

hjimce的专栏 卧薪尝胆

目录视图

摘要视图

RSS 订阅

个人资料



hjimce



访问：219769次

积分：3638

等级：

排名：第6686名

原创：116篇 转载：1篇

译文：1篇 评论：166条

个人简介

声明:博主的编写,主要参考网上资料,并结合个人见解,仅供学习、交流使用,如有侵权,请联系博主删除,原创文章转载请注明出处。来自福建厦门的一枚小菜鸟.....博主
qq: 1393852684, 微博: http://weibo.com/5372176306
/profile?profile_ftype=1&is_all=1#_0

博客专栏



深度学习

文章：41篇

阅读：137956

文章分类

图像处理 (17)

机器学习 (19)

深度学习 (46)

数据挖掘 (0)

基础知识 (21)

图形处理 (13)

自然语言 (0)

文章存档

[移动信息安全的漏洞和逆向原理](#) [【观点】世界上最好的语言是什么](#) [58到家周俊鹏：webpack PK fis，我更喜欢前者](#)

深度学习（十三）caffe之训练数据格式

2015-10-19 12:02

5841人阅读

评论(1)

收藏

举报

本文章已收录于：



深度学习知识库

分类：深度学习（45）

版权声明：本文为博主原创文章，欢迎转载，转载请注明出处、作者信息。

caffe之训练数据格式

原文地址：<http://blog.csdn.net/hjimce/article/details/49248231>

作者：hjimce

caffe对于训练数据格式，支持：lmdb、h5py.....，其中lmdb数据格式常用于单标签数据，像分类等，经常使用lmdb的数据格式。对于回归等问题，或者多标签数据，一般使用h5py数据的格式。当然好像还有其它格式的数据可用，不过我一般使用这两种数据格式，因此本文就主要针对这两种数据格式的制作方法，进行简单讲解。

一、lmdb数据

lmdb用于单标签数据。为了简单起见，我后面通过一个性别分类作为例子，进行相关数据制作讲解。

1、数据准备

首先我们要准备好训练数据，然后新建一个名为train的文件夹和一个val的文件夹：



train文件存放训练数据，val文件存放验证数据。然后我们在train文件下面，把训练数据性别为男、女图片各放在一个文件夹下面：



同样的我们在val文件下面也创建文件夹：

2016年10月 (1)
2016年09月 (2)
2016年07月 (6)
2016年06月 (6)
2016年05月 (5)

展开

阅读排行

hjimce算法类博文目录 (8190)
深度学习 (十八) 基于R (7673)
深度学习 (六) caffe入门 (6289)
深度学习 (四) 卷积神经 (6207)
深度学习 (十) keras学 (6186)
深度学习 (五) caffe环境 (6018)
深度学习 (二) theano环 (5808)
深度学习 (十三) caffe之 (5783)
深度学习 (二十九) Batc (5701)
深度学习 (四十一) cud (5636)

评论排行

图像处理 (十二) 图像融 (18)
深度学习 (二) theano环 (13)
深度学习 (九) caffe预测 (11)
深度学习 (十五) 基于级 (10)
深度学习 (二十九) Batc (8)
深度学习 (十四) 基于C (7)
hjimce算法类博文目录 (6)
深度学习 (十八) 基于R (6)
图像处理 (二) Seam C (5)
深度学习 (三) theano入 (5)

推荐文章

* 程序员10月书讯, 评论得书
* Android中Xposed框架篇---修改系统位置信息实现自身隐藏功能
* Chromium插件 (Plugin) 模块 (Module) 加载过程分析
* Android TV开发总结-构建一个TV app的直播节目实例
* 架构设计: 系统存储-MySQL简单主从方案及暴露的问题

