炼丹

Tensorflow读取数据2-tfrecord

标签： Tensorflow TFRecords

2017-04-29 22:47 358人阅读 评论

分类： Tensorflow（1）

目录(?) [+]

原文地址： http://blog.csdn.net/u010911921/article/details/70991194

上篇博客谈到了Tensorflow从文件中读取数据，当时采用的是CIFAR-10中的二进制数据，这次记录一下官网推荐的比较通用和高效的数据文件类型的读取——TFRecord文件，这是tensorflow指定的标准格式。

1.TFRecords

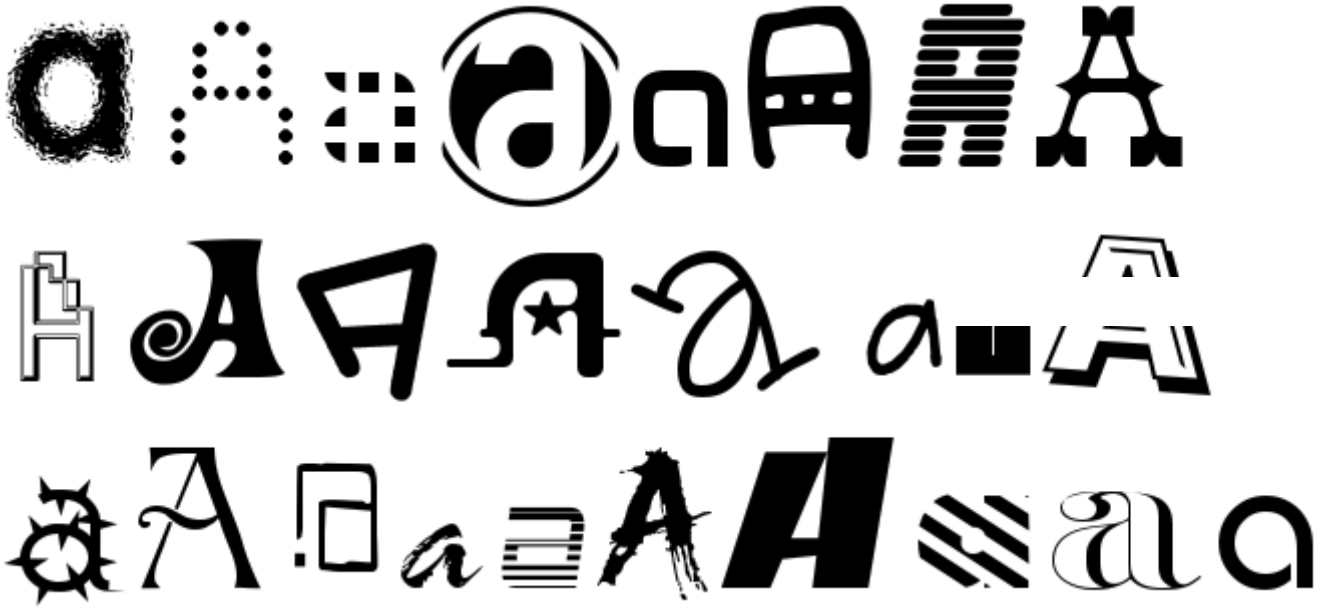
TFRecords本质上是一种二进制文件，他的优点是可以更好的利用内存空间,缺点是生成过程比较耗费时间，特别是数据量比较大的情况下。文件包含了一个 tf.train.Example 的缓冲协议(protocol buffer)其中协议块中包含了字

段 Features .当用程序获得数据以后，就可以将其填充到 Example 的协议缓冲区(protocol buffer)中，然后在将协议缓冲区序列化为字符串，最后通过 tf.python_io.TFRecordWriter 将字符串写入文件。

当从TFRecords文件中读取数据时，可以利用 tf.TFRecordReader 和 tf.parse_single_example 解码器，将 Example 缓冲协议中的内容解析为 Tensor 张量

2.notMNIST 数据集

在实验中采用的数据集合时notMNIST数据集，这个数据集合是由一些各种形态的字母组成的数据集合
由 a~j 10个字母组成，下图是 a 对应的一些图片：



另外需要注意的是，下载的数据集中有几张图片有损坏，所以处理的时候注意跳过。

* 快速集成一个视频直播功能

3.生成TFRecords文件

为了生成TFRecords文件首先是从数据集中，将图片路径放置到一个 image_list ，样本的标签放置到一个 label_list 中。

```
#!/usr/bin/env python3
# --*-- encoding:utf-8 --*--

import tensorflow as tf
import numpy as np
import os
import matplotlib.pyplot as plt
import skimage.io as io

def get_file(file_dir):
    """
    get full image directory and correspond labels
    :param file_dir:
    :return:
    """
    images =[]
    temp =[]
    for root ,sub_folders,files in os.walk(file_dir):
        #image directories
        for name in files:
            images.append(os.path.join(root,name))
        #get 10 sub-folder names
        for name in sub_folders:
            temp.append(os.path.join(root,name))

    labels =[]
    for one_folder in temp:
        n_img = len(os.listdir(one_folder))
```

