余昌黔|书山有路

ycszen

关注 发私信

TensorFlow高效读取数据的方法

标签： tensorflow 深度学习

2016-08-17 19:20 25120人阅读 评论(33) 收

分类：

深度学习（6） tensorflow

目录(?) [+]

关闭

概述

最新上传的mcnn中有完整的数据读写示例，可以参考。

关于Tensorflow读取数据，官网给出了三种方法：

- **供给数据(Feeding)**： 在TensorFlow程序运行的每一步， 让Python代码来供给数据。
- **从文件读取数据**： 在TensorFlow图的起始， 让一个输入管线从文件中读取数据。
- **预加载数据**： 在TensorFlow图中定义常量或变量来保存所有数据(仅适用于数据量比较小的情况)。

对于数据量较小而言，可能一般选择直接将数据加载进内存，然后再分 batch 输入网络进行训练（tip:使用这种方法时，结合 yield 使用更为简洁，大家自己尝试一下吧，我就不赘述了）。但是，如果数据量较大，这样的方法就不适用了，因为太耗内存，所以这时最好使用tensorflow提供的队列 queue，也就是第二种方法 **从文件读取数据**。对于一些特定的读取，比如csv文件格式，官网有相关的描述介绍一种比较通用，高效的读取方法（官网介绍的少），即使用tensorflow内定标准格式——

太长不看，直接看源码请猛戳我的github，记得加星哦。

TFRecords

关闭

TFRecords其实是一种二进制文件，虽然它不如其他格式好理解，但是它能更好的利用内存，更方便复制和移动，并且不需要单独的标签文件（等会儿就知道为什么了）… …总而言之，这样的文件格式好处多多，所以让我们用起来吧。

TFRecords文件包含了 tf.train.Example 协议内存块(protocol buffer)(协议内存块包含了字段 Features)。我们可以写一段代码获取你的数据， 将数据填入到 Example 协议内存块(protocol buffer)，将协议内存块序

列化为一个字符串， 并且通过 tf.python_io.TFRecordWriter 写入到TFRecords文件。

从TFRecords文件中读取数据， 可以使用 tf.TFRecordReader 的 tf.parse_single_example 解析器。这个操作可以将 Example 协议内存块(protocol buffer)解析为张量。

接下来，让我们开始读取数据之旅吧~

生成TFRecords文件

我们使用 tf.train.Example 来定义我们要填入的数据格式，然后使用 tf.python_io.TFRecordWr

```
import os
import tensorflow as tf
from PIL import Image

cwd = os.getcwd()

'''
此处我加载的数据目录如下：
0 -- img1.jpg
     img2.jpg
     img3.jpg
     ...
1 -- img1.jpg
     img2.jpg
     ...
2 -- ...
 这里的0， 1， 2...就是类别，也就是下文中的classes
 classes是我根据自己数据类型定义的一个列表，大家可以根据自己的数据情况灵活运用
```

```
...
'''
writer = tf.python_io.TFRecordWriter("train.tfrecords")
for index, name in enumerate(classes):
    class_path = cwd + name + "/"
    for img_name in os.listdir(class_path):
        img_path = class_path + img_name
            img = Image.open(img_path)
            img = img.resize((224, 224))
        img_raw = img.tobytes()              #将图片转化为原生bytes
        example = tf.train.Example(features=tf.train.Features(feature={
            "label": tf.train.Feature(int64_list=tf.train.Int64List(value=[index])),
            'img_raw': tf.train.Feature(bytes_list=tf.train.BytesList(value=[img_raw]))
        }))
        writer.write(example.SerializeToString())  #序列化为字符串
writer.close()
```

关于 Example Feature 的相关定义和详细内容，我推荐去官网查看相关API。

基本的，一个 Example 中包含 Features ， Features 里包含 Feature （这里没s）的字典。最后， Feature 里包含有一个 FloatList ， 或者 ByteList ，或者 Int64List

关闭

就这样，我们把相关的信息都存到了一个文件中，所以前面才说不用单独的label文件。而且读取也很方便。

接下来是一个简单的读取小例子：

```
for serialized_example in tf.python_io.tf_record_iterator("train.tfrecords"):
    example = tf.train.Example()
    example.ParseFromString(serialized_example)

