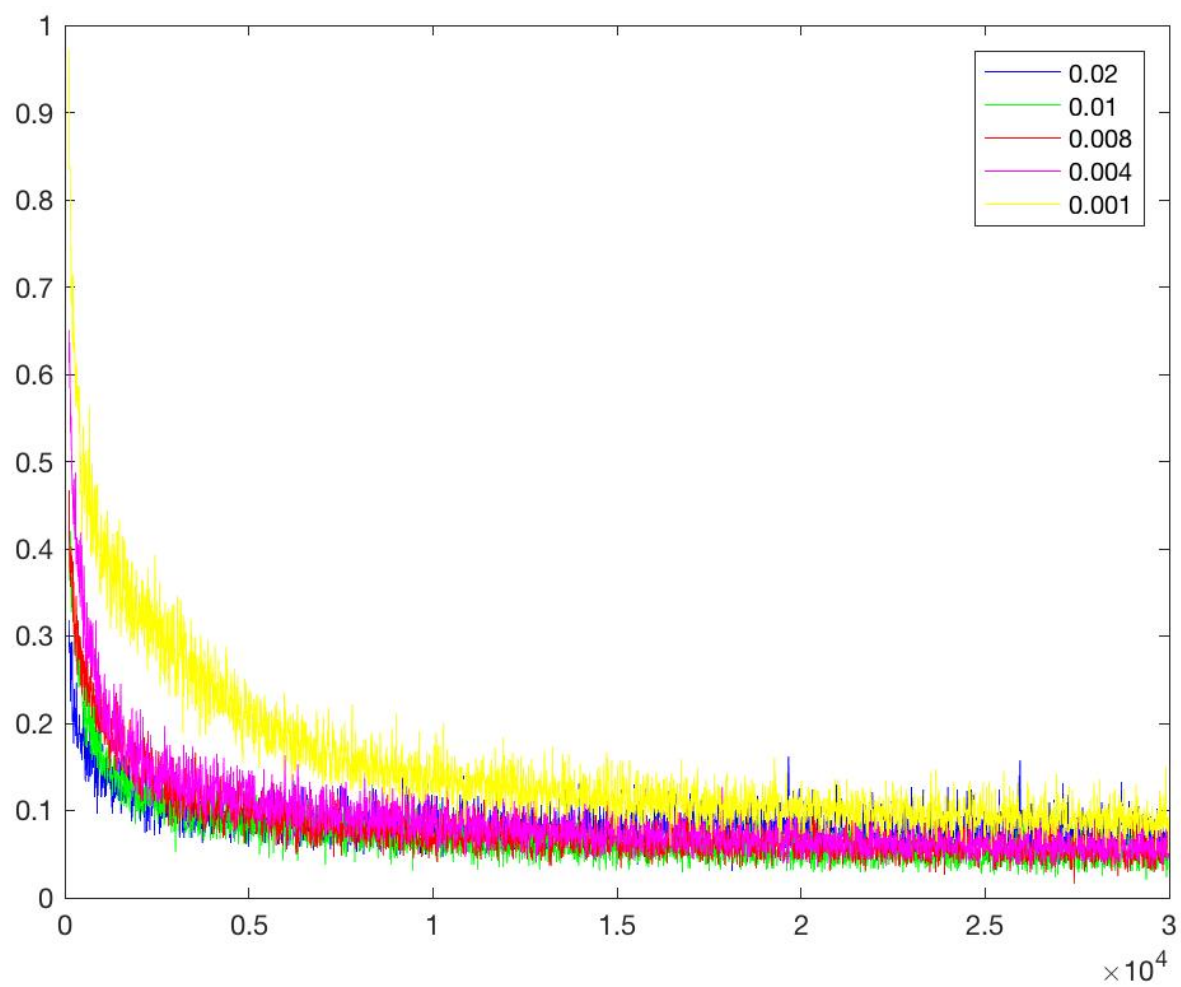
一、loss对比

1、learning_rate参数

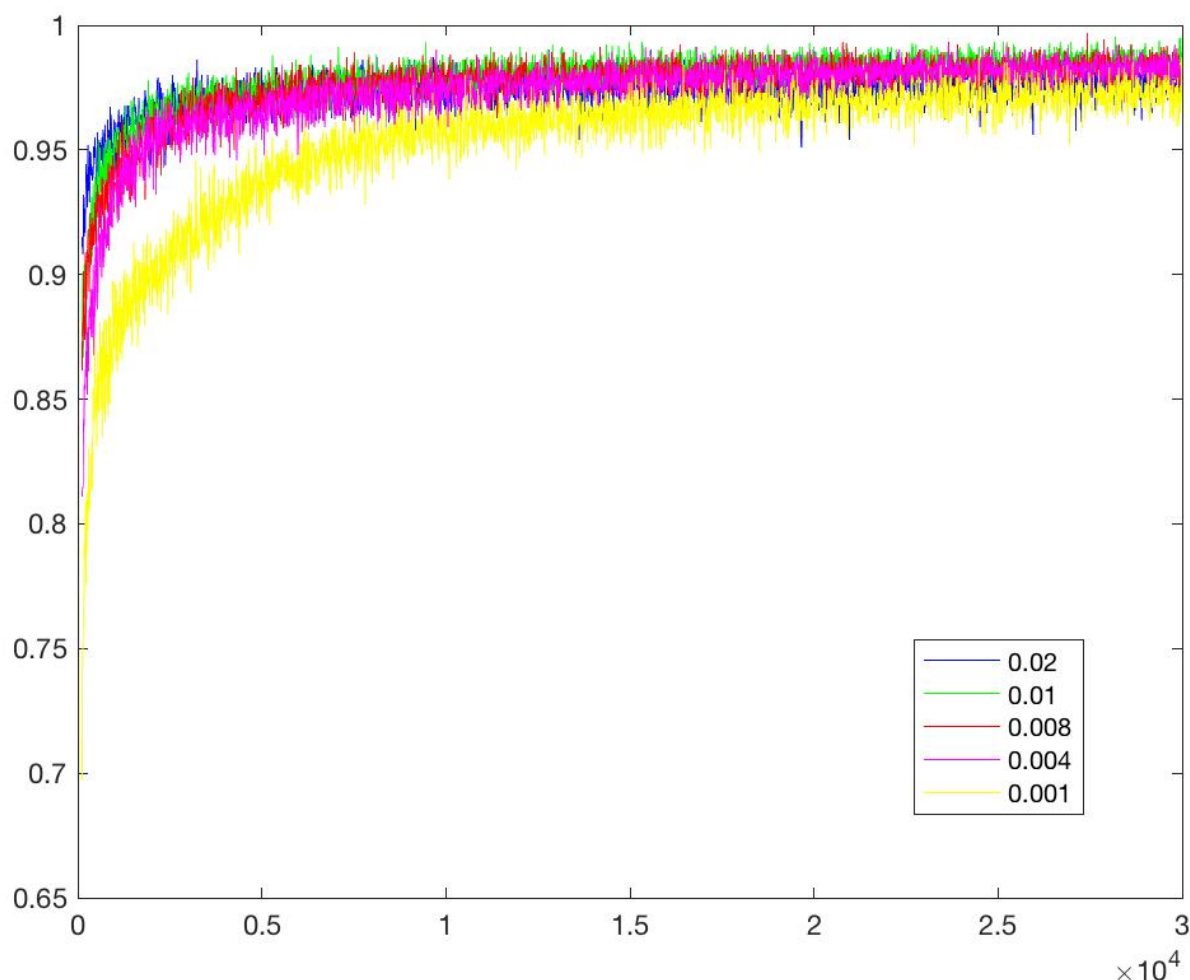
首先进行调参对比的是learning_rate参数。在调参过程中，我默认剩下两个参数分别为weight_decay为0.0001，momentum为0.9。max_epoch的值为30。

