# Deep-Learning实验报告

## （CNN 识别MNIST）实验

日期：2017.7.5

学号：24320142202455

姓名：林金鹏

## 实验目标：

CNN的简单使用

## 实验内容：

利用CNN进行MNIST数据集的识别，并进行对照实验和优化。

## 实验相关知识：

* 卷积

首先，卷积就是进行特征提取，每个神经元的输入与前一层的局部接受域相连，并提取该局部的特征。

然后，通过权值共享，对于这个图像上的所有位置，我们都能使用同样的学习特征。实现了特征在图片上的位置无关性。

然后，通过增加卷积核来提取不同的特征值

使用卷积层可以极大的减少参数的数目，使学习的问题变得容易。

* 池化

保持 invariance(不变性)，让神经网络更关注是否存在某些特征而不是特征具体的在图片中的位置。

保留主要的特征同时减少参数(降维，效果类似PCA)和计算量，防止过拟合，提高模型泛化能力。

池化有L1池化（最大值池化），和L2 池化（平均值池化）。

* 在全连接层弃权（dropout）

Dropout 是为了减少隐藏层中的由于权值过多而出现的过度拟合，每次训练通过随机丢弃一些激活的权值，可以让神经网络摆脱对某些权值的过度依赖，抑制多度拟合的趋势。

卷积层有相当⼤的先天的对于过度拟合的抵抗。所以不在卷积层进行弃权。

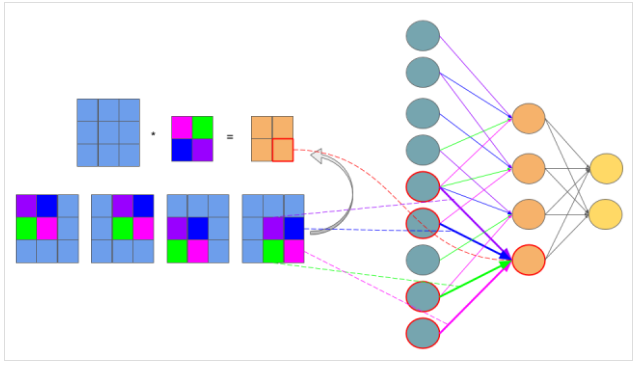
原因是共享权重，意味着卷积滤波器被强制从整个图像中学习。这使他们不太可能去选择在训练数据中的局部特质。于是就很少有必要来应⽤其它规范化，例如弃权。

* 修正线性神经元(ReLu)

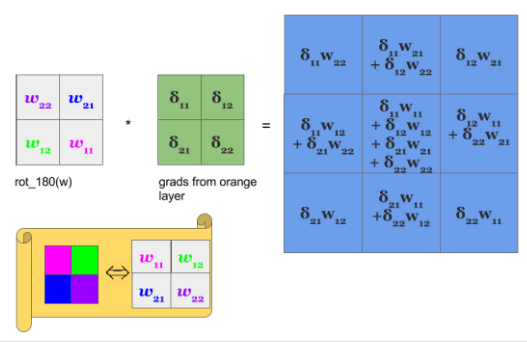
使用Relu 作为激活函数，而不是使用 sigmoid ，通常性能更好，而且可以加快训练的速度3-5倍。

* 卷积 中的反向传播

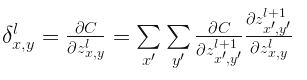
网址：https://grzegorzgwardys.wordpress.com/2016/04/22/8/



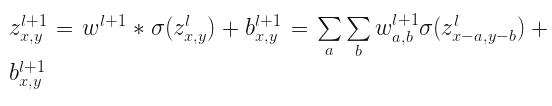
上图中，对3x3 的矩阵进行卷积，卷积用的过滤器 是2x2 的过滤器，步长为1，结果是将3x3 的矩阵卷积成 2x2 的矩阵。权值总共有2x2 = 4 个，相同颜色的块和线代表共享的权值。跟全链接的区别是没有了全链接，而且权值进行了共享。