关闭

```
        letter = one_folder.split('/')[-1]

        if letter =='A':
            labels = np.append(labels,n_img*[1])
        elif letter =="B":
            labels = np.append(labels,n_img*[2])
        elif letter =='C':
            labels = np.append(labels,n_img*[3])
        elif letter =="D":
            labels = np.append(labels,n_img*[4])
        elif letter =="E":
            labels = np.append(labels,n_img*[5])
        elif letter =="F":
            labels = np.append(labels,n_img*[6])
        elif letter =="G":
            labels = np.append(labels,n_img*[7])
        elif letter =="H":
            labels = np.append(labels,n_img*[8])
        elif letter =="I":
            labels =np.append(labels,n_img*[9])
        else:
            labels = np.append(labels,n_img*[10])

    #shuffle
    temp = np.array([images,labels])
    temp = temp.transpose()
    np.random.shuffle(temp)

    image_list = list(temp[:,0])
    label_list = list(temp[:,1])
    label_list = [int(float(i)) for i in label_list]

    return image_list,label_list
```

关闭

当取得 image_list 和 label_list 以后，读取图片数据，然后利用 tf.train.Example 和 tf.train.Features 这两个函数来构建一个 example 然后将其序列化到文件中。基本上就是一个 Example 中包含 Features ， Features 中包含 Feature 字典， Feature 字典是由 float_list 、 bytes_List 或 int64_list 等构成。

```
#将label转换成int64类型，为了构建tf.train.Feature
def int64_feature(value):
    if not isinstance(value,list):
        value = [value]
    return tf.train.Feature(int64_list = tf.train.Int64List(value=value))

#将image转换成bytes类型，同样也是为了构建tf.train.Feature
def bytes_feature(value):
    return tf.train.Feature(bytes_list= tf.train.BytesList(value=[value]))

def convert_to_tfrecord(images,labels,save_dir,name):
    """
    convert all images and labels to one tfrecord file
    :param images:
    :param labels:
    :param save_dir:
    :param name:
    :return:
    """
    filename = os.path.join(save_dir,name+".tfrecords")
    n_samples = len(labels)

    if np.shape(images)[0] != n_samples:
        raise ValueError('Image size %d does not '
                         'match label size %d'%(images.shape[0],n_samples))

    #wait some time
    writer = tf.python_io.TFRecordWriter(filename)
    print("\n Transform start....")
    for i in np.arange(0,n_samples):
```

关闭

```
        try:
            image = io.imread(images[i])
            image_raw = image.tostring()
            label= int(labels[i])
            example = tf.train.Example(features =tf.train.Features(feature={'label':int64_feature(label),
                                                                    "image_raw":bytes_feature(image_raw)}))
            writer.write(example.SerializeToString())
        except IOError as e:
            print("could not read :",images[i])
            print("error:%s"%e)
            print('Skip it')
    writer.close()
    print("Transform done!")
```

这样就完成了TFRecord的生成，但是这个过程会花费较长的时间。

4.TFRecords解码

读取一个文件还是采用上一篇博客中的queue的形式来读取，首先是生成一个文件名层的队列，然后利用 tf.TFRecordReader() 产生的 reader 来读取，然后将其读取到的内容，用 tf.parse_single_example 函数以及 image_raw 读取以及分离出来，为后续操作做准备

```
def read_and_decode(tfrecords_file,batch_size):
    filename_queue = tf.train.string_input_producer([tfrecords_file])

    reader = tf.TFRecordReader()
    _,serialized_example = reader.read(filename_queue)
    img_features = tf.parse_single_example(serialized_example,
                                       features={"label":tf.FixedLenFeature([],tf.int64),
                                             "image_raw":tf.FixedLenFeature([],tf.string),})
    image = tf.decode_raw(img_features['image_raw'],tf.uint8)
    ##########################################################
```

关闭

```
    #
    #put dataaugmentation here
    ########################################################################

    image = tf.reshape(image,[28,28])
    label = tf.cast(img_features['label'],tf.int32)
    image_batch, label_batch = tf.train.batch([image,label],
                                              batch_size = batch_size,
                                              num_threads = 64,
                                              capacity=2000)
    return image_batch,tf.reshape(label_batch,[batch_size])
```

解码以后的后续过程和采用queue处理二进制文件相似。

全部代码下载地

址：https://github.com/ZhichengHuang/LearnTensorflowCode/blob/master/TFRecords/TFRecord_input.py

参考资料

1. https://www.tensorflow.org/programmers_guide/reading_data
2. http://stackoverflow.com/questions/33849617/how-do-i-convert-a-directory tensorflow
3. https://github.com/kevin28520

上一篇 Tensorflow读取数据1

下一篇 解决Caffe训练过程中loss不变问题

相关文章推荐

- Tensorflow之构建自己的图片数据集TFrecords
- 【免费】深入理解Docker内部原理及网络配置--王…
- Tensorflow 训练自己的数据集（二）（TFRecord）
- SDCC 2017之区块链技术实战线上峰会--蔡栋
- tensorflow中关于队列使用的实验
- php零基础到项目实战
- TensorFlow学习笔记（二十四）自制TFRecord数…
- C语言及程序设计入门指导
- TensorFlow 制作自己的TFRecord数据集
- Android入门实战
- 数据读取之TFRecords
- 5天搞定深度学习框架Caffe
- Tensorflow建立与读取TFrecorder文件
- Tensorflow读取数据2-tfrecord
- TensorFlow高效读取数据——TFRecord
- TensorFlow学习记录-- 7 .TensorFlow高效读取数据..

查看评论

关闭

暂无评论

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