对于这项参数，我主要是进行了5次实验，其中learning_rate的参数设定分别为：0.01，0.02，0.008，0.004和0.001。如下图，为这5次实验在测试集上表现出的loss和accuracy的情况，图中显示了不同参数对应曲线的颜色。

loss图：



accuracy图:



从图中可以看出，

最终的结果为参数设为0.02时accuracy为98.00%，0.01时为98.13%，0.008时为98.23%，0.004时为98.01%，0.001时为97.21%，测试集结果与训练集结果趋势一致。

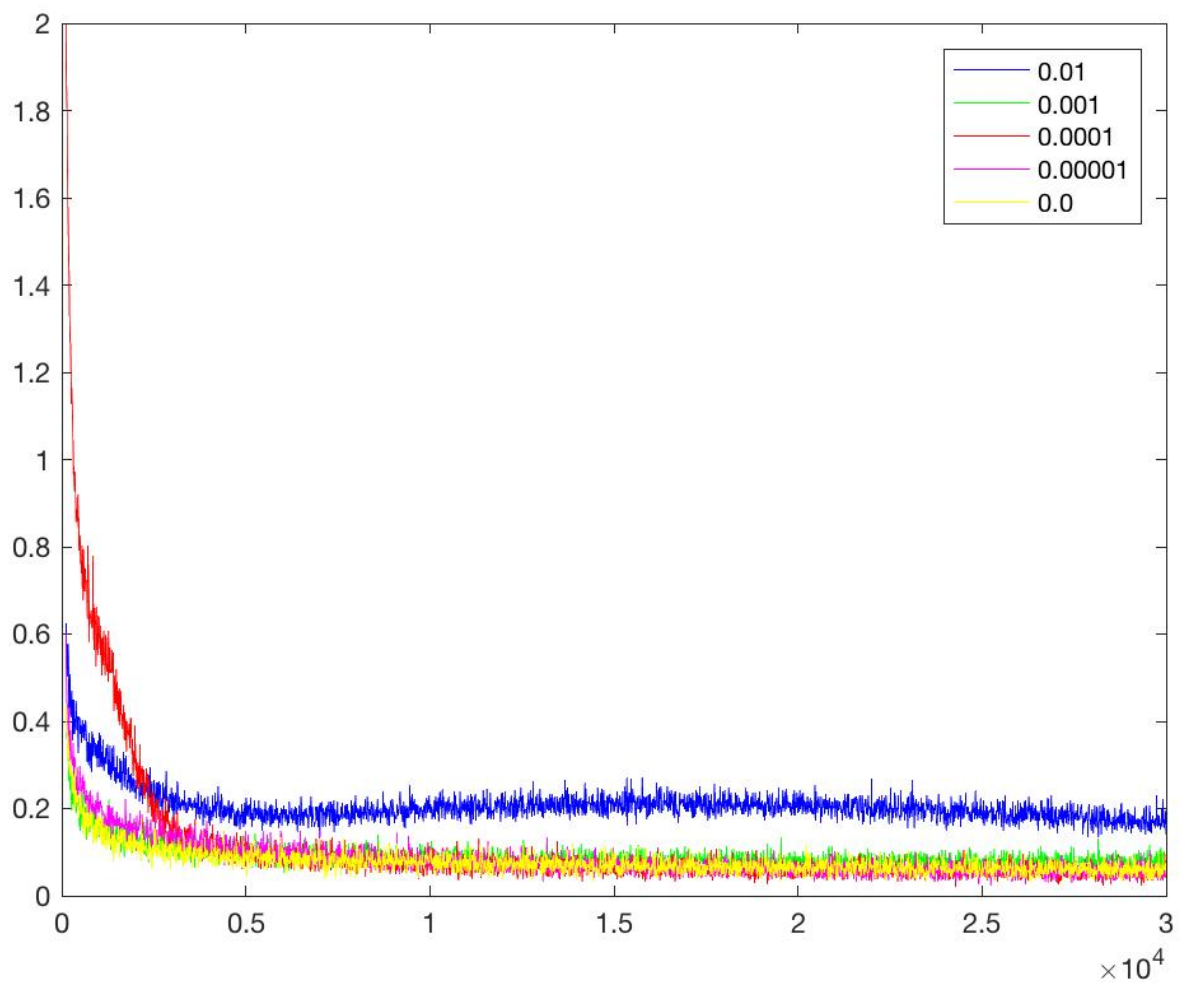
结果显示，0.02会导致后面震荡较大，不利于准确率进一步提升，因此最终的结果不如0.01和0.008好。0.001的下降速度太慢了，最终收敛结果也不好。0.004，0.008和0.01的结果相近，从这次实验中看，0.008和0.01的效果略好一些。