```

```
    image = example.features.feature['image'].bytes_list.value
    label = example.features.feature['label'].int64_list.value
    # 可以做一些预处理之类的
    print image, label
```

使用队列读取

一旦生成了TFRecords文件，为了高效地读取数据，TF中使用队列（ queue ）读取数据。

```
def read_and_decode(filename):
    #根据文件名生成一个队列
    filename_queue = tf.train.string_input_producer([filename])

    reader = tf.TFRecordReader()
    _, serialized_example = reader.read(filename_queue)   #返回文件名和文件
    features = tf.parse_single_example(serialized_example,
                                       features={
                                           'label': tf.FixedLenFeature([], tf.int64),
                                           'img_raw' : tf.FixedLenFeature([],
                                       })

    img = tf.decode_raw(features['img_raw'], tf.uint8)
    img = tf.reshape(img, [224, 224, 3])
    img = tf.cast(img, tf.float32) * (1. / 255) - 0.5
    label = tf.cast(features['label'], tf.int32)

    return img, label
```

之后我们可以在训练的时候这样使用

关闭

```
img, label = read_and_decode("train.tfrecords")

#使用shuffle_batch可以随机打乱输入
img_batch, label_batch = tf.train.shuffle_batch([img, label],
                                                batch_size=30, capacity=2000,
                                                min_after_dequeue=1000)
init = tf.initialize_all_variables()

with tf.Session() as sess:
    sess.run(init)
    threads = tf.train.start_queue_runners(sess=sess)
    for i in range(3):
        val, l= sess.run([img_batch, label_batch])
        #我们也可以根据需要对val， l进行处理
        #l = to_categorical(l, 12)
        print(val.shape, l)
```

至此，tensorflow高效从文件读取数据差不多完结了。

恩？等等...什么叫差不多？对了，还有几个**注意事项**：

关闭

第一，tensorflow里的graph能够记住状态（ state ），这使得 TFRecordReader 能够记住 tfrecord 的位置，并且始终能返回下一个。而这就要求我们在使用之前，必须初始化整个graph，这里我们使用了函数 tf.initialize_all_variables() 来进行初始化。

第二，tensorflow中的队列和普通的队列差不多，不过它里面的 operation 和 tensor 都是符号型的（ symbolic ），在调用 sess.run() 时才执行。

第三， TFRecordReader 会一直弹出队列中文件的名字，直到队列为空。

总结

1. 生成tfrecord文件
2. 定义 record reader 解析tfrecord文件
3. 构造一个批生成器（ batcher ）
4. 构建其他的操作
5. 初始化所有的操作
6. 启动 QueueRunner

例子代码请戳我的github，如果觉得对你有帮助的话可以加个星哦。

- 上一篇 深度学习框架Torch7解析-- Tensor篇
- 下一篇 深度学习最全优化方法总结比较（SGD，Adagrad，Adadelta，Adam，Adamax，Nadam）

- tensorflow载入数据的三种方式
- 【免费】深入理解Docker内部原理及网络配置--王…
- 深度学习（五十六）tensorflow项目构建流程
- SDCC 2017之区块链技术实战线上峰会--蔡栋
- Tensor Flow shuffle_batch 的方式读csv文件的例子
- php零基础到项目实战
- tensorflow加载数据的几种方式
- C语言及程序设计入门指导
- TensorFlow读取二进制文件数据到队列
- Android入门实战
- tensorflow的数据输入
- 5天搞定深度学习框架Caffe
- 从原理到代码：大牛教你如何用 TensorFlow 亲手搭..
- Tensorflow从文件读取数据
- 图片存储为cifar的Python数据格式
- TensorFlow高效读取数据的方法

查看评论

Chromer163 13楼 2017-08-02 22:1

博主你好，我有一个问题想请教一下，就是你的代码里面的class是你自己的图片的类别，那么我的数据集是lfpw，里面是一个人脸特征点定位的数据集，每张图片的标定数据是68个坐标，那么这种情况下，我的label应该如何设置？

