TFRecord详解

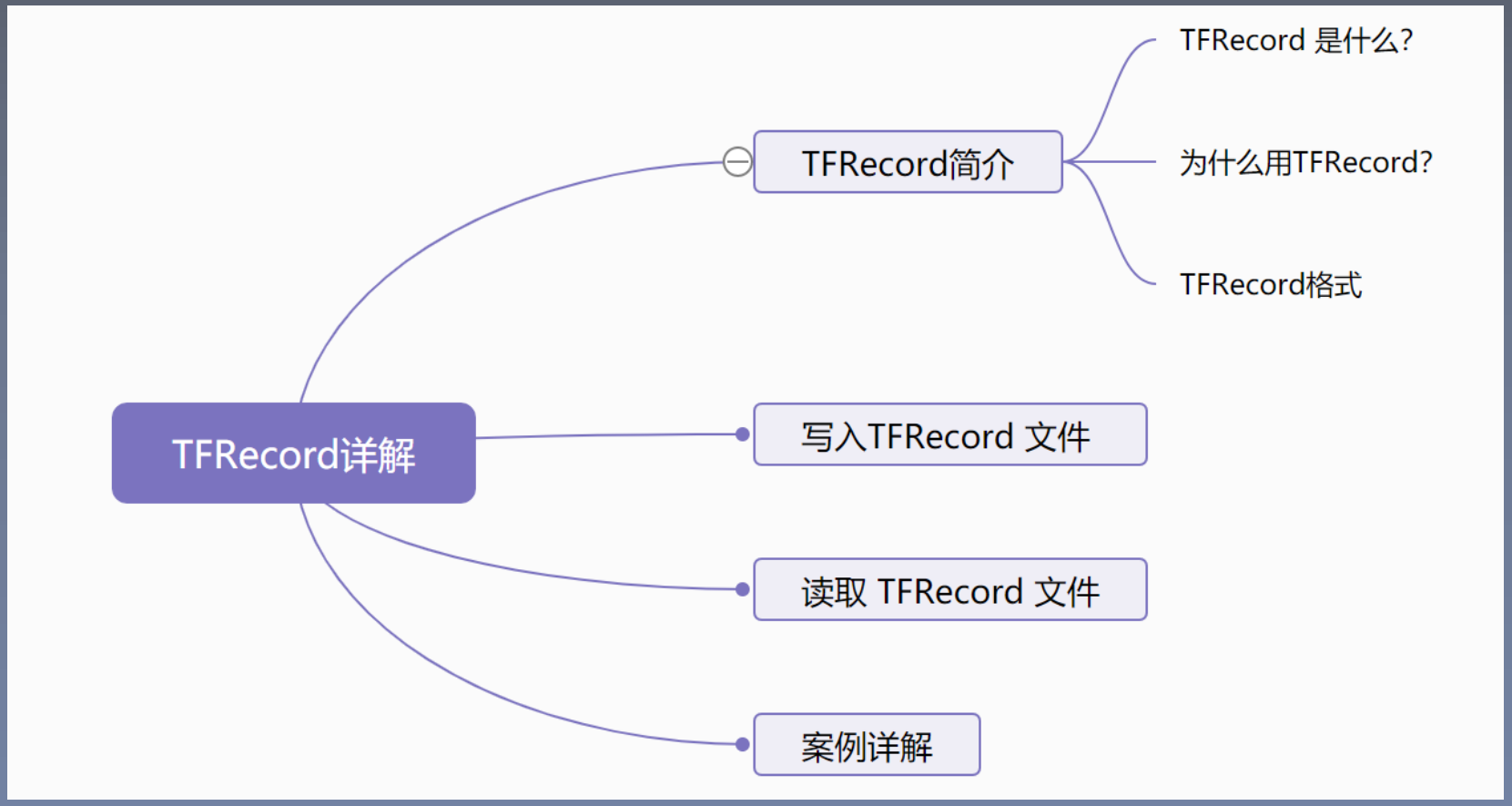
导师: GAUSS

深度之眼
deepshare.net

目录

知识树

Knowledge tree



深度之眼
deepshare.net

TFRecord简介

TFRecord 是什么?

TFRecord 是Google官方推荐的一种数据格式，是Google专门为TensorFlow设计的一种数据格式。

实际上，TFRecord是一种二进制文件，其能更好的利用内存，其内部包含了多个**tf.train.Example**， 而Example是protocol buffer数据标准的实现，在一个Example消息体中包含了一系列的**tf.train.feature**属性，而 每一个feature 是一个key-value的键值对，其中，key 是string类型，而value 的取值有三种：

TFRecord 是什么?