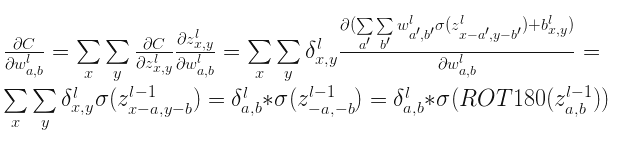
可以看出，grads 可以看成下一层的偏导数值，即：



前向传播的过程如下（l+1 层值与 l 层的关系）：

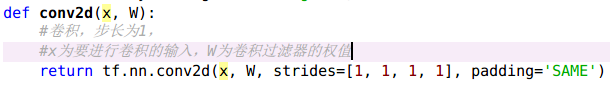


利用链式法则进行反向求导：

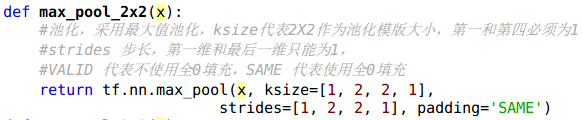


## 关键代码：

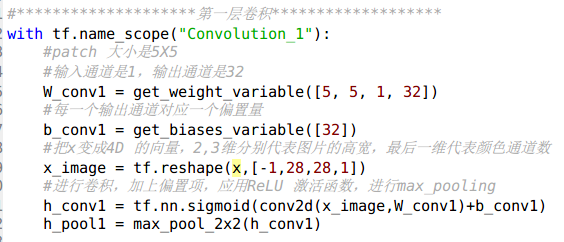
Tensorflow 卷积函数



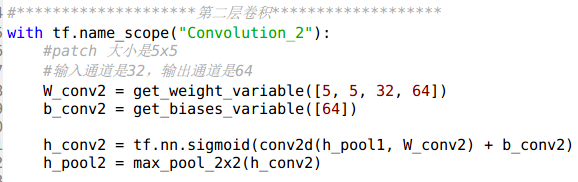
Tensorflow 池化函数