2、weight_decay参数

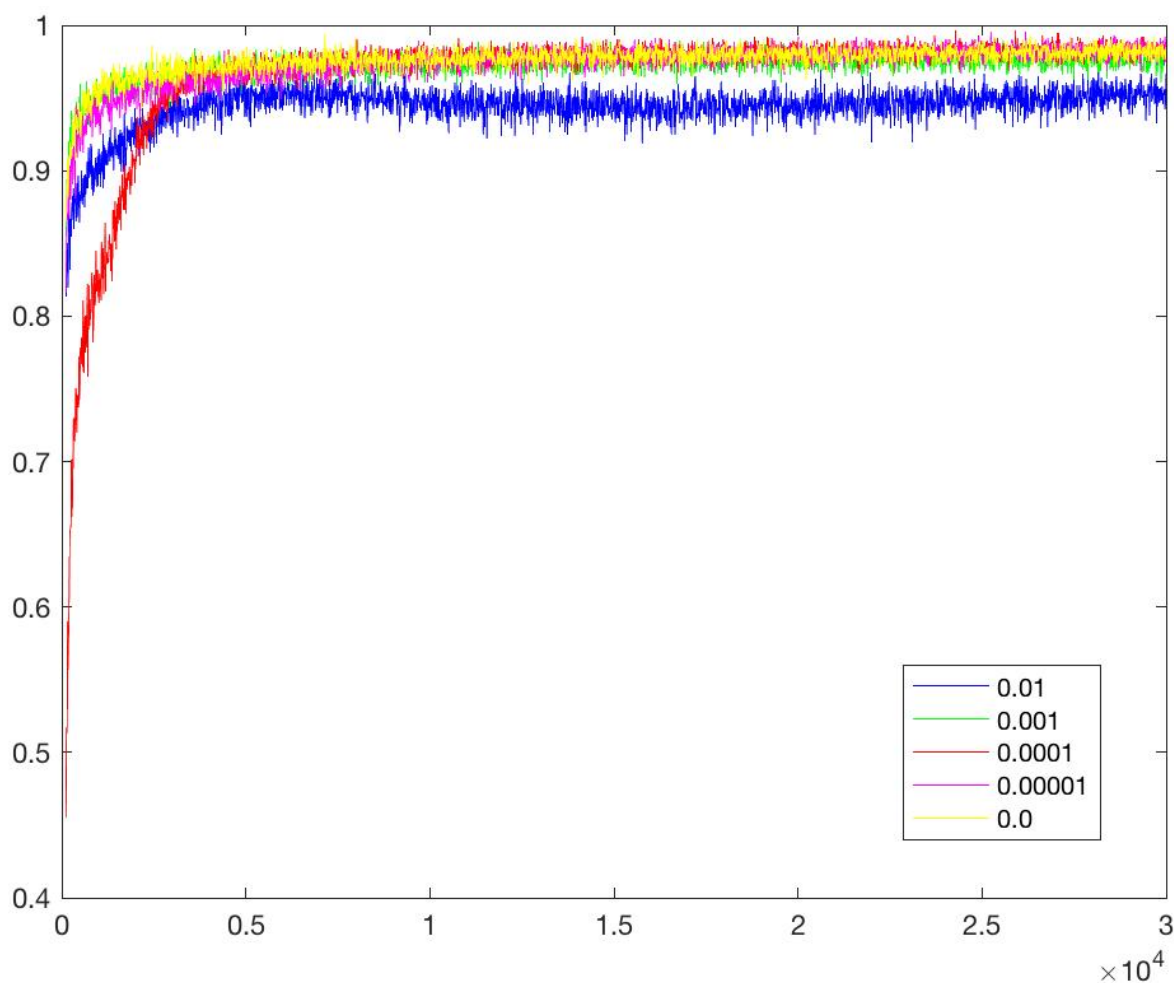
第二个进行的实验为weight_decay参数的对比实验。我选择了如下参数设定进行对比实验，数值分别为：0.01，0.001，0.0001，0.00001，0.0。在对比实验中，其余几项参数分别固定为learning_rate为0.01，momentum为0.9，max_epoch固定为50。

下图为这5次实验在测试集上表现出的loss和accuracy的情况，图中显示了不同参数对应曲线的颜色。

loss图：



accuracy图:



图中可以看出，在完成了50个epoch的情况下，结果的收敛情况还是比较好的。

图中的蓝线为0.01的情况，可以看出，其明显在loss还较高的情况下就完成了收敛，原因是weight_decay数值太多，导致后拖力太强，使loss无法继续减小，因此该数值不好。

其次是0.001的情况。虽然在一开始的时候，0.001的loss下降情况十分的迅速，但是很快便达到了上限，之后被剩下集中情况反超，最终loss下降情况仅好于了0.01的情况，因此也不好。

最后就是剩余三组数据的对比。其中0.0001的情况在刚开始的下降速度十分的缓慢，但是后面还是追赶上了其他的情况，我认为这可能是该数值下的一种feature。

图中经过对比我们可以发现，黄色的线为参数设为0的时候，明显其震荡更大，并且下降程度不如另外两种好。而0.00001看起来loss的下降情况与0.0001差不多，然而它们在最终在测试集的表现上则存在差距。

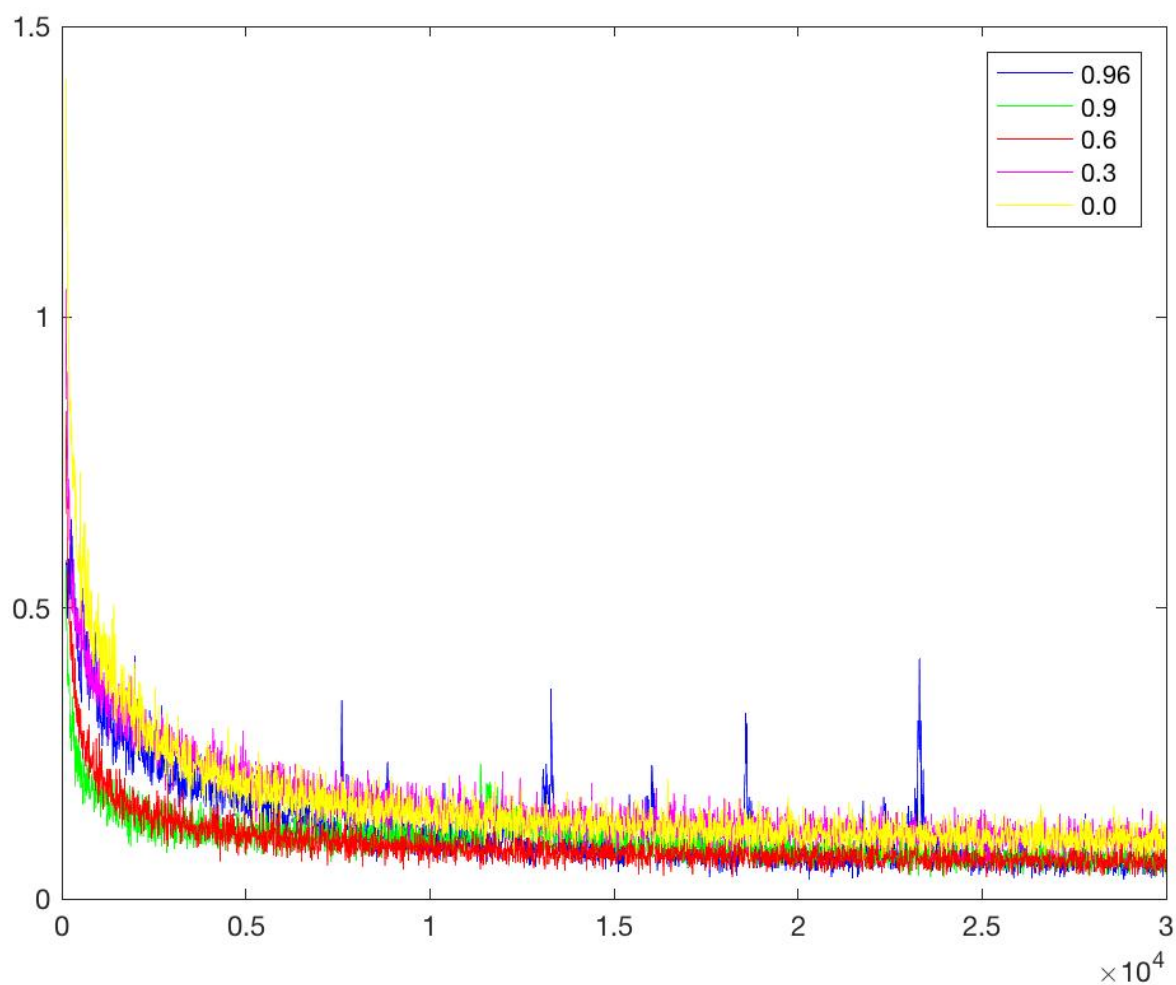
最终在测试集中的表现分别为：0.01时为95.78%，0.001时为97.74%，0.0001时为98.17%，0.00001时为98.00%，0.0时为97.96%。因此0.0001的表现更好。其中0.0001和0.00001的比较最为关键，0.0001的表现更好一些。我认为这是因为0.00001数值设定过小，导致过拟合较为严重，因此在训练集表现相近的情况下，测试集的表现0.0001的时候更好些。因此，最终我认为，weight_decay的最好取值为0.0001。

