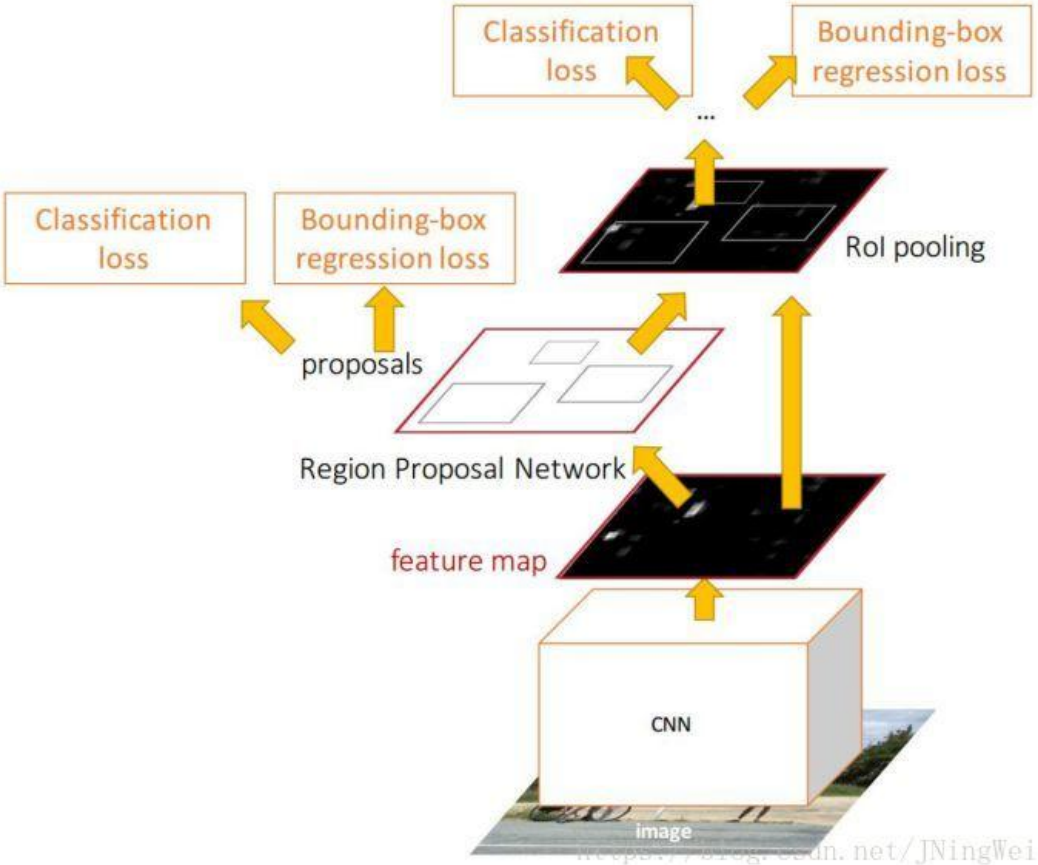
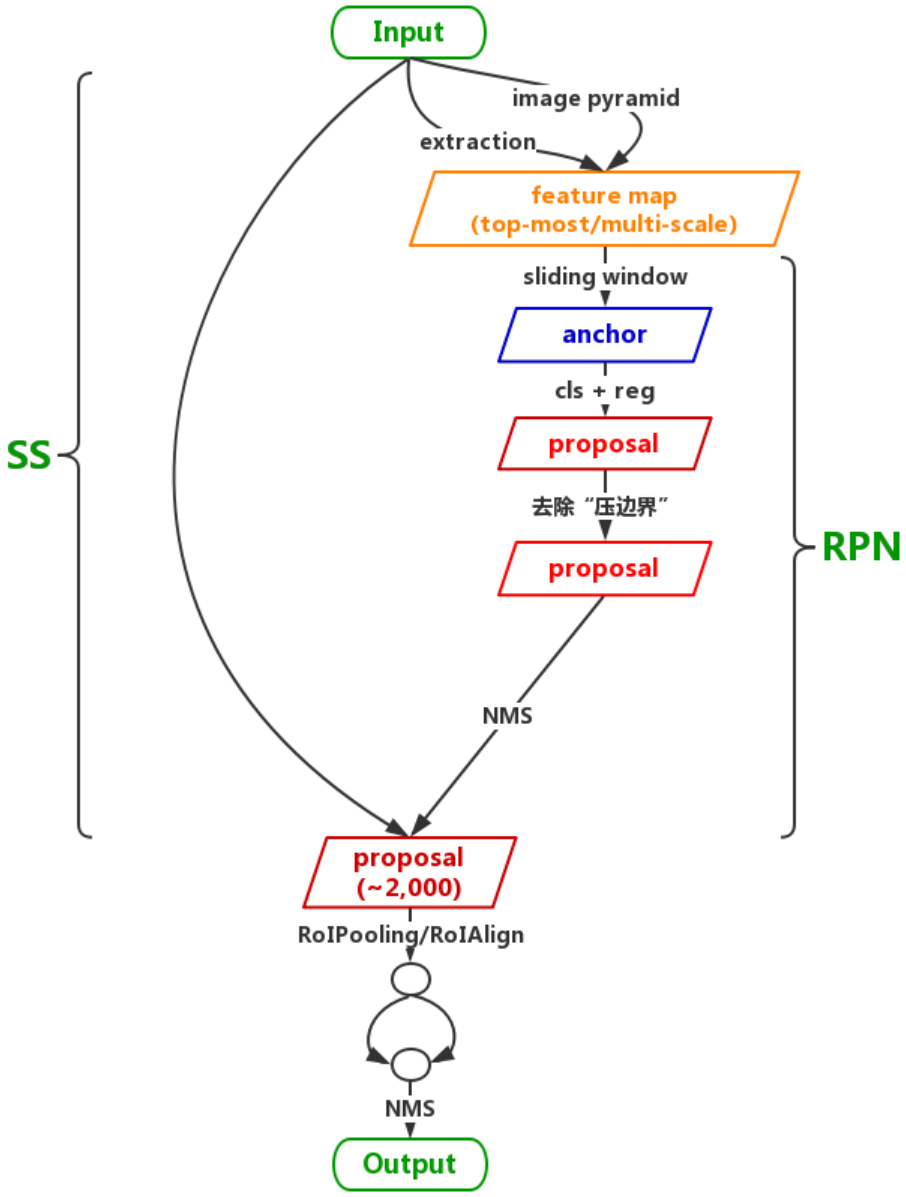
RPN (区域候选网络)