最新评论

深度学习 (十) keras学习笔记
Houyuzizi: 这个运行代码的环境是什么? IDLE吗? python的编程环境是什么? 谢谢
深度学习 (二十八) 基于多尺度: u011670183: 你好 是不是因为coarse scale的输出, 需要concatenate到fine-scale网络...
基础知识(十三)dlib python人脸检测 ZHAIXINGZHAIYUE: >> faces = detector(rgbimg, 1)参数 1 有什么作用?
深度学习 (十一) RNN入门学习 u014759306: 讲得非常清楚, 但代码看的还是有点吃力, 这个代码是什么功能呢? 请问大神
深度学习 (十) keras学习笔记 qq_18449869: 你好, 请问"keras



两个文件也是分别存我们用于验证的图片数据男女性别文件。我们在test_female下面存放了都是女性的图片, 然后在test_male下面存放的都是验证数据的男性图片。

2、标签文件.txt文件制作。

接着我们需要制作一个train.txt、val.txt文件, 这两个文件分别包含了我们上面的训练数据的图片路径, 以及其对应的标签, 如下所示。

```
second_batch/train_female/985.jpg 1
second_batch/train_female/990.jpg 1
second_batch/train_female/991.jpg 1
second_batch/train_female/992.jpg 1
second_batch/train_female/994.jpg 1
second_batch/train_female/995.jpg 1
second_batch/train_female/997.jpg 1
second_batch/train_male/10.jpg 0
second_batch/train_male/1007.jpg 0
second_batch/train_male/1008.jpg 0
second_batch/train_male/1009.jpg 0
second_batch/train_male/1010.jpg 0
second_batch/train_male/1014.jpg 0
second_batch/train_male/1015.jpg 0
second_batch/train_male/1016.jpg 0
second_batch/train_male/1021.jpg 0
second_batch/train_male/1022.jpg 0
```

我们把女生图片标号为 1, 男生图片标记为 0。标签数据文件txt的生成可以通过如下代码, 通过扫描路径男、女性别下面的图片, 得到标签文件train.txt和val.txt:

```
[python]
01. <span style="font-family:Arial;font-size:18px;"><span style="font-size:18px;"><span style="font-size:18px;">import os
02. import numpy as np
03. from matplotlib import pyplot as plt
04. import cv2
05. import shutil
06.
07.
08. #扫描文件
09. def GetFileList(FindPath,FlagStr=[]):
10.     import os
11.     FileList=[]
12.     FileNames=os.listdir(FindPath)
13.     if len(FileNames)>0:
14.         for fn in FileNames:
15.             if len(FlagStr)>0:
16.                 if IsSubString(FlagStr,fn):
17.                     fullfilename=os.path.join(FindPath,fn)
18.                     FileList.append(fullfilename)
19.             else:
20.                 fullfilename=os.path.join(FindPath,fn)
21.                 FileList.append(fullfilename)
22.
23.
24.     if len(FileList)>0:
25.         FileList.sort()
26.
27.     return FileList
28. def IsSubString(SubStrList,Str):
29.     flag=True
30.     for substr in SubStrList:
31.         if not(substr in Str):
32.             flag=False
33.
34.     return flag
35.
36. txt=open('train.txt','w')
37. #制作标签数据,如果是男的,标签设置为0,如果是女的标签为1
38. imgfile=GetFileList('first_batch/train_female')
39. for img in imgfile:
40.     str=img+'\t'+ '1'+'\n'
41.     txt.writelines(str)
```

也支持保存训练好的参数，然后加载已经训练好的参数，进行继续训练。”是如何实现的...

深度学习（二十八）基于多尺度；
dddd56789: 博主您好，请问paper中的代码是用Python哪个版本写的，是不是直接运行demo_depth.p...

图像处理（十二）图像融合(1)Se
hjimce: @Just_Do_IT_: 到网上找opencv要怎么用，然后找opencv的泊松融合函数要怎...

图像处理（十二）图像融合(1)Se
Just_Do_IT_: 作为一只菜鸟，能问一下这段代码要怎么运行吗？

深度学习（二十九）Batch Norm
dsfour: 很好的一篇博文，赞一个

深度学习（三）theano入门学习
qq_36137078: @qq_35104595: 您好，No module named loaddata这个问题解决了吗？求...