3、momentum参数

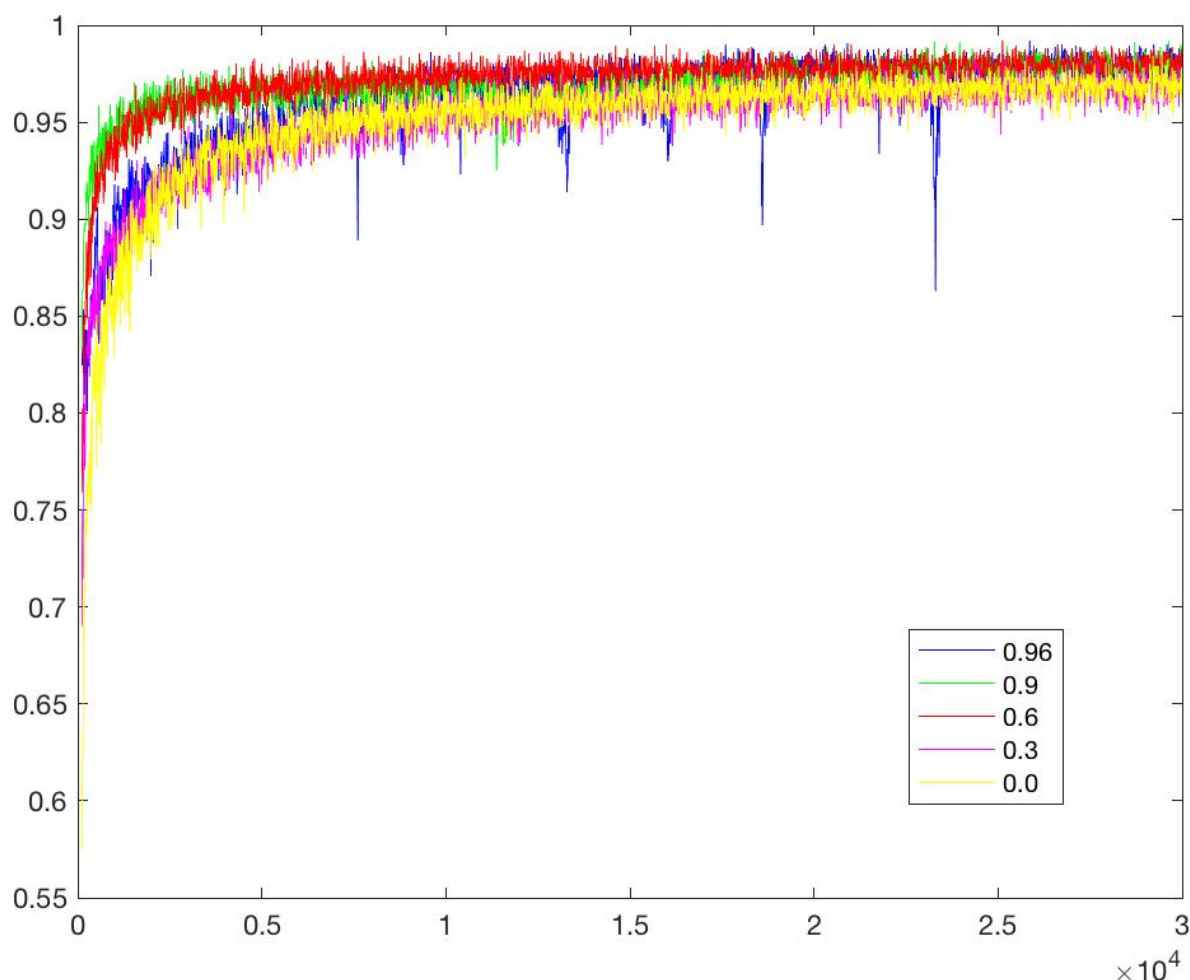
第三个进行对比实验的参数是momentum。momentum主要为训练过程中的惯性系数，在实验中我选择了如下个数值进行对比：0.96，0.9，0.6，0.3和0.0。在对比实验中，其余的参数默认为learn_rate为0.01，weight_decay为0.0001，max_epcoh为50。

下图为这5次实验在测试集上表现出的loss和accuracy的情况，图中显示了不同参数对应曲线的颜色。

loss图：



accuracy图：



从图中情况看，主要可以分为两种类型。第一种为0.96，0.3和0.0；第二种为0.9和0.6。

首先是第一种，他们在最开始的下降都相对较慢一些。其中，0.96的下降速度稍快，但是比起另一组来说还是很慢了，并且在最后收敛的时候，会出现较大的震荡，很不稳定，因此该数值不是一个好的参数。而0.3和0.0在下降速率和最终收敛结果上面都表现不好，因此我认为这两个数值是不好的参数。

第二组参数在loss下降速率和收敛情况方面均优于前一组数据。其中在刚开始的时候0.9的下降速率更快，但是最终的收敛情况为0.6更好且0.6的震荡更加的小。

最终5组实验的测试集准确率分别为：0.96时为97.83%，0.9时为97.75%，0.6时为97.80%，0.3时为96.97%，0.0时为97.12%。前三个表现差不多，但是0.6明显收敛更稳定，因此我认为0.6为最好的一个参数。