- bytes_list: 可以存储string 和byte两种数据类型。
- float_list: 可以存储float(float32)与double(float64) 两种数据类型 。
- int64_list: 可以存储：bool, enum, int32, uint32, int64, uint64 。

值的一提的是，TensorFlow 源码中到处可见 .proto 的文件，且这些文件定义了TensorFlow重要的数据结构部分，且多种语言可直接使用这类数据，很强大。

为什么用TFRecord?

TFRecord 并非是TensorFlow唯一支持的数据格式，你也可以使用CSV或文本等格式，但是对于TensorFlow来说，TFRecord 是最友好的，也是最方便的。前面提到，TFRecord内部是一系列实现了protocol buffer数据标准的Example。对于大型数据，相比其余数据格式，protocol buffer类型的数据优势很明显。

在数据集较小时，我们会把数据全部加载到内存里方便快速导入，但当数据量超过内存大小时，就只能放在硬盘上来一点点读取，这时就不得不考虑数据的移动、读取、处理等速度。使用TFRecord就是为了**提速和节约空间**的。

参考：https://halfrost.com/protobuf_encode/

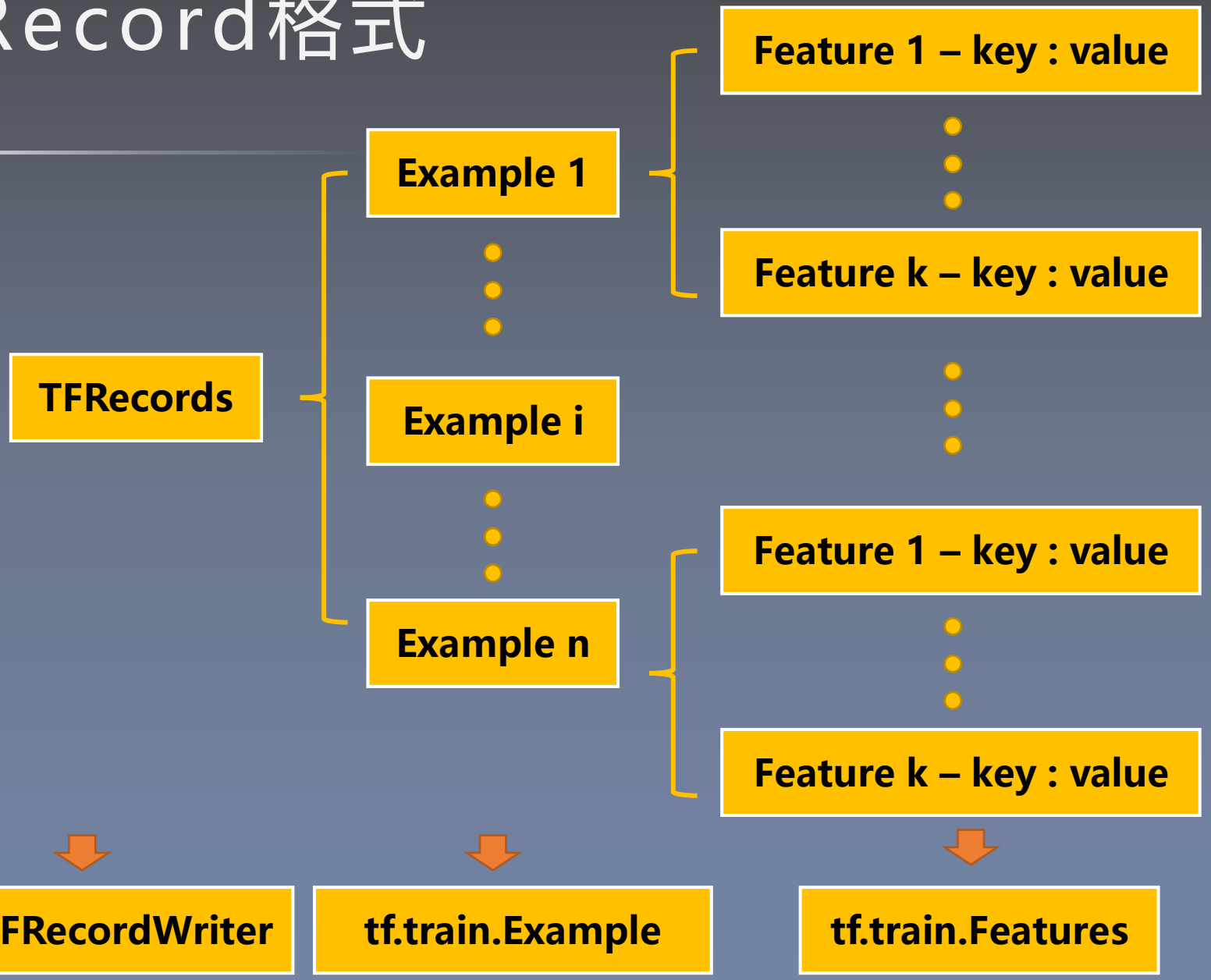
https://zhuanlan.zhihu.com/p/50808597

TFRecord格式

TFRecord 可以理解为一系列序列化的 tf.train.Example 元素所组成的列表文件，而每一个 tf.train.Example 又由若干个 tf.train.Feature 的字典组成。

```
[
    {   # example 1 (tf.train.Example)
        'feature_1': tf.train.Feature,
        ...
        'feature_k': tf.train.Feature
    },
    ...
    {   # example N (tf.train.Example)
        'feature_1': tf.train.Feature,
        ...
        'feature_k': tf.train.Feature
    }
]
```

TFRecord格式



TFRecord格式

- bytes_list: 可以存储string 和byte两种数据类型。
- float_list: 可以存储float(float32)与double(float64) 两种数据类型 。
- int64_list: 可以存储：bool, enum, int32, uint32, int64, uint64 。

- int64_list: tf.train.Feature(int64_list = tf.train.Int64List(value=输入))
- float_list: tf.train.Feature(float_list = tf.train.FloatList(value=输入))
- bytes_list ： tf.train.Feature(bytes_list=tf.train.BytesList(value=输入))

注：输入必须是list(向量)

深度之眼
deepshare.net

写入TFRecord 文件

生成TFRecord格式数据

为了将形式各样的数据集整理为 TFRecord 格式，我们可以对数据集中的每个元素进行以下步骤：

- 读取该数据元素到内存；
- 建立 Feature 的字典；
- 将该元素转换为 tf.train.Example 对象（每一个 tf.train.Example 由若干个 tf.train.Feature 的字典组成）；
