

CNN搭建

数据读入

AlexNet

训练方式

可视化

Tensorboard

结果分析

待解决项

AlexNet作为CNN的一类，本文是结合Tensorflow在oxflowers数据上对其的实现。

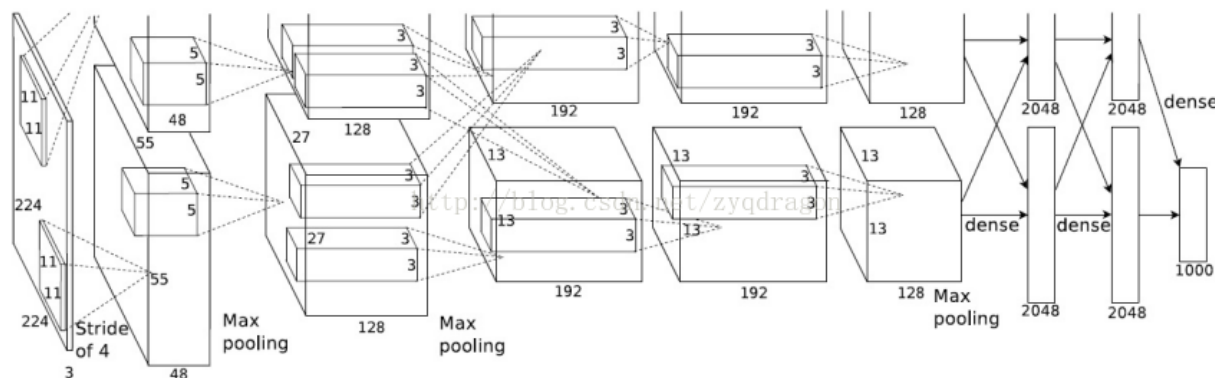
CNN搭建

数据读入

本部分目的是实现oxflowers的数据读入，并分割为训练集和测试集。从如下[数据连接](#)下载源数据并阅读数据说明。数据集包含1360张花朵的图片，按顺序每80个为一类，总计17类。首先要对数据集分割并加上标签，之后用PIL包的Image函数将图片读为张量，根据数据集中的datasplits.mat将数据集分割为训练集和测试集，最后定义shuffle的规则方便分批训练。代码以及详尽的注释在[LoadData.py](#)

AlexNet

本部分着重介绍AlexNet网络的结构。AlexNet的详细分析可以参见这篇[博客](#)，作为卷积神经网络的一种特殊形式，它首先用三层“卷积—ReLU—池化—LRN”的结构提取特征，其中Local Response Normalization (LRN)是[AlexNet](#)2012年首次提出来的，然后再搭建三个全连接层得到输出，具体结构见下图。



注意原始网络的最终输出是1000维的one-hot向量，对应着数据集是1000个标签，且网络在提取特征的时候都是分割成两两部分便于在两个GPU上计算，本文区别于原始网络的点在于最终输出是17维的one-hot且没有分割成两部分进行计算。

本网络用到了CNN中的stride, padding, pooling, dropout等概念，此外一个技巧是在训练MLP加入L2惩罚有助于提高精度，参见Tensorlayer的[说明文档](#)，网络部分的代码见[AlexNet.py](#).

训练方式

网络的训练就是分成batch训练，比较复杂的是加了很多summary以方便可视化，代码见[train.py](#).

可视化

Tensorboard

Tensorboard的坑是始料未及的。基本的操作是在代码中加入summary，并tf.summary.FileWriter下来，这些网上的教程都有提到，但有两点没提到的需要注意：

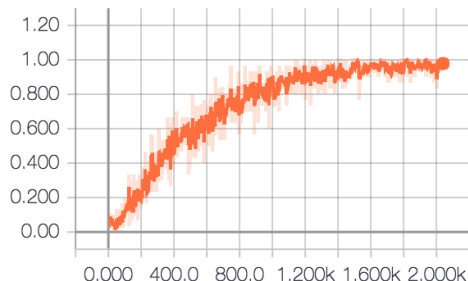
1. 现在不会自动弹出网页地址，需要自行在Chrome浏览器中输入<http://localhost:6006>。
2. 指令为在命令行中输入tensorboard --logdir [总结文件events的上层文件夹地址]。但是却一直报错，而错误的原因最后发现是自己的电脑名字取得不好，导致events的后缀名是".alpoise's_PC "可能是单引号为汉字字符，也可能单引号本身就与字符串矛盾，总归把电脑改名为dormpc后顺利运行。

结果分析

我们训练了80个epoch，每30步在测试集上进行一次测试。由于是在本人的台式机上使用GPU GTX970训练的，所以80个epoch大致只需要15分钟。首先来看训练集上的表现：

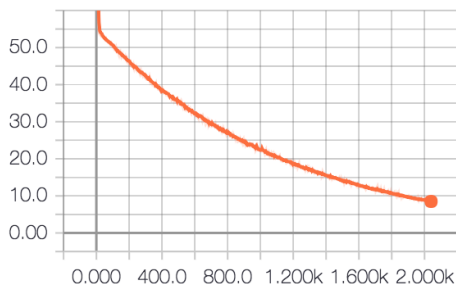
accuracy_1

accuracy_1



loss_1

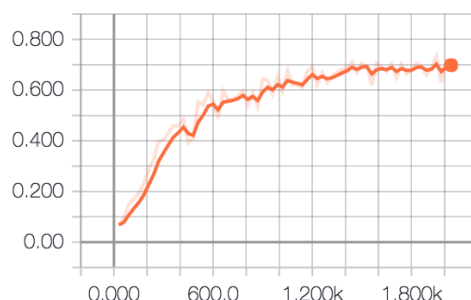
loss_1



从loss的下降可以看到Adam优化器在本问题是十分有效的，但是样本内的准确度在1200步之后就几乎为1了，这很可能意味着过拟合。我们接着来看在测试集上的表现：

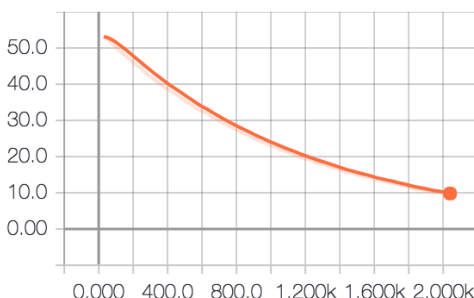
accuracy_1

accuracy_1



loss_1

loss_1



测试集上最终准确率只有70%左右，考虑到样本集容量非常有限，所以这个精度还是不错的。如果采用预训练给出的参数对模型初始化可以进一步提高预测力，但已经与本实验初衷有些远，暂且不表。此外，在实验过程中也有尝试对网络结构进行一些修改，如全连接的最后一层节点数是从1000到17等，效果并没有改观作为中间过程也就没有记录。

待解决项

变量总结的INFO对变量b3:0让修改为b3_0，summary name is illegal;

1. 对TF的变量名管理和tf.summary还不算会，还是照猫画虎的阶段;
2. 在尝试把dense层也用函数封装后，再调用变量L2_loss时没效果，猜测还是变量调用的问题;
3. debug时修改图全局变量的方式，能节约loaddata的时间。

综上，下次作业之前要搞懂tf.Variable和tf.summary