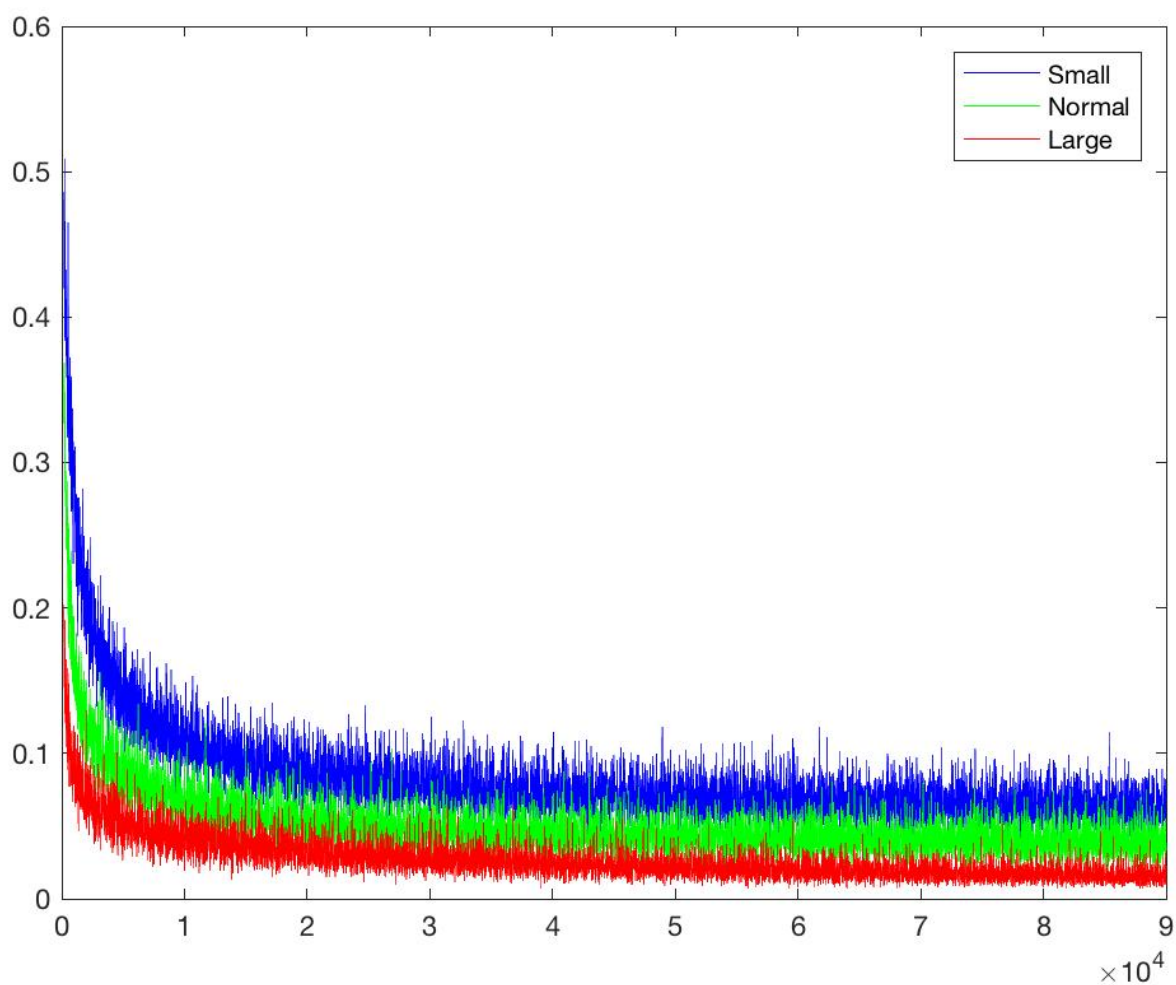
4、不同网络结构对比

最后进行对比的是不同结构下的网络的运行情况。结构分为三种：Small，Normal和Large。其中small为只有一个卷积和pooling层，且channel为4，kernel_size为3；Normal即为作业中默认的结构，两层卷积和pooling，channel为4，kernel_size为3；Large为两层卷积和pooling，channel扩大为8，kernel_size为3。

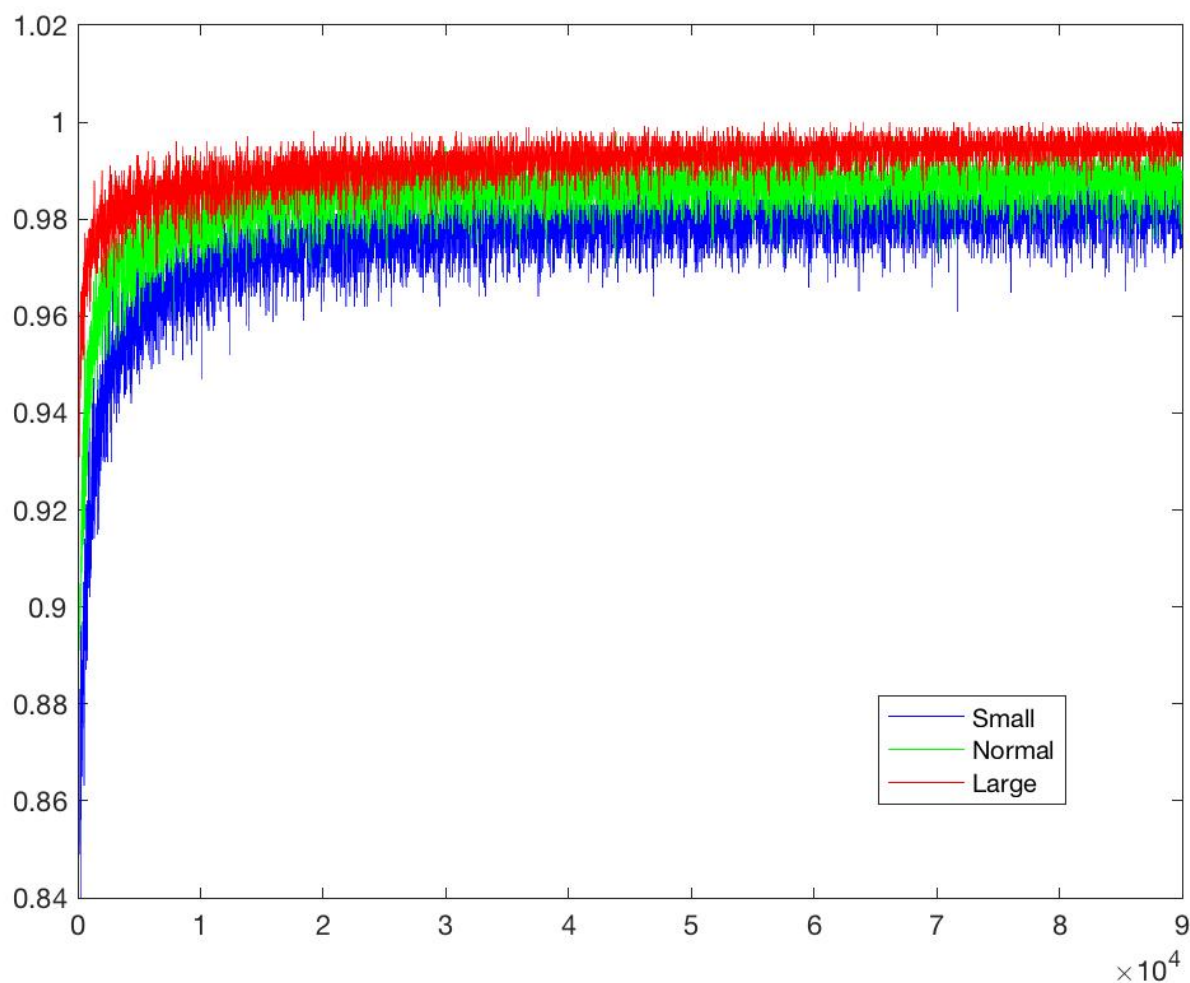
在此番对比之中，三个网络的参数设定均为：learning_rate为0.01，weight_decay为0.0001，momentum为0.9，max_epoch为150。其中，唯一的例外是在最大的Large结构中，learning_rate采取了随着epoch的大小由0.02至0.005正比下降的策略。