关闭

learningJavachuxue 12楼 2017-07-11 17:24发表

楼主执行这段代码 val, l= sess.run([img_batch, label_batch])阻塞了，是怎么回事

智障儿童欢乐多A 11楼 2017-06-30 11:40发表

楼主请问你的数据类别class是如何存储的呢？

mu0_0mu 10楼 2017-04-07 19:51发表

你好，请问数据类型的列表是怎么定义的？？

xwdkobe 9楼 2017-04-05 15:49发表

楼主，看了你的代码，能问一下你扣扣，加你扣扣请教一下吗？谢谢了

谁主沉浮---data 8楼 2017-03-23 11:10发表

for serialized_example in tf.python_io.tf_record_iterator("train.tfrecords"):
example = tf.train.Example()
example.ParseFromString(serialized_example)

image = example.features.feature['image'].bytes_list.value
label = example.features.feature['label'].int64_list.value
# 可以做一些预处理之类的
print image, label

楼主之前已经定义write来写入TFRecord了，而且用tf.train.Example构造了数据结构，那么这段是来做什么的？

关闭

ycszen Re: 2017-03-23 22:37发表

回复谁主沉浮---data：后面这一段是一个小小的读取的例子，用来验证一下写入成功没有以及看看tfrecords里的结构。

完美妖姬 7楼 2017-02-28 18:43发表

for index, name in enumerate(classes):
楼主，这里的classes指的是什么

---

imageprocessin Re: 2017-03-20 17:04发表

回复完美妖姬：classes 怎么定义？？

完美妖姬 Re: 2017-03-20 21:37发表

回复imageprocessin：看你的代码，应该指的是你的数据，你是用字典存储的数据吧，因为classes在之前没有明确给出，所以有点迷惑。不过现在看，已经没有障碍了~

jsjs0827 Re: 2017-03-20 21

回复imageprocessin：我也是卡这里了 classes

ycszen Re: 2017-03-23 22:3

回复jsjs0827：classes是指自己数据的类别

关闭

jsjs0827

回复imageprocessin：我也是卡这里了 classes

狐狸117 6楼 2017-02-12 23:16发表

十分感谢博主，我是从tensorlayer的tutorials中看到的中文推荐是你的博客，看来源码感觉很精炼，感谢感谢。

ycszen Re: 2017-03-23 22:32发表

回复狐狸117：谢谢

茁壮小草 5楼 2017-01-22 14:58发表

你好，你能后对你的这篇博客更新下呢？tensorflow更新后，蛮多的函数变化了，还有是没有具体的例子，理解有点困难啊！谢谢！

ycszen Re: 2017-03-23 22:32发表

回复茁壮小草：给出了新的例子

codingShip 4楼 2017-01-05 13

我想请问下，这种实现方式只能对数据进行一次使用，队列空了后就没法再次使用了。但是在很多场景下，例如train.data数据是需要多次使用的，这样的话岂不是每次都得读入队列？

ycszen Re: 2017-03-23 22:3

回复codingShip：数据都是批量读入队列的，队列空了就会读取下一批数据。这样能保证读取的高效性

关闭

JayUSA 3楼 2016-12-18 22:08发表

你好，请问，您得到的label是one-hot的形式吗？我输出label之后只是一个整型的数字。

ycszen Re: 2016-12-25 15:49发表

回复JayUSA：这里的label就是一个数字，并不是one-hot格式

希希梦 2楼 2016-10-07 09:54发表

请问为什么不给一下PIL里面的文件呢？

sinat_25236837 1楼 2016-09-03 18:53发表

您好，麻烦您看一下出现这个错误是怎么回事?
tensorflow.python.framework.errors.OutOfRangeError: RandomShuffle Queue '_2_shuffle_batch/random_shuffle_queue' is closed and has insufficient elements (requested 100, current size 44)
[[Node: shuffle_batch = QueueDequeueMany[component_types=[DT_FLOAT, DT_INT32], timeout_ms=-1, _device="/job:localhost/replica:0/task:0/cpu:0"](shuffle_batch/random_shuffle_queue, shuffle_batch/n)]]

ycszen Re: 2016-10-21 19

回复sinat_25236837：我的demo程序中并未使用Coordinator，其实应该使用Coordinator对batch，队列进行管理，让它在读完所有batch的时候停止，这样就不会出现out of range的问题了。

heituzii Re: 2016-10-19 11:2

回复sinat_25236837：你好，请问你问题解决了吗，我也遇到了类似的问题

关闭

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

关闭