问题

这个问题拖到现在才开始整理属实不应该，实习面华为的时候被问过，我对 Cascade RCNN 的理解貌似面试官不太认同，于是便把这个问题放入了列表中，然后前天面科大讯飞的时候又被问到 Cascade RCNN 的 motivation 是什么，我这时候的回答不知道面试官满不满意，但是更加强烈地觉得这个网络可能还有更多的精髓，因此这篇详细来整理一下。

先介绍一下 Faster RCNN中相关的一些点：

在 Faster RCNN 中，训练阶段，经过 RPN 之后，会提出 2000 左右个 proposals ，这些 proposals 会被送入到 Fast R-CNN 结构中，在 Fast R-CNN 的结构中，首先会计算每个 proposals 和 gt 之间的 IOU ，然后通过人为设定的 IOU 阈值（通常为 0.5），把这些 proposals 分成正样本和负样本（之后正样本才会参与到后面的 bbox 回归学习，从损失函数的表达中也可以看出来，只有正样本才被算入损失函数中），并对这些样本进行采样，使得他们之间的比例尽量满足（1:3，二者总数为 128 ），之后这 128 个proposals 被送入到 Roi Pooling 层，最后进行类别分类和 box 回归。

在 inference 阶段，RPN 网络提出了 300 左右的 proposals，这些 proposals 被送入到 Fast RCNN 结构中，与 training 阶段不同的是，inference 阶段没有办法对这些 proposals 重采样，因为 inference 阶段不知道 gt，所以只能把他们都作为正样本，直接进入 Roi Pooling ，之后进行分类和box 回归。

Cascade 中所做的实验以及所揭示的问题

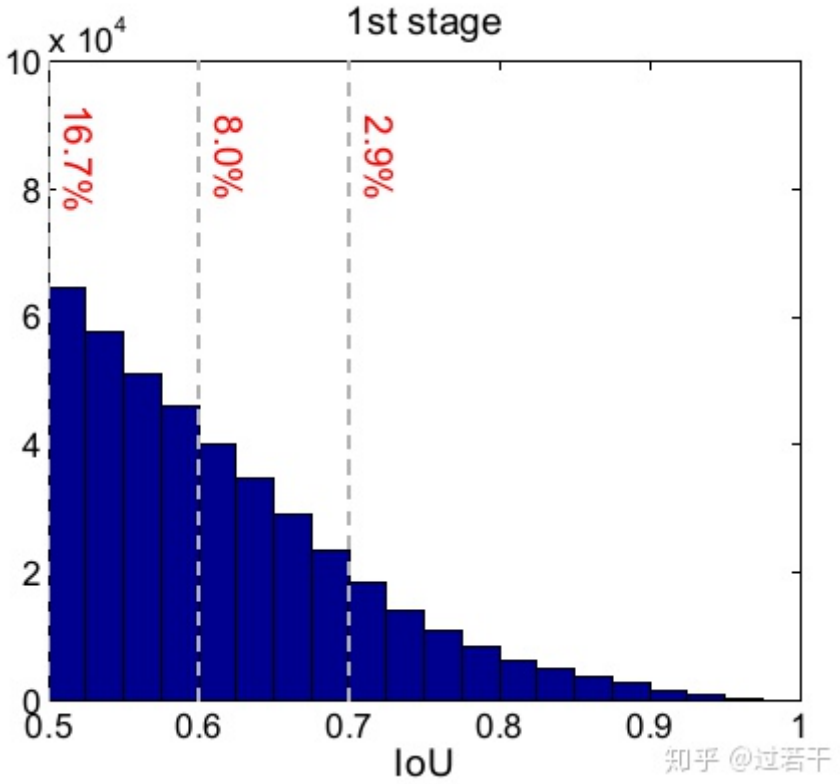
要提高Fast R-CNN目标检测的精度，一个非常直觉的做法便是提高判定是正负样本的 IOU 阈值，这样后面的 detector 便接收的是更高质量的 proposals ，自然能产生高精度的 box 。但是这样便会产生两个问题：

1. **过拟合问题**。提高了 IOU 阈值，满足这个阈值条件的 proposals 必然比之前少了，容易导致过拟合。
2. **更加严重的 mismatch 问题**。（这样的话训练阶段模型看到的都是质量比较高的的 proposals，会导致过拟合，而 inference 阶段由于没有经过阈值的采样，因此会存在很多的质量较差的 proposals，而detector 之前都没见过这么差的 proposal ，精度便下降了。）

> ### mismatch
>
> 从上面我们明显可以看出，training 阶段和 inference 阶段，bbox回归器的输入分布是不一样的，training阶段的输入 proposals 质量更高（因为被采样过，IOU > threshold ），而 inference 阶段的则相对差点，这就是论文中提到的 **mismatch** 问题，这个问题是固有存在的，通常 threshold 取 0.5 时，mismatch 问题还不会很严重。

