人工智能导论大作业 3——手写数字识别 实验报告

引用声明：

本实验报告中的代码参考了如下网址：
https://www.kaggle.com/flaport/tensorflow-cnn-lb-0-98929
“A Convolutional Neural Network for MNIST Classification. This solution got me a score of 0.98929 on the leaderboard.”
(Kaggle 正确率 98.929%)
http://www.tensorfly.cn/tfdoc/tutorials/mnist_pros.html
“MINST 进阶 | TensorFlow 中文社区”
(MINST 正确率 99.2%)
同时，神经网络的架构设计还收了如下网址内容的启发：
https://blog.csdn.net/zchang81/article/details/78593913
“卷积神经网络中 10 大拍案叫绝的操作”
（借鉴了其中的第二条与第三条）
除此之外，我最终提交的代码的具体实现都是由我独立完成的。

模型流程分析：

我实现的模型共有三个版本。
版本 1
基本是对两个代码参考网址中代码的结合与复现。
具体流程是，将 42000 个训练数据分为 32000 的训练集与 10000 的验证集，训练集按照每组 100 张图片输入模型进行训练；
图片按照 5*5 卷积，2*2 最大池化，5*5 卷积，2*2 最大池化的顺序经过两个卷积层，使用 ReLU 作为激发函数；
卷积后的输出流到有 100 个神经元的全连接隐层，经过 50%的 Dropout，再由全连接层按 SoftMax 输出预测的概率；
最后使用 RMSPropOptimizer,通过最小化交叉熵的方式训练模型，得到结果。
最终得到了 97.728%的正确率。
版本 2
在版本 1 的基础上，实现了数据扩增；
通过平移扫描，将一张 28 * 28 的图片变为 25 张 24 * 24 的图片，扩增了数据量；
1050000 个训练数据-》1M 训练集，50K 验证集；
同时把隐层神经元数量提升到了 1000 个。
最终得到了 99.357%的正确率。
版本 3
受引用的第三条网址的启发，修改了模型架构；
尝试在数据卷积前加入一个高保留率（0.9 以上）的 Dropout 以对抗过拟合，最终发现对正确率正面影响不大，但适当调整保留率似乎可以达到加快训练速度的效果（纯体感，很可能是胡诌）；
在原有结构的基础上，让原始图片同时再按照 3*3 卷积，2*2 最大池化，3*3 卷积，2*2 最大池化的顺序经过两个卷积层，然后让两个卷积层的第二层同时输出到一个 256 神经元的全

连接层，再经过 64 神经元的第二个全连接层输出 10 个概率；
最终得到了 99.485%的正确率。

实验结果：

最高正确率 99.485%，提交次数 10（账号昵称：dbmwzhj，2018.5.25 的第 405 名）

参数效果比较：

实验中，我主要调整过的参数是 Dropout 的保留率、卷积核的大小、全连接层神经元的数目和卷积核的个数。
全连接层神经元数目的增加，理论上有利于改善实验结果；但事实上，这一改善的效果并不明显，而且过多的全连接神经元容易引起参数数量的爆炸，因此应该本着够用即可的心态进行调参，卷积核个数同理。
关于卷积核大小的改变，我受引用网址 3 的启发很大。通过使用不同大小的卷积核，我们可以提取出不同尺度的特征，从而提高判别正确率。
关于 Dropout 的保留率以及在卷积层前使用 Dropout 的效果，我有以下想法：其一，在卷积层之前使用 Dropout 确实能达到抗过拟合的效果，减小训练集正确率与验证集正确率的差距；但同时过早的 Dropout 相当于加入噪音，会对模型判断精度的上限有限制。所以我觉得，卷积层前的 Dropout 可以通过按训练时间与效果逐渐提升保留率的方法（如从 0.8 渐渐提到 0.95、0.97）来避免在训练前期由于过拟合拖慢训练时间，发挥其优点、避开其限制；其二，SoftMax 前的 Dropout 优势就非常明显了了，而且保留率应该保持在 0.5 左右。这一层的保留率太高会使得模型产生明显的过拟合，而低于 0.5 的保留率又会拖慢训练速度。

问题思考：

实验训练的停止，我认为还是应该参照验证集正确率以及验证集和训练集正确率之差来动态地决定；前者反映训练进度，后者反映模型过拟合程度。如果在静态条件下（即没有人力根据训练进度及时停止训练），相比较而言，我认为固定迭代次数能比较有效地防止过拟合，而单纯根据验证集正确率是否达到某一值来停止训练，一方面相当于在间接地对验证集做标注，另一方面如果设立的标准过高也必然引起过拟合。
关于参数的初始化，各 W 参数基本以正态分布初始化，而各 B 参数（即偏置）基本按 0 初始化；正态分布的 W 参数有利于使不同的卷积核学习到不同的特征，而偏置的初值按逻辑应该从零开始，除非实验证明以某一常数开始能加快训练速度。
关于防止过拟合的方法，Dropout 和验证集都很重要；Dropout 的效果我在参数分析里已经写过，而验证集的使用是检验模型泛化能力的直接方法。
关于 CNN 相对全连接网络的优点，我认为主要有二：其一，也就是众所周知的减小参数个数；其二，是可以通过卷积核来提取出局部特征，由局部再到整体。

心得体会：

体验了三个晚上从只会用 python 自带的 IDLE 到 99.4%正确率的过程……
主要的感想有以下几点：

一、调库很重要，我自己造轮子三个礼拜都不一定做得完这个实验；
二、深度学习的调参过程确实有很强的工程因素，需要不断的实验来熟悉其架构以寻找灵感；
三、沟通和交流能让实验事半功倍。我的引用网址 3 里的很多其它结构也完全都可以试着应用在这个问题上，还完全没到需要自己设计结构提升效果的程度。