下图为这3次实验在测试集上表现出的loss和accuracy的情况，图中显示了不同参数对应曲线的颜色。

loss图：



accuracy图：



从图中我们可以看出来，随着网络的增大，运行的结果越来越好，其中Large网络的收敛速度，收敛情况都是最好的。但是其运行速度很慢，Small网络的运行速度是Normal的4倍，是Large的12倍左右。

最终对比结果为，Small的accuracy为98.12%，Normal的accuracy为98.68%，Large的accuracy为98.87%。

二、对比MLP

1、运行时间

首先是关于运行时间的对比。在时间方面，MLP显然比CNN的运行速度要快许多倍。我认为这是由于在CNN中，本身网络的层数就较多，神经节点也较多，同时还存在大量的卷积运算，这使得CNN的速度会大幅低于MLP的速度。

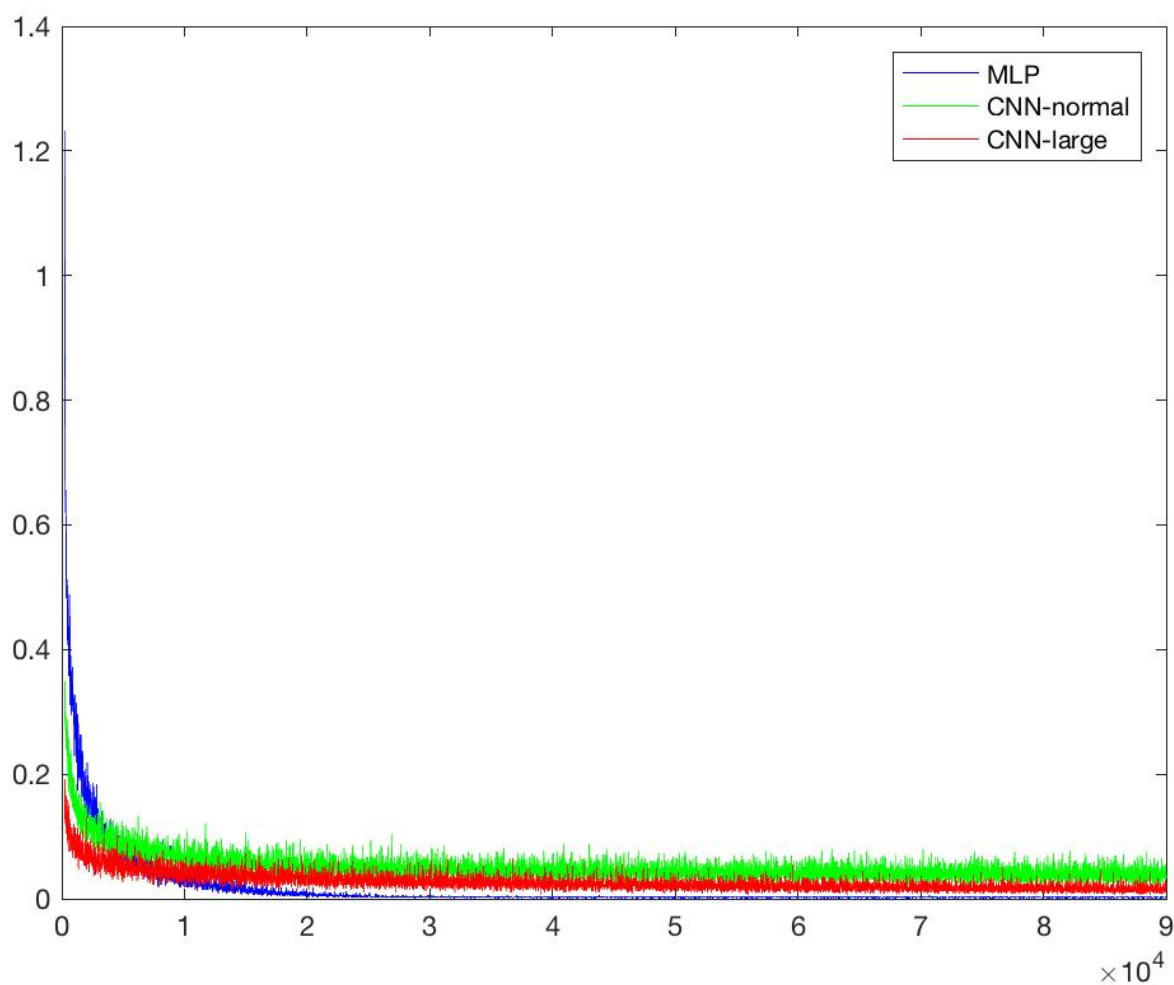
根据上一次作业的结果，有800个结点的2层MLP的速度大约为0.65s，50个iterations，而这次的双层CNN网络（作业默认的结构），速度为10.5s，50个iterations。速度明显为MLP快于CNN。