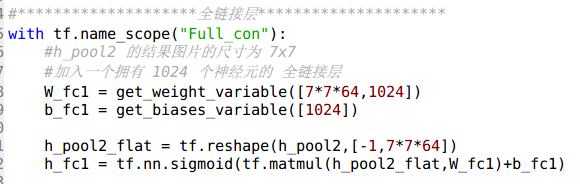
第一层卷积池化



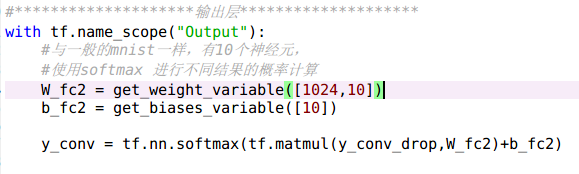
第二层卷积池化



全连接层：

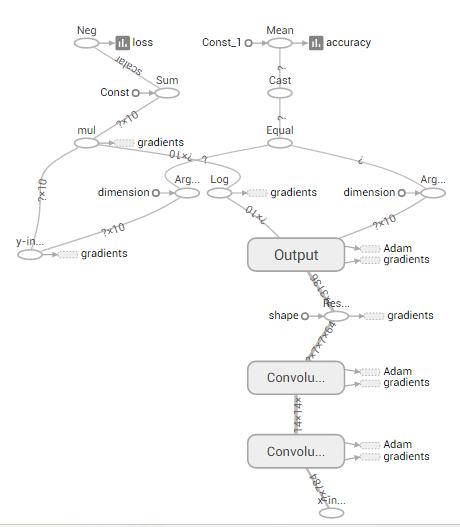


输出层



## 实验结果分析：

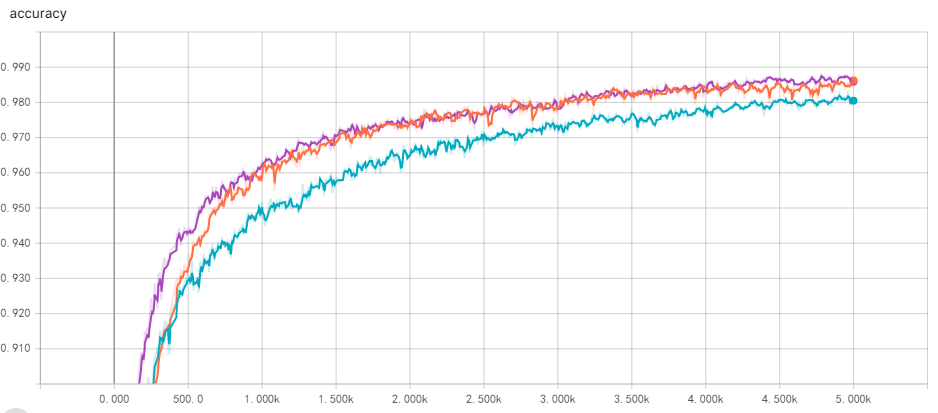
整个计算图的框架：

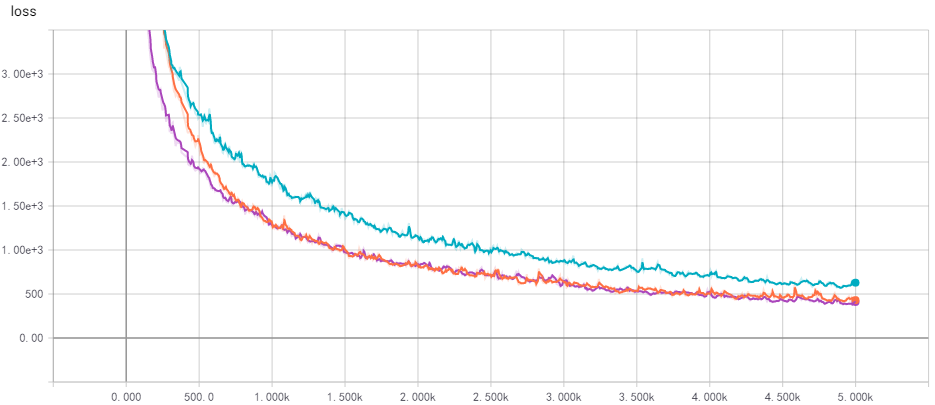


不同的卷积池化层进行对比：

1层卷积池化（蓝色），2层卷积池化（紫色），3层卷积池化(橙色)