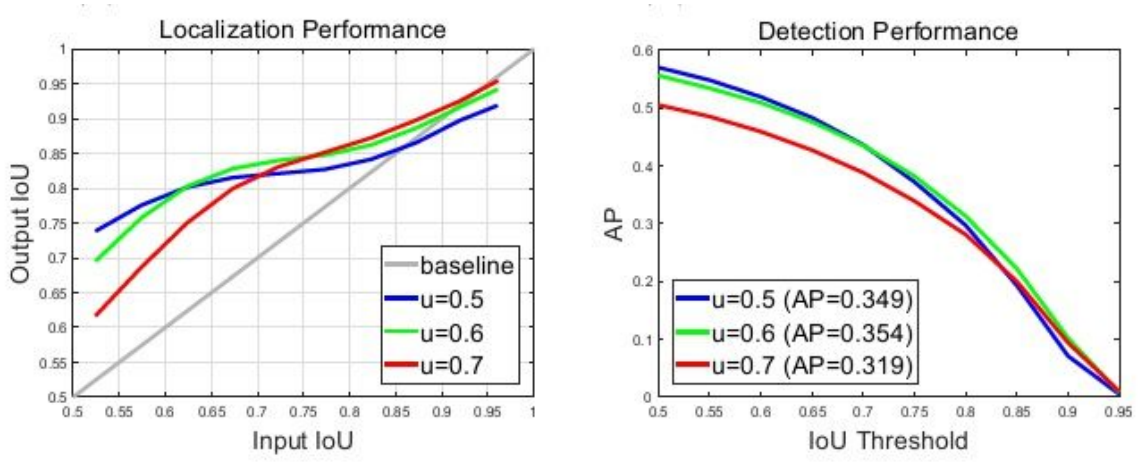
因此作者做了实验来验证这样的说法：

实验一：统计 RPN 输出的 proposals 在各个 IOU 范围内的数量。



可以看出，IOU 在 0.6，0.7 以上的proposals 数量很少，直接提高阈值的话，确实有可能出现上述两个问题。

实验二：将IOU 阈值设为 0.5，0.6，0.7时，proposals的分布以及检测精度



(c) Regressor (d) Detector

Figure 1. The detection outputs, localization and detection performance of object detectors of increasing IoU threshold u.

(c) 图中横轴表示RPN的输出 proposal 的IoU，纵轴表示proposal经过 box reg 的新的IoU。这个图可以这么理解：当横坐标为 0.55 也就是RPN的输出 proposal 的IoU在 0.5 左右时，将 IOU 的阈值设为 0.5 的表现比其他两个更高阈值的更好。横坐标为 0.65或者 0.75时的情况同样。

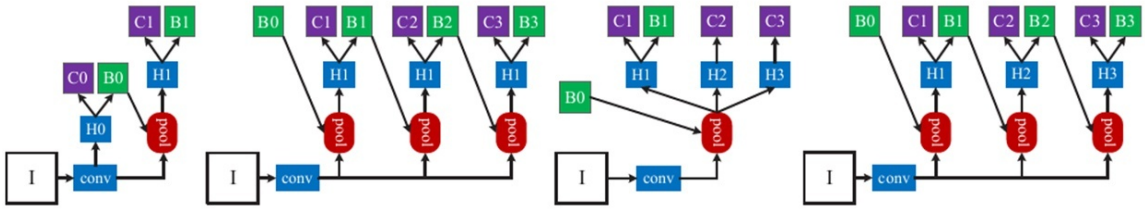
因此可以得出以下结论：

1. 只有proposal自身的阈值和训练器训练用的阈值较为接近的时候，训练器的性能才最好。
2. 单一阈值训练出的检测器效果非常有限，单一阈值不能对所有的Proposals都有很好的优化作用。

(d) 图中横轴表示inference阶段，判定box为tp的IoU阈值，纵轴为mAP。可以看到IoU阈值从0.5提到0.7时，AP下降很多。进一步说明了 mismatch 的问题。

Cascade 的结构设计

做了上面两个实验之后，那么改进的思路便很明显了。既然单一一个阈值训练出的检测器效果有限，作者就提出了muti-stage的结构，每个stage都有一个不同的IoU阈值，而且阈值是逐步提高。



(a) Faster R-CNN (b) Iterative BBox at inference (c) Integral Loss (d) Cascade R-CNN

Figure 3. The architectures of different frameworks. "I" is input image, "conv" backbone convolutions, "pool" region-wise feature extraction, "H" network head, "B" bounding box, and "C" classification. "B0" is proposals in all architectures.

(b) 中的 detector 的是共享的，而且三个分支的IOU阈值都取 0.5 。而经过上面的分析，我们已经知道单一阈值0.5，是无法对所有proposal取得良好效果的。此外，detector会改变样本的分布，这时候再使用同一个共享的H对检测肯定是有影响的。

(c) 第一个stage的输入IoU的分布很不均匀，高阈值proposals数量很少，导致负责高阈值的detector很容易过拟合。此外在inference时，3个detector的结果要进行ensemble，但是它们的输入的IoU大部分都比较低，这时高阈值的detector也需要处理低IoU的proposals，它就存在较严重的mismatch问题，它的detector效果就很差了。

总结

RPN提出的proposals大部分质量不高，导致没办法直接使用高阈值的detector，Cascade R-CNN使用cascade回归作为一种重采样的机制，逐stage提高proposal的IoU值，从而使得前一个stage重新采样过的proposals能够适应下一个有更高阈值的stage。

- 每一个stage的detector都不会过拟合，都有足够满足阈值条件的样本。
- 更深层的detector也就可以优化更大阈值的proposals。
- 每个stage的H不相同，意味着可以适应多级的分布。
- 在inference时，虽然最开始RPN提出的proposals质量依然不高，但在每经过一个stage后质量都会提高，从而和有更高IoU阈值的detector之间不会有很严重的mismatch。

补充

第一个 stage 的输入是 RPN 提出的 2000个 rois，筛选了128个，然后后面 2 个 stage 都是继续沿用这 128 个。其实从图中也可以看出，但是不知道为啥很多人在这里有疑问。

参考资料

[Cascade R-CNN 详细解读]