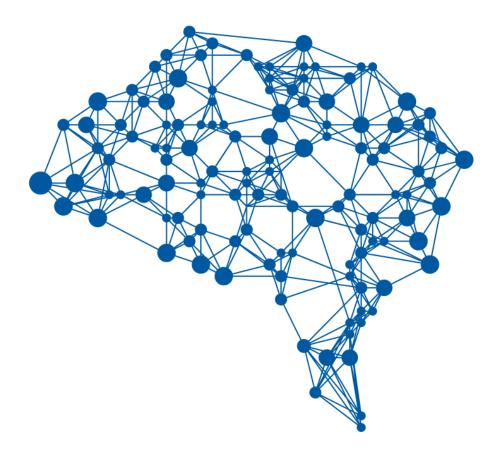
Deep Learning Homework 2



安捷 1601210097 2017 年 3 月 28 日

1 算法实现简介

在这次作业中,为了能够更好的实验不同优化算法的性能及对参数的敏感性,我放弃了上次作业中使用的 MLP 网络,重新实现了一个有两个卷积层和一个全连接层的 CNN,并针对 mnist 数据集测试了不同优化算法的性能,性能测试主要分为两部分:

- 1. 使用每个算法默认的学习率及其他超参数,分别测试算法达到 99% 测试准确率的迭代次数,以此来显示算法的性能;
- 2. 对每一个算法,分别使用默认学习率或超参数、比默认参数高一个数量级、低一个数量级三组参数分别测试其在上一步达到 99% 准确率的迭代次数时达到了准确率,以此作为衡量算法对学习率或超参数敏感性的指标;

2 算法性能测试

表 1: 优化算法达到 99% 准确率所需迭代次数

3 算法超参数稳定性测试

表 2: 算法不同超参数下固定迭代次数达到的准确率

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

5 总结

通过这次作业,我学习了 tensorflow 的基本使用方法,在实现了 MLP 之后,我又实现了 CNN 并成功部署到了科研实验中,接下来我将尝试用 tensorflow 实现 FCN 用于图像分割;在完成作业的过程中,我同时利用上述测试环境中的 CPU 与 GPU 进行了训练,发现 GPU 训练的速度大约是 CPU 训练速度的两倍,之后的作业在显存足够的情况下我将主要使用 GPU 进行训练。

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