

Deep Learning Homework 3



安捷 1601210097

2017 年 4 月 9 日

1 算法实现简介

在这次作业中，我基于上一次作业实现的 CNN 模型，给 CNN 中添加了 batch normalization 层；同时，为了使得算法的代码更为清楚，我对上一次实现的代码进行了重构。

2 算法实现的函数功能简介

函数名称	功能
conv_2d	卷积层线性单元
max_pool	最大池化层
evaluation	计算准确率函数
batch_norm	batch norm 层
conv_net	卷积网络演化层

表 1: 算法实现的函数功能表

3 数值实验结果

我测试了增加 batch normalization 层的 cnn 的训练情况，我发现，在添加 batch normalization 之后，在使用 Adam 算法，学习率设置为 0.01 的情况下，经过 10000 次迭代，算法在训练集达到了 98.7% 的准确率，性能远远不及不使用 batch normalization 的情况，对比如下图：上表的结果令我比较惊讶，并且我发现，由于添加 bn 层之后每一次训练的计

优化算法	学习率	迭代次数	测试集准确率
Adam	0.001	2500	99%
Adam	0.01	10000	98.7%

表 2: 算法性能对比

算量加大，每一次训练的时间变长。在这个数值试验中，bn 层主要起到了学习率去单位化的作用，增加了训练时的稳定性，即学习率可以设置为原来的 10 倍算法依然可以收敛，但是没有起到加速训练的作用。

4 代码运行环境及测试平台信息

在没有 NVIDIA GPU 及 CUDA 支持的环境下代码依然可以运行，只是速度较慢

Python Version: 3.6.0
Tensorflow Version: tensorflow-gpu-1.0.1
CUDA Version: 8.0
OS: Arch Linux
Kernel: x86_64 Linux 4.10.4-1-ARCH
CPU: Intel Core i7-6700K @ 8x 4.2GHz
GPU: GeForce GTX 1060 6GB
RAM: 16003MiB

表 3: 代码运行环境及测试环境表

5 总结

通过这次作业，我学习了 tensorflow 实现 cnn with bn 的基本方法，了解了 bn 的原理和提高算法训练稳定性的作用，但是由于我在数值试验中没有观察到 bn 提高了算法训练的速度，因此对于 bn 可以提高网络的训练性能依然存在疑问。