2、收敛情况

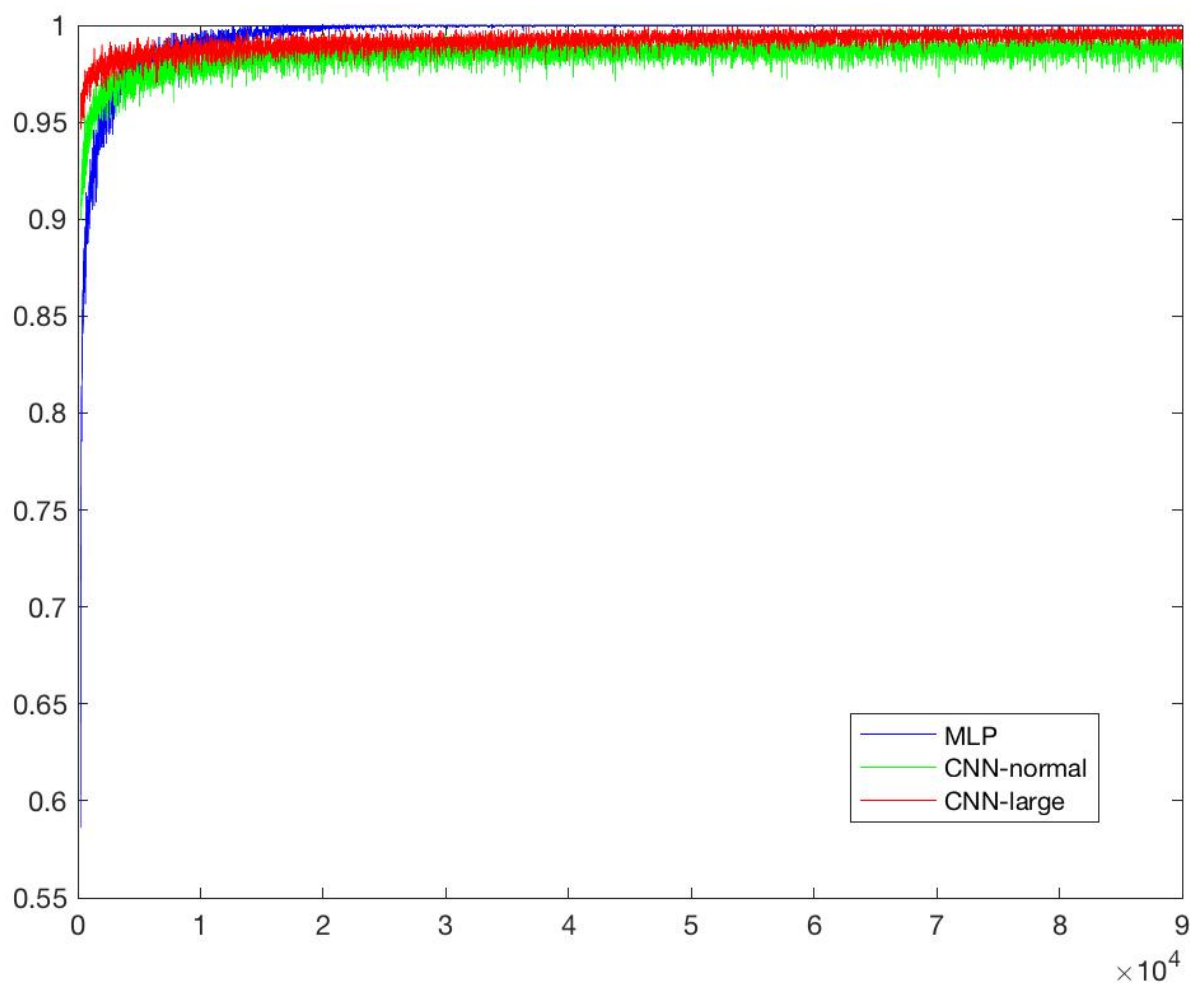
在将上次作业的MLP的loss变为Softmax之后，我重新按照如下参数进行了测试：learning_rate为0.01，weight_decay为0.0001，momentum为0.9，max_epoch为150。并与上文提到的Normal和Large网络进行对比：

下图为这3次实验在测试集上表现出的loss和accuracy的情况，图中显示了不同参数对应曲线的颜色。

loss图：



accuracy图：



从图中可以看出，MLP在最开始的收敛速度不如CNN两个网络快，但是在训练集上，收敛最终的情况为MLP更好一些，数值基本接近0了，并且什么震荡。

但是最终的结果为MLP为98.35%，而CNN的两个分别为98.68%和98.87%，为CNN的表现更加好些，因此MLP的过拟合明显更加的严重。

3、参数对比

在参数方面，与上一次的调参相对比，由于MLP结构不同参数不同，因此我们只对比momentum和weight_decay。

momentum中，MLP中多为0.9更好，但是在本次的CNN实验中，实验结果为0.6略好于0.9的情况。

weight_decay方面，仍然为0.0001更好一些，效果相近。

参数个数上，CNN中增多了channel这个参数，来分别控制不同的特征。

4、准确率

在上一次MLP的实验中我最好的情况是两层Relu的情况，为98.67%，而本次实验中最后最好的结果为98.87%。可能存在一些网络规模上的差距，但是从现在的角度来看，CNN的准确率要高于之前MLP的。