文章搜索



```
42.
43. imgfile=GetFileList('first_batch/train_male')
44. for img in imgfile:
45.     str=img+'\t'+'\0'+'\n'
46.     txt.writelines(str)
47. txt.close()
```

把生成的标签文件，和train\val文件夹放在同一个目录下面：



需要注意，我们标签数据文件里的文件路径和图片的路径要对应的起来，比如val.txt文件的某一行的图片路径，是否在val文件夹下面：

```
test_female/94.jpg 1
test_female/95.jpg 1
test_female/96.jpg 1
test_female/97.jpg 1
test_female/98.jpg 1
test_female/99.jpg 1
test_male/0.jpg 0
test_male/1.jpg 0
test_male/10.jpg 0
test_male/100.jpg 0
test_male/101.jpg 0
test_male/102.jpg 0
test_male/103.jpg 0
test_male/104.jpg 0
test_male/105.jpg 0
```

3、生成lmdb数据

接着我们的目的就是要通过上面的四个文件，把图片的数据和其对应的标签打包起来，打包成lmdb数据格式，打包脚本如下：

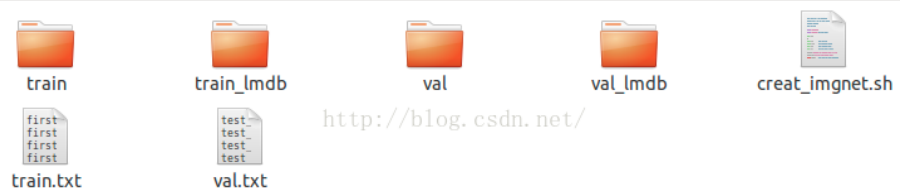
```
[python]
01. <span style="font-family:Arial;font-size:18px;"><span style="font-size:18px;">#!/usr
    /bin/env sh
02. # Create the imagenet lmdb inputs
03. # N.B. set the path to the imagenet train + val data dirs
04.
05. EXAMPLE=. # 生成模型训练数据文件夹
06. TOOLS=../build/tools # caffe的工具库，不用变
07. DATA=. # python脚步处理后数据路径
08.
09. TRAIN_DATA_ROOT=train/ #待处理的训练数据
10. VAL_DATA_ROOT=val/ # 带处理的验证数据
11.
12.
13.
14. # Set RESIZE=true to resize the images to 256x256. Leave as false if images have
15. # already been resized using another tool.
16. RESIZE=true#是否需要图片进行resize
17. if $RESIZE; then
18.     RESIZE_HEIGHT=256
19.     RESIZE_WIDTH=256
20. else
21.     RESIZE_HEIGHT=0
22.     RESIZE_WIDTH=0
23. fi
24.
25. if [ ! -d "$TRAIN_DATA_ROOT" ]; then
26.     echo "Error: TRAIN_DATA_ROOT is not a path to a directory: $TRAIN_DATA_ROOT"
27.     echo "Set the TRAIN_DATA_ROOT variable in create_imagenet.sh to the path" \
28.     "where the ImageNet training data is stored."
29.     exit 1
30. fi
31.
32. if [ ! -d "$VAL_DATA_ROOT" ]; then
33.     echo "Error: VAL_DATA_ROOT is not a path to a directory: $VAL_DATA_ROOT"
34.     echo "Set the VAL_DATA_ROOT variable in create_imagenet.sh to the path" \
```

```

35.         "where the ImageNet validation data is stored."
36.     exit 1
37. fi
38.
39. echo "Creating train lmdb..."
40.
41. GLOG_logtostderr=1 $TOOLS/convert_imageset \
42.     --resize_height=$RESIZE_HEIGHT \
43.     --resize_width=$RESIZE_WIDTH \
44.     --shuffle \
45.     $TRAIN_DATA_ROOT \
46.     $DATA/train.txt \
47.     $EXAMPLE/train_lmdb
48.
49. echo "Creating val lmdb..."
50.
51. GLOG_logtostderr=1 $TOOLS/convert_imageset \
52.     --resize_height=$RESIZE_HEIGHT \
53.     --resize_width=$RESIZE_WIDTH \
54.     --shuffle \
55.     $VAL_DATA_ROOT \
56.     $DATA/val.txt \
57.     $EXAMPLE/val_lmdb
58.
59. echo "Done."

