



利用caffe与Imdb读写图像数据

由于有关caffe的开发资料实在太少，单单是这个问题就困扰了我半天。最后终于找到了一个大腿—[beenfrog](#)，也是一个正在学习caffe框架的研究人员。博客mark下，以后有问题可以去这里找。

本文代码部分主要参考于[此文](#)。

简述

Imdb是一种轻量级的数据库，caffe中主要就是使用Imdb模块来进行图像数据集的保存。据说是因为Imdb有读取速度快，支持多线程、多进程并发，等这样那样的优点（具体见[官网](#)，虽然我暂时没有看出来，据我所知网上查找Imdb文档的人大都仅仅是为了使用caffe的），注意到这个数据库其实并没有任何压缩处理的作用，他的目的只是为了快速的索引和存取。他的数据都会带着一定的数据结构从而使的体积略微增大。

事实上如果仅仅看Imdb的用法是无法直接应用于图像文件的处理的。由于caffe是将图像以他自带的数据类型的形式传入Imdb中的，因此我们必须结合caffe的数据类型才能完成读取和使用。

lmdb的安装

参考官方文档，安装如下依赖即可：

```
1 sudo apt-get install cython
2 sudo apt-get install libffi-dev python-dev build-essential
3 sudo apt-get install python-cffi
4 sudo easy_install lmdb
```

实践再次印证了用easy_install 安装要比pip好的多。

生成数据文件

这里以zhengfang/文件夹下的1.png至1000.png为例

```
1 #coding:utf-8
2 import lmdb
3 import numpy as np
4 import cv2
5 import caffe
6 from caffe.proto import caffe_pb2
7
8 #basic setting
9 lmdb_file = 'lmdb_data'#期望生成的数据文件
10 batch_size = 200      #lmdb对于数据进行的是先缓存后一次性写入从而提高效率，因此定义一个batch_size控制每次写入的量。
11
12 # create the leveldb file
13 lmdb_env = lmdb.open(lmdb_file, map_size=int(1e12))#生成一个数据文件，定义最大空间
14 lmdb_txn = lmdb_env.begin(write=True)           #打开数据库的句柄
15 datum = caffe_pb2.Datum()                       #这是caffe中定义数据的重要类型
16
17 for x in range(1000):
18     x+=1
19     img=cv2.imread('zhengfang/'+str(x)+'.png').convert('RGB')    #从zhengfan
20     g/文件夹中依次读取图像
21
22     # save in datum
23     data = img.astype('int').transpose(2,0,1)    #图像矩阵，注意需要调节维度
24     #data = np.array([img.convert('L').astype('int')]) #或者这样增加维度
25     label = x                                     #图像的标签，为了方便存储，这个
26     必须是整数。
27     datum = caffe.io.array_to_datum(data, label)  #将数据以及标签整合为一个数据项
28
29     keystr = '{:0>8d}'.format(x-1)                #lmdb的每一个数据都是由键值对
30     构成的，因此生成一个用递增顺序排列的定长唯一的key
31     lmdb_txn.put( keystr, datum.SerializeToString())#调用句柄，写入内存。
32
33     # write batch
34     if x % batch_size == 0:                        #每当累计到一定的数据量，使用c
35     omit方法写入硬盘。
36     lmdb_txn.commit()
37     lmdb_txn = lmdb_env.begin(write=True)          #commit之后，之前的txn就不能
38     用了，必须重新开一个。
39     print 'batch {} writen'.format(x)
```

```
lmbd_env.close()
的时候打不开。。。
```

#结束后记住释放资源，否则下次用的时候打不开。。。

输出：

```
1 batch 200 writen
2 batch 400 writen
3 batch 600 writen
4 batch 800 writen
5 batch 1000 writen
```

照着注释，很好理解了。值得一提的是，生成的文件是以文件夹的形式保存的，内部由data.mdb lock.mdb组成，不用管他。

这里需要强烈注意的一点是，datum里的图像是需要有长宽高三个维度的，而且第一个维度默认是通道数，而我们平常的图像格式是第三位是通道。因此我们需要在读入普通图像的时候将他转置一下存入datum中，或者在外面增加一层。

读取数据文件

读取上面生成的数据文件。

```
1 #coding:utf-8
2
3 import caffe
4 import lmbd
5 import numpy as np
6 import cv2
7 from caffe.proto import caffe_pb2
8
9 lmbd_env = lmbd.open('lmbd_data') #打开数据文件
10 lmbd_txn = lmbd_env.begin()      #生成句柄
11 lmbd_cursor = lmbd_txn.cursor()  #生成迭代器指针
12 datum = caffe_pb2.Datum()       #caffe定义的数据类型
13
14 for key, value in lmbd_cursor:   #循环获取数据
15     datum.ParseFromString(value) #从value中读取datum数据
16
17     label = datum.label          #获取标签以及图像数据
18     data = caffe.io.datum_to_array(datum)
19     print data.shape
20     print datum.channels
21     image = data.transpose(1,2,0)
22     cv2.imshow('cv2.png', image) #显示
23     cv2.waitKey(0)
24
25 cv2.destroyAllWindows()
26 lmbd_env.close()
```

输出：（图像略）

```
1 (3, 27, 72)
2 3
```

基本是和写入程序一一相对，很好理解。同样需要注意的是如需要显示图像，则需要将数据转置回来。



Myths



2016年3月3日

caffe, Machine Learning, Python

 Previous post

 Next post

发表评论

Your email address will not be published.

Name

Email

...

发表评论

近期文章

- » [python类变量和成员变量的使用注意点](#)
- » [MySQL配置远程访问的巨坑----安全组](#)
- » [Apache配置ssl支持https](#)
- » [Apache配置代理服务器](#)
- » [MySQL python接口更新问题](#)

2016年九月

一	二	三	四	五	六	日
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

« 8月

标签云