- 将该 tf.train.Example 对象序列化为字符串，并通过一个预先定义的 tf.io.TFRecordWriter 写入 TFRecord 文件。

生成TFRecord格式数据示例

深度之眼
deepshare.net

Cats vs dogs数据集示例

```
test_cat_filenames = [test_cats_dir + filename for filename in os.listdir(test_cats_dir)]
test_dog_filenames = [test_dogs_dir + filename for filename in os.listdir(test_dogs_dir)]
test_filenames = test_cat_filenames + test_dog_filenames
test_labels = [0] * len(test_cat_filenames) + [1] * len(test_dog_filenames)  # 将 cat 类的标签设为0，dog 类的标签设为1
```

1

```
with tf.io.TFRecordWriter(test_tfrecord_file) as writer:
    for filename, label in zip(test_filenames, test_labels):
        image = open(filename, 'rb').read()      # 读取数据集图片到内存，image 为一个 Byte 类型的字符串
        feature = {                              # 建立 tf.train.Feature 字典
            'image': tf.train.Feature(bytes_list=tf.train.BytesList(value=[image])),  # 图片是一个 Bytes 对象
            'label': tf.train.Feature(int64_list=tf.train.Int64List(value=[label]))   # 标签是一个 Int 对象
        }
        example = tf.train.Example(features=tf.train.Features(feature=feature)) # 通过字典建立 Example
        serialized = example.SerializeToString() #将Example序列化
        writer.write(serialized)   # 写入 TFRecord 文件
```

2

3

4

注意

tensorflow feature类型只接受list数据，但如果数据类型是矩阵或者张量该如何处理？

- 转成list类型：将张量flatten成list(也就是向量)，再用写入list的方式写入。
- 转成string类型：将张量用.tostring()转换成string类型，再用tf.train.Feature(bytes_list=tf.train.BytesList(value=[input.tostring()]))来存储。

深度之眼
deepshare.net

读取TFRecord文件

读取TFRecord文件步骤

而读取 TFRecord 数据则可按照以下步骤：

- 通过 tf.data.TFRecordDataset 读入原始的 TFRecord 文件（此时文件中的 tf.train.Example 对象尚未被反序列化），获得一个 tf.data.Dataset 数据集对象；
- 定义Feature结构，告诉解码器每个Feature的类型是什么；
- 通过 Dataset.map 方法，对该数据集对象中的每一个序列化的 tf.train.Example 字符串执行 **tf.io.parse_single_example** 函数，从而实现反序列化。