```

通过运行上面的脚本，我们即将得到文件夹train_lmdb\val_lmdb：



我们打开train_lmdb文件夹



并查看一下文件data.mdb数据的大小，如果这个数据包好了我们所有的训练图片数据，查一下这个文件的大小是否符合预期大小，如果文件的大小才几k而已，那么就代表你没有打包成功，估计是因为路径设置错误。我们也可以通过如下的代码读取上面打包好的数据，把图片、和标签打印出来，查看一下，查看lmdb数据请参考下面的代码：

python lmdb数据验证：

```

[python]
01. <span style="font-family:Arial;font-size:18px;"><span style="font-
02. size:18px;"># coding=utf-8
03. caffe_root = '/home/hjimce/caffe/'
04. import sys
05. sys.path.insert(0, caffe_root + 'python')
06. import caffe
07.
08. import os
09. import lmdb
10. import numpy
11. import matplotlib.pyplot as plt
12.
13. def readlmdb(path, visualize = False):
14.     env = lmdb.open(path, readonly=True, lock=False)
15.
16.     datum = caffe.proto.caffe_pb2.Datum()
17.     x=[]
18.     y=[]
19.     with env.begin() as txn:
20.         cur = txn.cursor()
21.         for key, value in cur:

```

```

22.         # 转换为datum
23.         datum.ParseFromString(value)
24.         # 读取datum数据
25.         img_data = numpy.array(bytearray(datum.data))\
26.             .reshape(datum.channels, datum.height, datum.width)
27.         print img_data.shape
28.         x.append(img_data)
29.         y.append(datum.label)
30.         if visualize:
31.             img_data=img_data.transpose([1,2,0])
32.             img_data = img_data[:, :, ::-1]
33.             plt.imshow(img_data)
34.             plt.show()
35.         print datum.label
36.     return x,y

```

通过上面的函数，我们可以是读取相关的lmdb数据文件。

4、制作均值文件。

这个是为了图片归一化而生成的图片平均值文件，把所有的图片相加起来，做平均，具体的脚本如下：

```

[python]
01. #!/usr/bin/env sh
02. # Compute the mean image from the imagenet training lmdb
03. # N.B. this is available in data/ilsrvrc12
04.
05. EXAMPLE=.
06. DATA=train
07. TOOLS=../build/tools
08.
09. $TOOLS/compute_image_mean $EXAMPLE/train_lmdb \      # train_lmdb是我们上面打包好的lmdb数据文
   件
10.     $DATA/imagenet_mean.binaryproto
11.
12. echo "Done."

```

运行这个脚本，我们就可以训练图片均值文件：imagenet_mean.binaryproto

至此，我们得到了三个文件：imagenet_mean.binaryproto、train_lmdb、val_lmdb，这三个文件就是我们最后打包好的数据，这些数据我们即将作为caffe的数据输入数据格式文件，把这三个文件拷贝出来，就可以把原来还没有打包好的数据删了。这三个文件，我们在caffe的网络结构文件，数据层定义输入数据的时候，就会用到了：

```

[python]
01. name: "CaffeNet"
02. layers {
03.   name: "data"
04.   type: DATA
05.   top: "data"
06.   top: "label"
07.   data_param {
08.     source: "train_lmdb" # lmdb格式的训练数据
09.     backend: LMDB
10.     batch_size: 50
11.   }
12.   transform_param {
13.     crop_size: 227
14.     mirror: true
15.     mean_file:"imagenet_mean.binaryproto" # 均值文件
16.   }
17.   include: { phase: TRAIN }
18. }
19. layers {
20.   name: "data"
21.   type: DATA
22.   top: "data"
23.   top: "label"
24. }