Overview

RPN 的本质是 “ 基于滑窗的无类别 obejct 检测器 ” ：



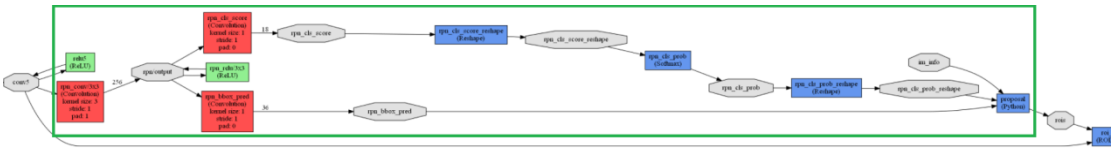
RPN 所在的位置：



Note：

1. 只有在 train 时，cls+reg 才能得到强监督信息(来源于 ground truth)。即 ground truth 会告诉 cls+reg 结构，哪些才是真的前景，从而引导 cls+reg 结构学得正确区分前后景的能力；在 reference 阶段，就要靠 cls+reg 自力更生了。
2. 在 train 阶段，会输出约 2000 个 proposal，但只会抽取其中 256 个 proposal 来训练 RPN 的 cls+reg 结构；到了 reference 阶段，则直接输出最高 score 的 300 个 proposal。此时由于没有了监督信息，所有 RPN**并不知道这些 proposal 是否为前景**，整个过程只是惯性地推送一波无 tag 的 proposal 给后面的 Fast R-CNN。
3. RPN 的运用使得 region proposal 的额外开销就只有一个两层网络。