（上图，准确率，下图，损失值）



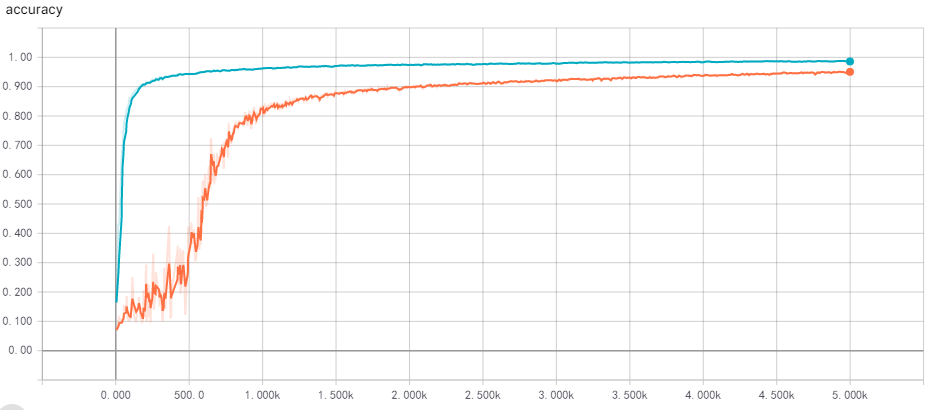


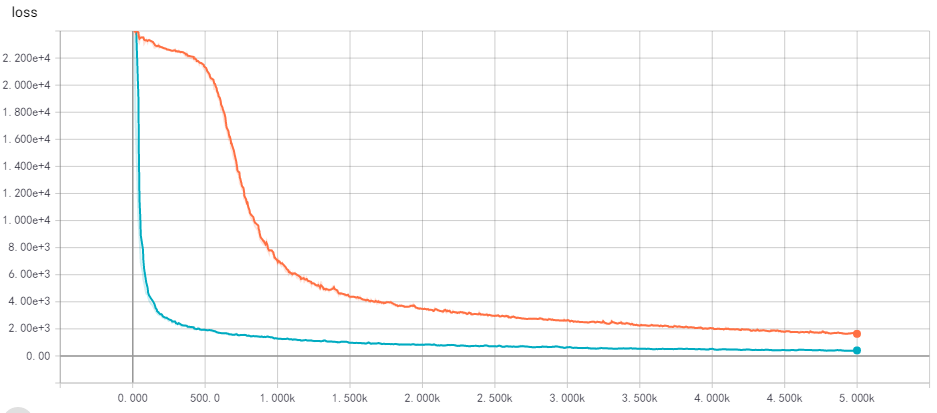
* 可以看出，只有一层卷积池化效果并不够好，而两层的卷积池化效果是最好的，超过两层后，学习会变得比较缓慢。

不同激活函数进行对比：

激活函数sigmoid（橙色），激活函数RELU（蓝色）

（上图，准确率，下图，损失值）



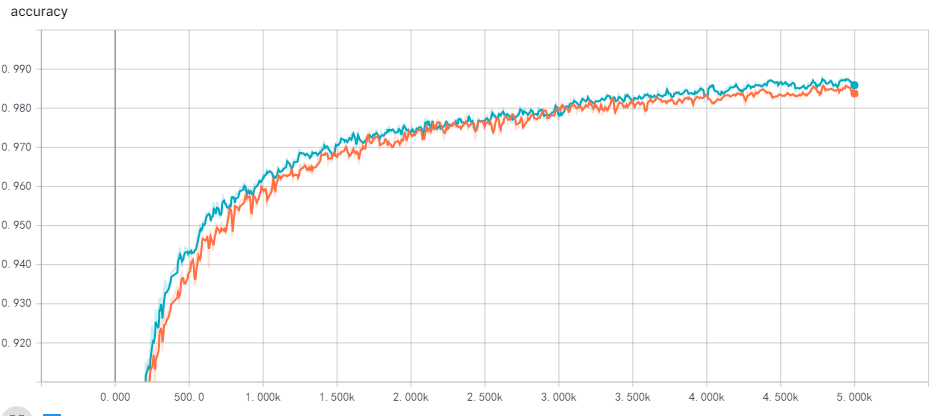


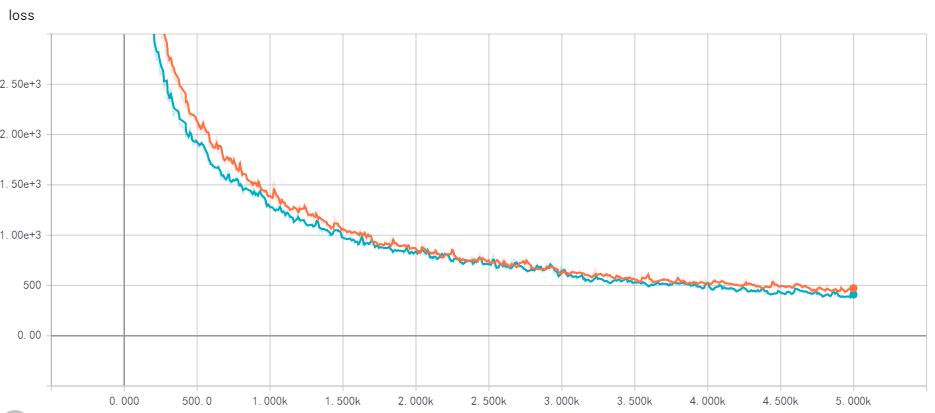
* 可以看出使用relu 激活函数学习速度明显比较快。使用sigmoid由于是s型的虚线，在开始时经历过一段较慢的学习过程，中间较快，最后再趋于平缓。