读取TFRecord文件示例


```
feature_description = { # 定义Feature结构，告诉解码器每个Feature的类型是什么
    'image': tf.io.FixedLenFeature([], tf.string),
    'label': tf.io.FixedLenFeature([], tf.int64),
}

def _parse_example(example_string): # 将 TFRecord 文件中的每一个序列化的 tf.train.Example 解码
    feature_dict = tf.io.parse_single_example(example_string, feature_description)
    feature_dict['image'] = tf.io.decode_jpeg(feature_dict['image'])    # 解码JPEG图片
    feature_dict['image'] = tf.image.resize(feature_dict['image'], [256, 256]) / 255.0
    return feature_dict['image'], feature_dict['label']

batch_size = 32

train_dataset = tf.data.TFRecordDataset("train.tfrecords")    # 读取 TFRecord 文件
train_dataset = train_dataset.map(_parse_example)
train_dataset = train_dataset.shuffle(buffer_size=23000)
train_dataset = train_dataset.batch(batch_size)
train_dataset = train_dataset.prefetch(tf.data.experimental.AUTOTUNE)
```

2

3

1

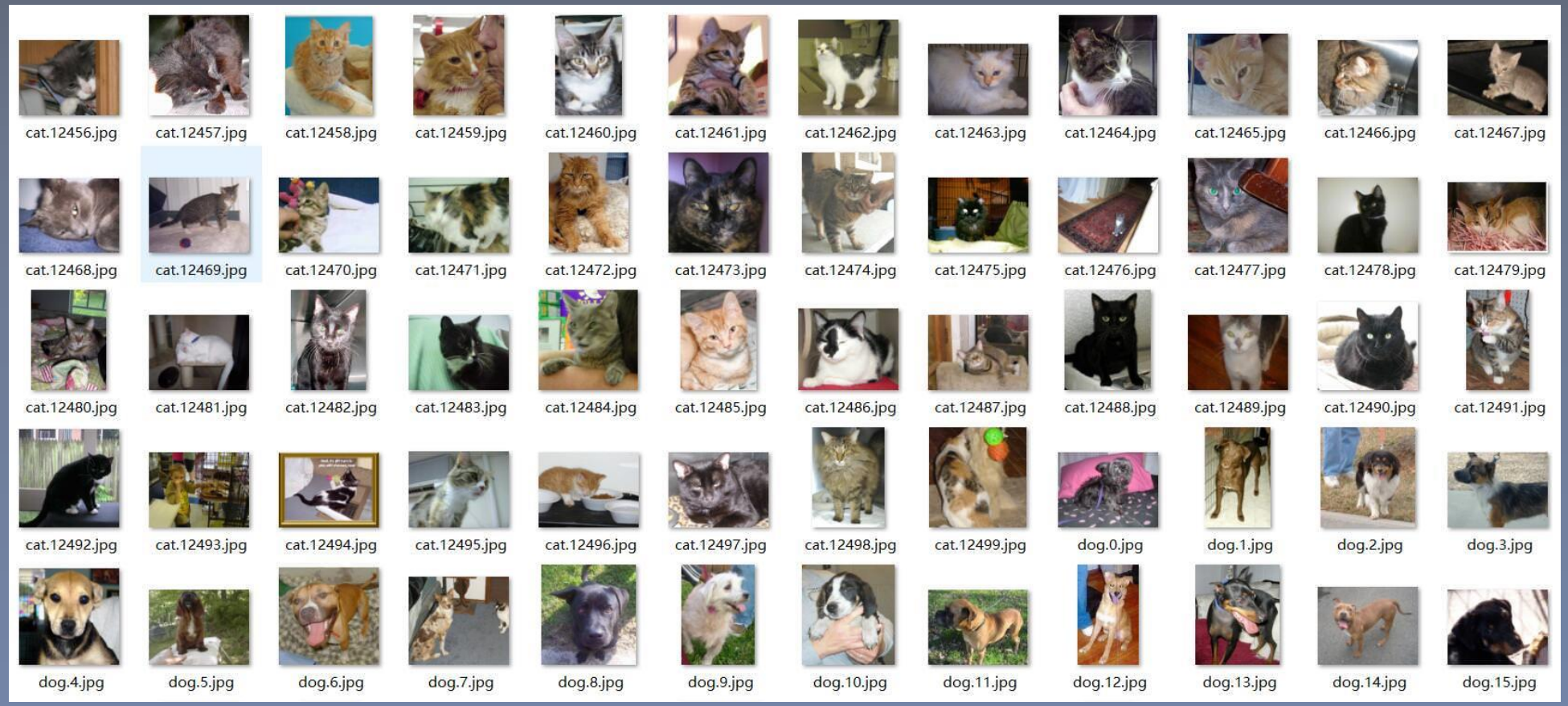
深度之眼
deepshare.net

案例讲解

实战二：Cats vs. Dogs比赛项目

深度之眼
deepshare.net

项目网址：https://www.kaggle.com/c/dogs-vs-cats

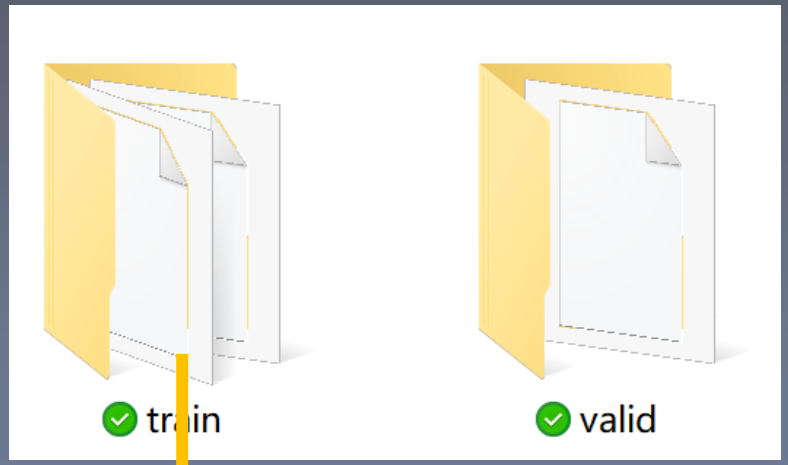


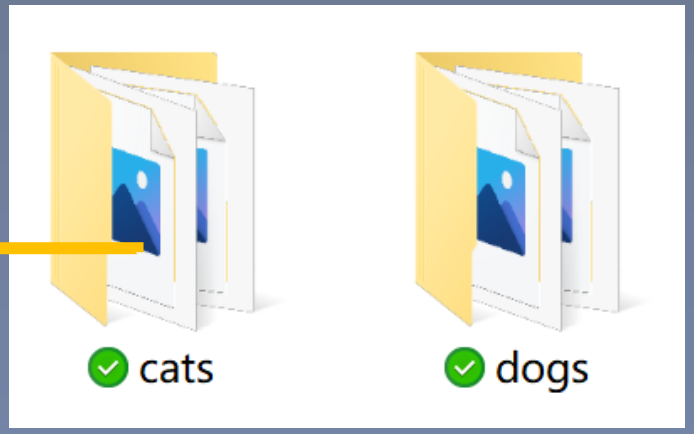
背景介绍

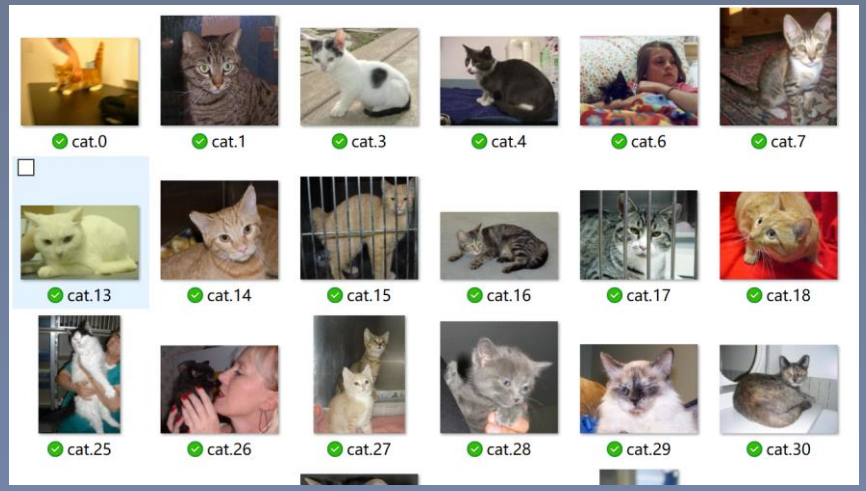
任务目标：Cats vs. Dogs（猫狗大战）是Kaggle大数据竞赛某一年的一道赛题，利用给定的数据集，用算法实现猫和狗的识别。

图像分类问题

数据集介绍







本节小结

Summary

TFRecord详解	TFRecord简介	什么是TFRecord？TFRecord作用，结构
	写入TFRecord 文件	写入步骤
	读取TFRecord文件	读取步骤
	案例讲解	实战二：Cats vs dogs比赛项目

结语

我 说

GAUSS老师个人公众号，主要分享NLP、推荐、比赛实战相关知识！

深度之眼
deepshare.net

联系我们：

电话：18001992849

邮箱：service@deepshare.net

Q Q： 2677693114

公众号

客服微信