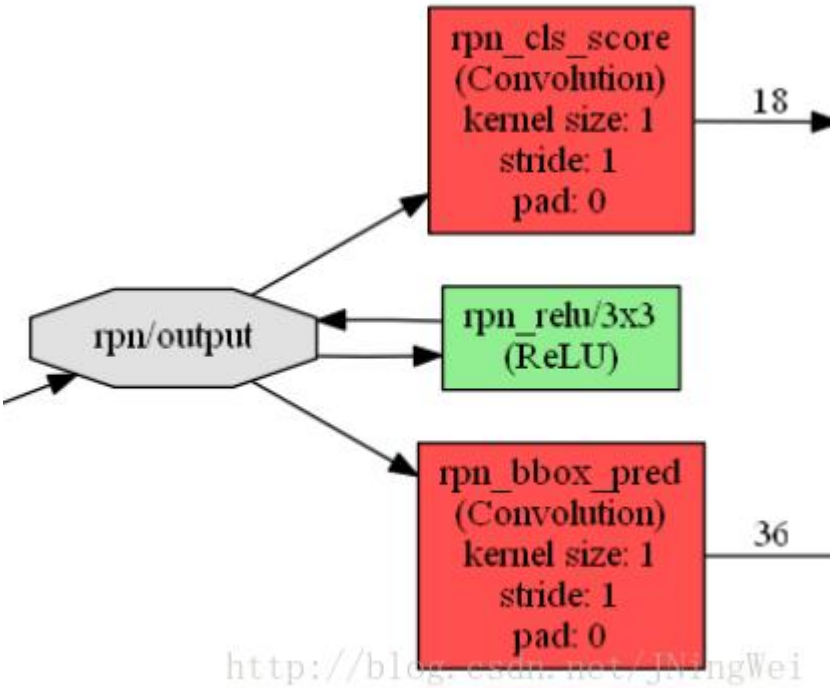
放大之后是这样：



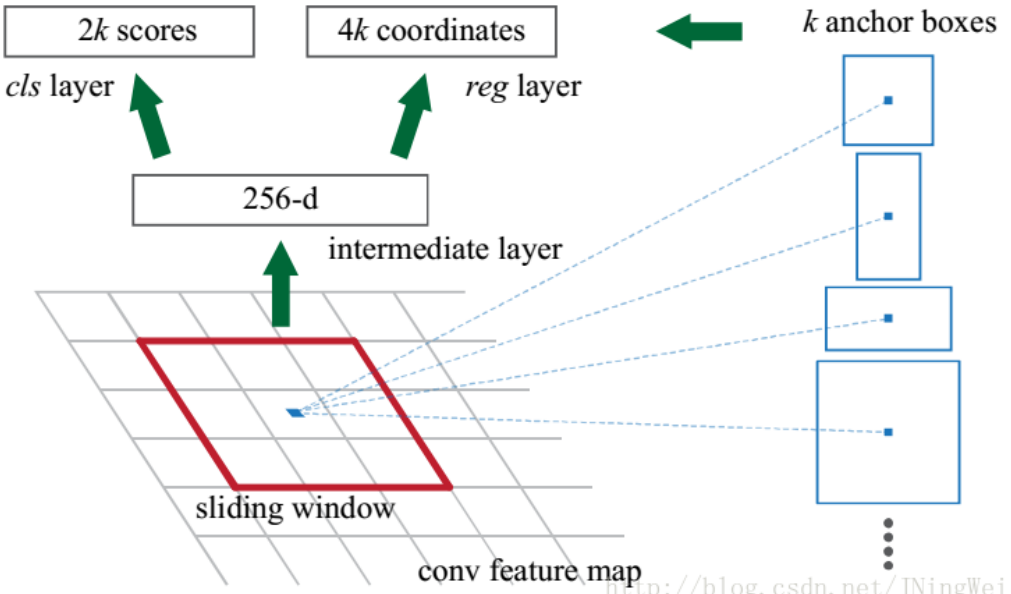
庖丁解牛

RPN 由以下三部分构成：

1.在 RPN 头部 ，通过以下结构生成 anchor（其实就是一堆有编号有坐标的 bbox）：

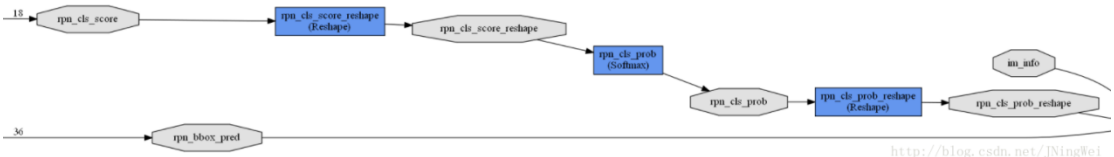


论文中的这幅插图对应的就是 RPN 头部：



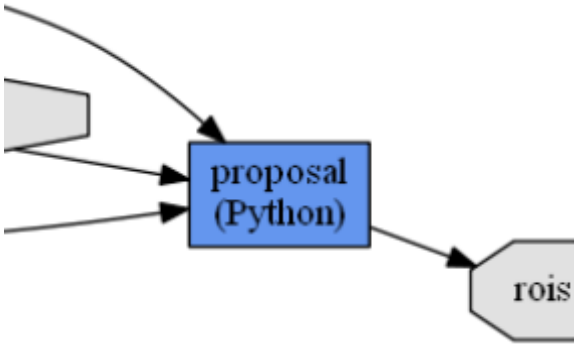
（曾经以为这张图就是整个 RPN，于是百思不得其解，走了不少弯路。。。）