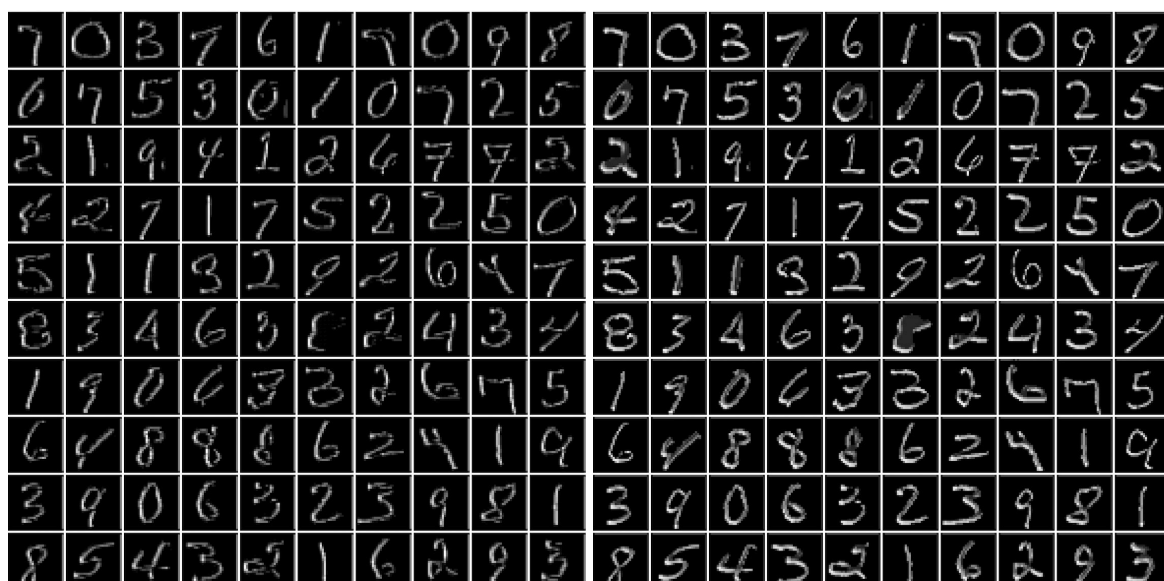
三、可视化

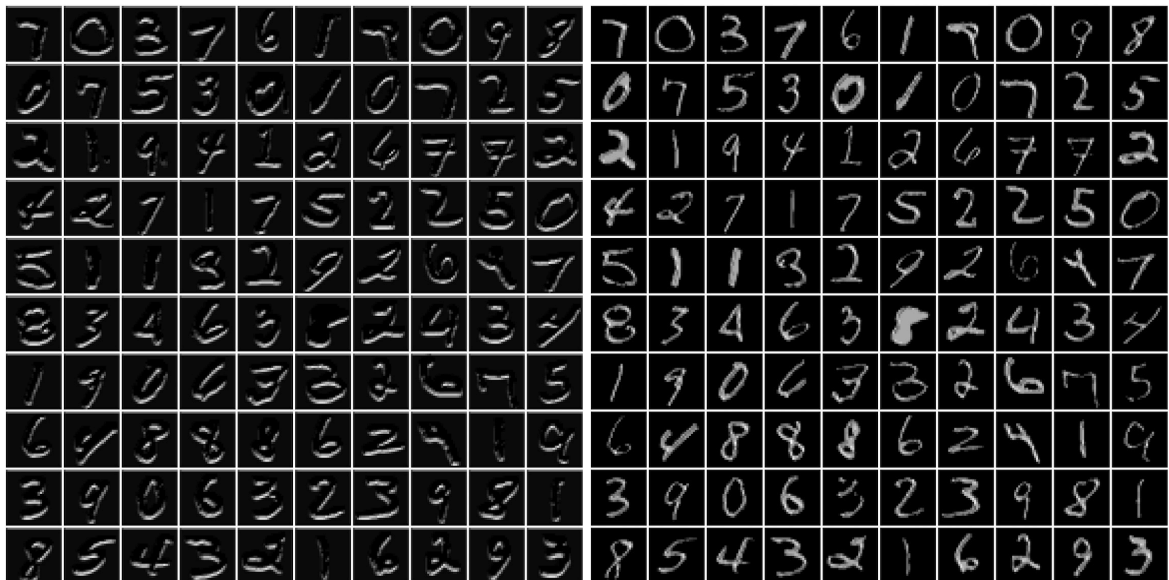
该部分的要求为输出第一个convolve层的4个channel的输出结果，因此我调用了助教在文档中给出的链接的网站中的函数，然后分别输出了input和4个channel的图像。

首先我输出了在刚开始的情况下每个channel的表现情况。首先是input:

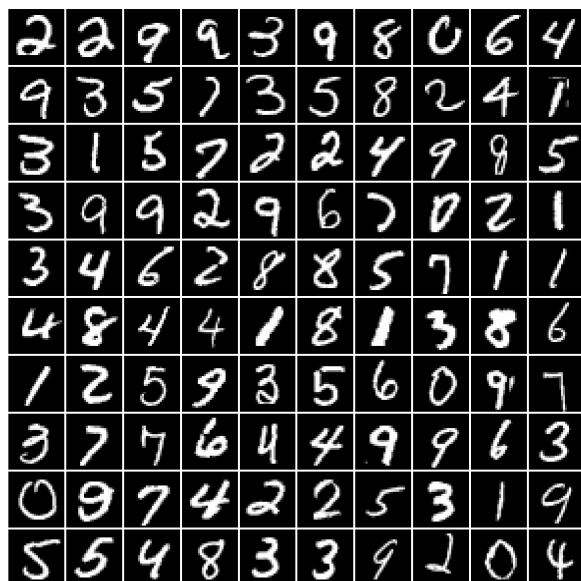


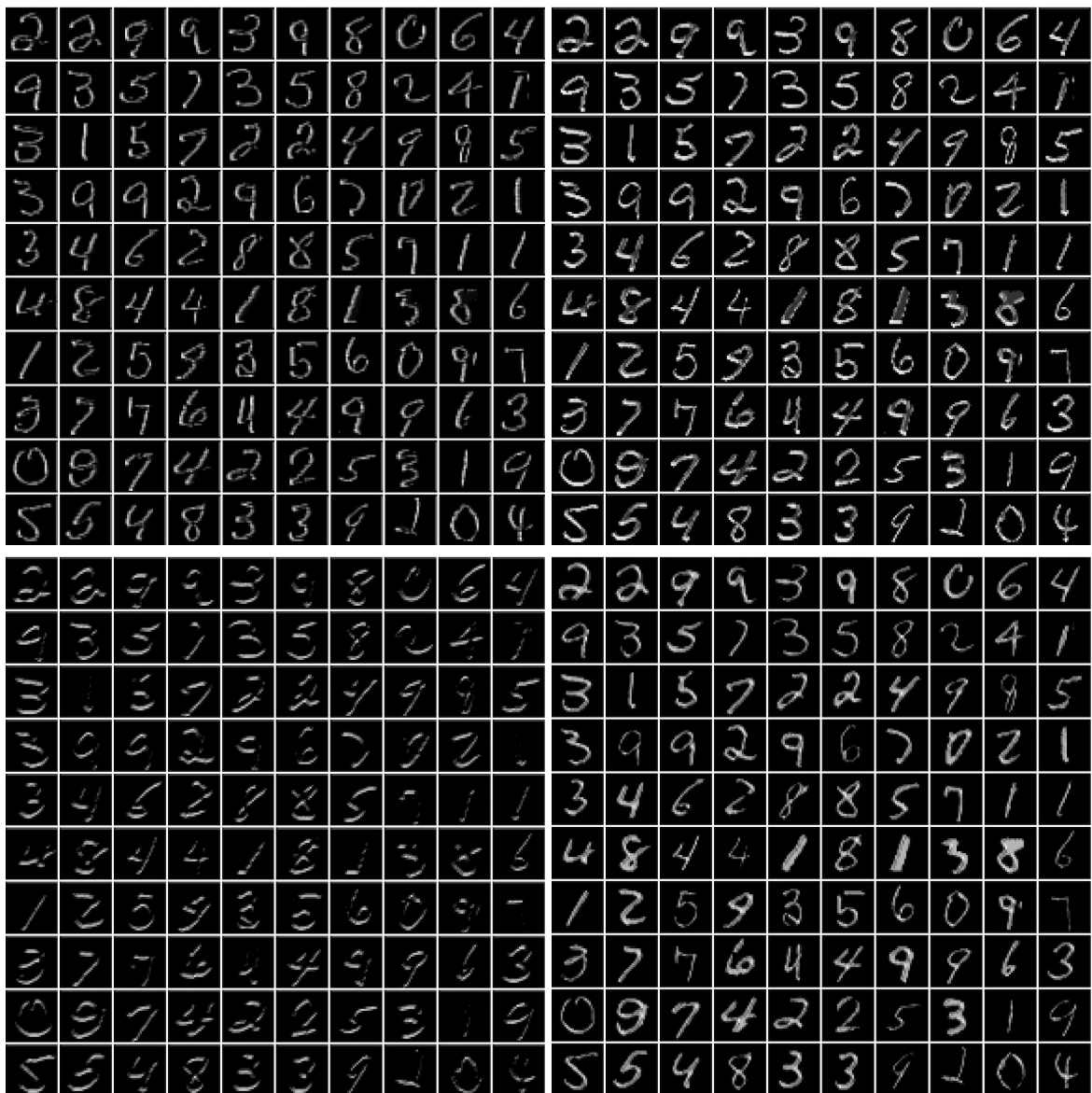
随后4个channel的结果分别为如下4张图：





可以看的出来，4张图中的数字表现各有不同，说明它们4个channel分别表现了数字的某个特征。为了影响学习对于这种表现的影响，我截取了50个iterations之后的情况，情况如下：





从图中可以看出，4张图仍然表现4个不同的特征，且表现的特征与前面的4个相同，说明学习并不好改变每一个channel表现的特征，只会让他们判断的更加准确。

四、总结

本次作业的难度较上一次的作业有所提高，所花费的时间也更多。但是可以在作业当中学到许多，也更加熟悉了CNN的结构与原理。期待下一次的作业。