```

```

25.     data_param {
26.         source: "val_lmdb" # lmdb格式的验证数据
27.         backend: LMDB
28.         batch_size: 50
29.     }
30.     transform_param {
31.         crop_size: 227
32.         mirror: false
33.         mean_file: "imagenet_mean.binaryproto" # 均值文件
34.     }
35.     include: { phase: TEST }
36. }

```

二、h5py格式数据

上面的lmdb一般用于单标签数据，图片分类的时候，大部分用lmdb格式。然而假设我们要搞的项目是人脸特征点识别，我们要识别出68个人脸特征点，也就是相当于136维的输出向量。网上查了一下，对于caffe多标签输出，需要使用h5py格式的数据，而且使用h5py的数据格式的时候，caffe是不能使用数据扩充进行相关的数据变换的，很是悲剧啊，所以如果caffe使用h5py数据格式的话，需要自己在外部，进行数据扩充，数据归一化等相关的数据预处理操作。

1、h5py数据格式生成

下面演示一下数据h5py数据格式的制作：

```

[python]

01. # coding: utf-8
02. caffe_root = '/home/hjimce/caffe/'
03. import sys
04. sys.path.insert(0, caffe_root + 'python')
05. import os
06. import cv2
07. import numpy as np
08. import h5py
09. from common import shuffle_in_unison_scary, processImage
10. import matplotlib.pyplot as plt
11.
12. def readdata(filepath):
13.     fr=open(filepath, 'r')
14.     filesplit=[]
15.     for line in fr.readlines():
16.         s=line.split()
17.         s[1:]=[float(x) for x in s[1:]]
18.         filesplit.append(s)
19.     fr.close()
20.     return filesplit
21. #因为我们的训练数据可能不是正方形，然而网络的输入的大小是正方形图片，为了避免强制resize引起的图片扭曲，所以我们采用填充的方法
22. def sqrtimg(img):
23.     height,width=img.shape[:2]
24.     maxlenght=max(height,width)
25.     sqrtimg0=np.zeros((maxlenght,maxlenght,3),dtype='uint8')
26.
27.     sqrtimg0[(maxlenght*.5-height*.5):(maxlenght*.5+height*.5),(maxlenght*.5-width*.5):(maxlenght*.5+width*.5)]=img
28.     return sqrtimg0
29.
30.
31. def generate_hdf5():
32.
33.     labelfile =readdata('../data/my_alige_landmark.txt')
34.     F_imgs = []
35.     F_landmarks = []
36.
37.
38.     for i,l in enumerate(labelfile):
39.         imgpath='../data/'+l[0]
40.
41.         img=cv2.imread(imgpath)
42.         maxx=max(img.shape[0],img.shape[1])
43.         img=sqrtimg(img) #把输入图片填充成正方形，因为我们要训练的图片的大小是正方形的图片255*255
44.         img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) #图片转为灰度图像
45.         f_face=cv2.resize(img,(39,39)) #把图片缩放成255*255的图片
46.         # F

```

```

47.         plt.imshow(f_face, cmap='gray')
48.
49.
50.         f_face = f_face.reshape((1, 39, 39))
51.         f_landmark = np.asarray(l[1:], dtype='float')
52.
53.         F_imgs.append(f_face)
54.
55.
56.         #归一化人脸特征点标签, 因为上面height等于width, 这里比较懒, 直接简写
57.         f_landmark = f_landmark / maxx #归一化到0 ~ 1之间
58.         print f_landmark
59.         F_landmarks.append(f_landmark)
60.
61.
62.     F_imgs, F_landmarks = np.asarray(F_imgs), np.asarray(F_landmarks)
63.
64.
65.     F_imgs = processImage(F_imgs) #图片预处理, 包含均值归一化, 方差归一化等
66.     shuffle_in_unison_scary(F_imgs, F_landmarks) #打乱数据
67.
68.     #生成h5py格式
69.     with h5py.File(os.getcwd() + '/train_data.h5', 'w') as f:
70.         f['data'] = F_imgs.astype(np.float32)
71.         f['landmark'] = F_landmarks.astype(np.float32)
72.     #因为caffe的输入h5py不是直接使用上面的数据, 而是需要调用.txt格式的文件
73.     with open(os.getcwd() + '/train.txt', 'w') as f:
74.         f.write(os.getcwd() + '/train_data.h5\n')
75.     print i
76.
77.
78. if __name__ == '__main__':
79.     generate_hdf5()

