设计的一种数据格式 1.Google官方推荐使用的数据格 式化存储工具,为TensorFlow量 身打造的。 原因 2.规范了数据的读写方式,数据 读取和处理的效率都会得到显著 **TFrecord** 的提高 对应于图像及相应的标注信息 key string 和byte tf.train.bytes_list 架构 example feature tf.train.float_list float(float32)与double(float64) value bool, enum, int32, uint32, tf.train.int64_list int64, uint64 1.使用<u>tf.io</u>.TFRecordWriter打开 TFRecords文件(指明位置) 2.获取要写入的数据,进行类型 转换 3.将类型转换后的数据传入 tf.train.Feature创建的特征中 tf.train.Feature 1.写入到TFrecord 4.将特征传入tf.train.Example创 建的example中 5.使用 example.SerializeToString()将 example序列化为字符串 6.使用writer.write将序列化后的 example写入TFRecords文件 7.使用writer.close()关闭文件 1.使用tf.data.TFRecordDataset 数据处理 来获取TFRecord文件中的数据 2.定义特征的描述方法,与写入 时是对应的 2.读取Tfrecord文件 3.使用 <u>tf.io</u>.parse_single_example将一 个example转换为原始数据 4.使用功能map方法对所有数据 进行处理获取最终的结果 YOLOV3案例 对数据集中的各个元素进行预处 重点: map Repeat, batch, shuffle 3.数据处理的pipeline 解耦了cpu处理数据的时间 和 GPU进行数据运算时间,在最后 prefetch 面使用prefetch 4.数据处理 将图像和box作调整(416*416) 1.CBL Conv+Bn+Leaky_relu激活函数 ResX由一个CBL和X个残差组件 2.ResX 构成,是Yolov3中的大组件 CBL和ResX来构成 3.Backbone CBL*5结果进行特征融合,经过 4.输出部分 CBL后获取网路输出结果 模型构建 将Backbone和输出部分进行串 5.yoloV3 接,并特征融合获取最终的检测 结果 1.利用网络输出结果进行坐标修 正, 获取检测框结果 6.输出结果转换 2.对置信度和分类结果进行 sigmoid获取最终结果 只有负责检测anchor才会计入损 1.回归损失 失,对x,y,w,h分别求均方误差 置信度的损失是二分类的交叉熵 2.置信度损失 损失函数,所有的box都计入损 1.损失函数 失计算 分类的损失是二分类的交叉熵损 3.分类损失 失, 只有负责检测目标的才计算 损失 负责检测目标的anchor 目标gt(ground Truth)中心点落 模型训练 入到的gridcell,对应与GT交并比 1.正样本 最大的anchor设置为正样本 confidence=1,框的目标值: gt, 类别: 真实的类别 2.正负样本的设置 gt中心点未落入的gridcell对应的 anchor 2.负样本 gt中心点落入的gridcell对应的与 GT交并比较小的anchor

所有的目标值设置为0

Google官方推荐的一种数据格

式,是Google专门为TensorFlow