# Tfrecords文件的生成与读取

## tfrecords文件的生成

### 1.1基本步骤：

1. 创建tfrecords文件：

writer = tf.python\_io.TFRecordWriter(tfrecord\_file)

1. 使用PIL的Image.open函数打开图像，

img=Image.open(img\_path)

1. 将图像转成二进制格式：

Img\_raw=img.tobytes()

1. 将label和图像封装成example:

example = tf.train.Example(features=tf.train.Features(feature={

"label": tf.train.Feature(int64\_list=tf.train.Int64List(value=[index])),

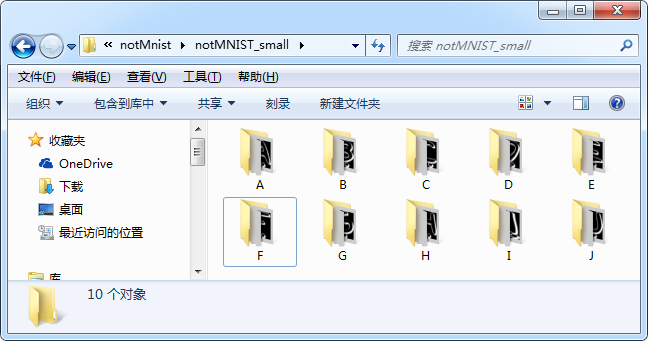
'img\_raw':tf.train.Feature(bytes\_list=tf.train.BytesList(value=[img\_raw]))}))

1. 将example序列化为字符串，写入tfrecords文件：

writer.writer(example.SerializeToString())

### 代码

如在notMNIST\_small文件夹下有以下目录，将其转成tfrecords格式：



|  |
| --- |
| *#coding=utf-8* **import** os **import** tensorflow **as** tf **import** numpy **as** np **from** PIL **import** Image *# 注意Image,后面会用到* **def** image2tfrecords(root\_folder, classes, tfrecord\_file):  writer = tf.python\_io.TFRecordWriter(tfrecord\_file) *# 要生成的文件* images=[]  labels=[]  **for** index **in** range(0, len(classes)):  class\_path = root\_folder + classes[index] + **'\\'  for** img\_name **in** os.listdir(class\_path):  img\_path = class\_path + img\_name *# 每一个图片的地址* images.append(img\_path)  labels.append(index)  *# 使用numpy的乱序功能进行乱序处理，先将其转成numpy的array格式*  temp = np.array([images, labels])  temp = temp.transpose()  np.random.shuffle(temp)  images = list(temp[:,0])  labels = [int(i) **for** i **in** temp[:,1]]  num = len(images)  print(**'convert start...'**)  **for** i **in** range(0, num):  **try**:  image\_path = images[i]  label = labels[i]  img = Image.open(image\_path)  img = img.resize([28, 28])  img\_raw = img.tobytes() *# 将图片转化为二进制格式* example = tf.train.Example(features=tf.train.Features(feature={  **"label"**: tf.train.Feature(int64\_list=tf.train.Int64List(value=[label])),  **'image\_raw'**: tf.train.Feature(bytes\_list=tf.train.BytesList(value=[img\_raw]))  })) *# example对象对label和image数据进行封装* writer.write(example.SerializeToString()) *# 序列化为字符串* **except** IOError **as** e:  print(**'cannot open '**,img\_path)  writer.close()  print(**'convert done!'**) |

测试：

|  |
| --- |
| cwd = **'C:\\Users\\Administrator\\PycharmProjects\\notMnist\\notMNIST\_small\\'** classes = [**'A'**, **'B'**, **'C'**, **'D'**, **'E'**, **'F'**, **'G'**, **'H'**, **'I'**, **'J'**] image2tfrecords(root\_folder=cwd, classes=classes, tfrecord\_file=**'train.tfrecords'**) |

## 读取tfrecords文件

### 2.1 读取tfrecords文件的基本步骤：

1. 使用tf.train.string\_input\_producer读取tfrecords文件，返回队列
2. 使用tf.TFRecordReader读取队列，返回serialized\_example
3. 使用[tf.TFRecordReader](http://wiki.jikexueyuan.com/project/tensorflow-zh/how_tos/tensorflow-zh/SOURCE/api_docs/python/io_ops.html)的[tf.parse\_single\_example](http://wiki.jikexueyuan.com/project/tensorflow-zh/how_tos/tensorflow-zh/SOURCE/api_docs/python/io_ops.html)解析器将Example协议内存块(protocol buffer)解析为张量img\_features，其中img\_features['image\_raw']为图像数据，img\_features['label']为标签数据
4. 使用tf.decode\_raw进行解码
5. 使用tf.train.batch生成batch

### 2.2 代码

|  |
| --- |
| *#coding=utf-8* **import** tensorflow **as** tf **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)  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,  capacity=1000,  num\_threads=64)  **return** image\_batch, tf.reshape(label\_batch, [batch\_size]) |

测试：

|  |
| --- |
| *#coding=utf-8* **import** tensorflow **as** tf **import** numpy **as** np **import** matplotlib.pyplot **as** plt **import** readTfrecord  BATCH\_SIZE = 10 tfrecord\_file = **'notMNIST\_train.tfrecords'  def** plot\_images(images, labels):  **for** i **in** np.arange(0, BATCH\_SIZE):  print(**'label:'**,labels[i])  plt.imshow(images[i])  plt.show()  image\_batch, label\_batch = readTfrecord.read\_and\_decode(tfrecords\_file=tfrecord\_file, batch\_size=BATCH\_SIZE) **with** tf.Session() **as** sess:  i = 0  coord = tf.train.Coordinator()  threads = tf.train.start\_queue\_runners(coord=coord)  **try**:  **while not** coord.should\_stop() **and** i < 2:#只查看2个batch  *# just plot one batch size* image, label = sess.run([image\_batch, label\_batch])  plot\_images(image, label)  i += 1  **except** tf.errors.OutOfRangeError:  print(**'done!'**)  **finally**:  coord.request\_stop()  coord.join(threads) |