```

利用上面的代码, 可以生成一个train.txt、train_data.h5的文件, 然后在caffe的prototxt中, 进行训练的时候, 可以用如下的代码, 作为数据层的调用:

```

[python]
01. layer {
02.     name: "hdf5_train_data"
03.     type: "HDF5Data" #需要更改类型
04.     top: "data"
05.     top: "landmark"
06.     include {
07.         phase: TRAIN
08.     }
09.     hdf5_data_param { #这个参数类型h5f5_data_param记得要更改
10.         source: "h5py/train.txt" #上面生成的train.txt文件
11.         batch_size: 64
12.     }
13. }

```

上面需要注意的是, 相比与lmdb的数据格式, 我们需要该动的地方, 我标注的地方就是需要改动的地方, 还有h5py不支持数据变换。

2、h5py数据读取

```

[python]
01. f=h5py.File('../h5py/train.h5', 'r')
02. x=f['data'][:]
03. x=np.asarray(x, dtype='float32')
04. y=f['label'][:]
05. y=np.asarray(y, dtype='float32')
06. print x.shape
07. print y.shape

```

可以通过上面代码, 查看我们生成的.h5格式文件。

在需要注意的是, 我们输入caffe的h5py图片数据为四维矩阵
(number_samples, nchannels, height, width) 的矩阵, 标签矩阵为二维
(number_samples, labels_ndim) ,同时数据的格式需要转成float32, 用于回归任务。

*****作者：hjimce 时间：2015.10.2 联系QQ：1393852684 原创文章，
转载请保留原文地址、作者等信息*****

顶

5

踩

0

上一篇 图像处理（二十三）基于调色板的图像Recoloring-Siggraph 2015

下一篇 深度学习（十四）基于CNN的性别、年龄识别

我的同类文章

深度学习（45）

• 深度学习（四十一）cuda8.0... 2016-07-22 阅读 5683

• 深度学习（四十）caffe使用... 2016-07-11 阅读 1634

• 深度学习（四十）优化求解... 2016-06-26 阅读 552

• 深度学习（三十八）卷积神... 2016-06-26 阅读 1428

• 深度学习（三十七）优化求... 2016-06-05 阅读 755

• 深度学习（四十二）word2v... 2016-06-02 阅读 889

• 深度学习（四十二）网络压... 2016-06-02 阅读 546

• 深度学习（三十六）异构计... 2016-05-26 阅读 869

• 深度学习（三十五）异构计... 2016-05-22 阅读 683

• 深度学习（三十八）初识DL... 2016-05-01 阅读 1760

• 深度学习（三十九）LSTM... 2016-04-24 阅读 1732

猜你在找

- Python脚本应用及学习方法

使用python操作Oracle

Python 项目实战_cmdb Day15

Python淘宝美眉图片下载爬虫视频教程

Python网络爬虫快速入门实战
- 神经网络与深度学习Caffe训练执行时爆出的Check

深度学习实战caffe windows 下训练自己的网络模型

深度学习caffe应用笔记--如何将图片转换为lmdb格式

神经网络与深度学习如何将别人训练好的model用到自

深度学习caffe平台--train_valprototxt文件中数据层及参

出错了！

查看评论

1楼 wangbo792450190 2016-08-18 13:57发表

博主你好，我图像太多（1万张），每张的标签（人数）已经写在excel中，图像名称皆是1.jpg, 2.jpg.....10000.jpg，请问我是否可以在excel中选择两列，一列为图像名称，一列对应每张图的标签（人数），然后把Excel导出TXT文件保存，如此制作是否可行？谢谢，期待回复。

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

