Faster-RCNN 代码实现

一 rpn实现

二 second stage实现

2.1 整体结构

输入的是输入是2k个分类+4k个回归.

输出是”最终”(final)预测的box的回归参数.

以及包含分类预测.

1. 对rpn框的后处理

输入的是rpn的输出结果.这里只是对回归框做后处理.

输出的是做大为64个预测物体的尺寸信息(centerx,centery,w,h).以及实际的预测物体的个数(num\_proposals)

这个函数可以认为起到一个**筛子的作用,**把一些不太像(需要展开)预测框的筛出去.

1. 结合筛选候选框和featuresmap,把对应位置的featuremap展开(flattened).
2. 提取分类框需要的featuresmap.(box\_classifier\_features).
3. 依据cls的feature**对物体做分类预测**.输出两个结果.

其一, 更加准确的回归框.(这个是否是前面的”筛选候选框”的子集?,应该是).

其二, 这些回归框对应物体的分类结果.

1. 随后在对rpn的预测框的回归结果做一个nms.(不知此处含义,这里并不是second 预测的结果).



2.2 self.\_postprocess\_rpn

输入的尺寸是很大的,符合rpn的输出特征.



1. Shape分为tf.shape的动态shape(根据每个样本变化),和getshape的静态shape.
2. 对rpn的预测框做一个”矫正”操作.

输入的是,一组rpn的anchors.还是比较大的尺寸.



输出是预测框.



三 loss function