不同的池化方式进行对比：

平均池化avg\_pool（橙色），最大值池化max\_pool（蓝色）

（上图，准确率，下图，损失值）

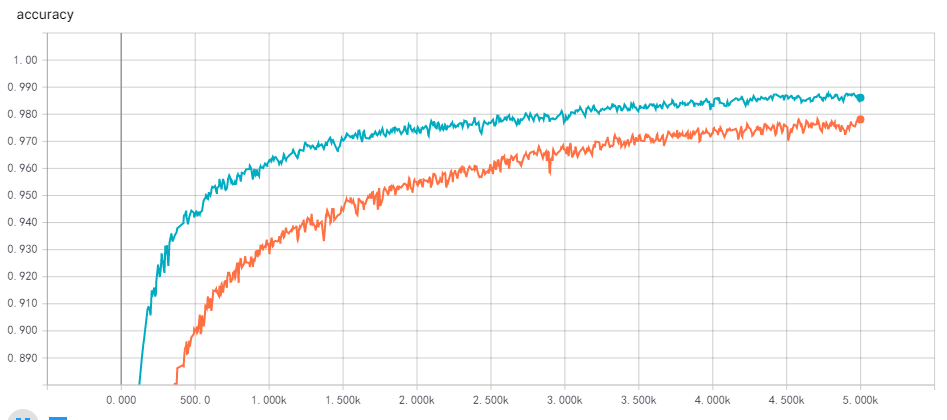


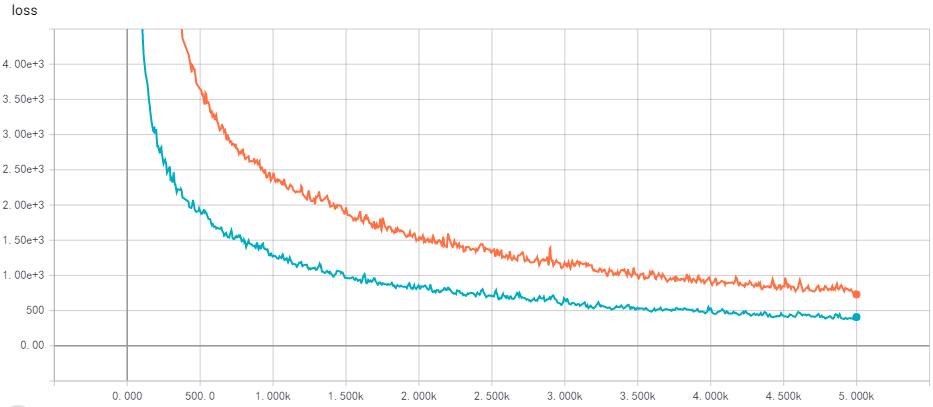


* 平均池化和最大值池化的差别不大，可以看出，最大值池化稍微优于平均值池化。

有无全连接层进行准确率和损失对比：

没有连接层（橙色），有连接层（蓝色）（上图，准确率，下图，损失值）

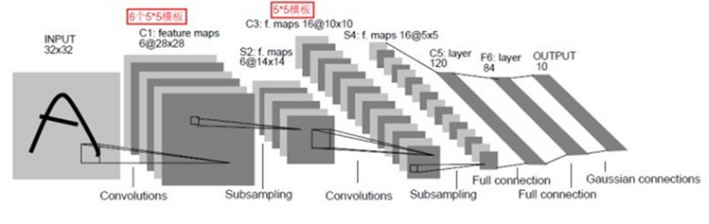




* 可以发现，没有全连接的训练速度比较慢，在开始时与有全连接差距较大。但随着训练的增加，差距在减小，最后的效果大约小于有全链接1%。可知，加入全连接让训练效果更好。
* 另一个方面，有全链接的网络训练相对较慢，为25min,而没有全链接为 22min。

## 实验总结：

CNN卷积网络可以大幅度提高MNIST的识别准确率到98%左右，并且通过使用RELU激活函数、在卷积池化后加入一个全链接层、使用合适的卷积层数、max\_pool池化、选择合适的卷积池化层数等手段可以优化整个cnn 卷积网络的训练速度和最终识别效果。



网上说用LeNet架构可以实现MNIST识别率达到99%，试了一下，用tensorflow 实现相同的Lenet架构，最大的准确率也没有超过99%，可能是需要其他的优化的手段才能更进一步达到99%以上的准确率。

经过改进，第一层卷积用了32个卷积核，第二层卷积用了64个卷积核，第一层全链接用了1024个神经元，第二层全链接用了512个神经元，经过15000次训练，可以达到99.1%的准确率。