2. 在 RPN 中部， 分类分支（cls） 和 边框回归分支（bbox reg） 分别对这堆 anchor 进行各种计算：



Note： two stage 型的检测算法在 RPN 之后 还会进行 再一次 的 分类任务 和 边框回归任务，以进一步提升检测精度。

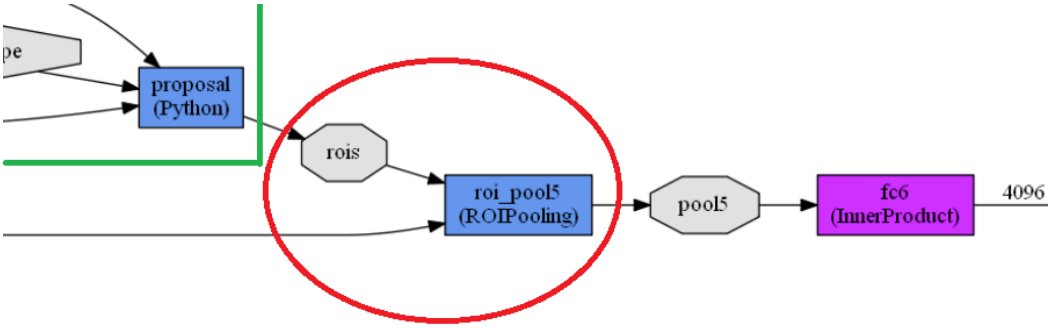
3. 在 RPN 末端，通过对 两个分支的结果进行汇总，来实现对 anchor 的 初步筛除（先剔除越界的 anchor，再根据 cls 结果通过 NMS 算法去重）和 初步偏移（根据 bbox reg 结果），此时输出的都改头换面叫 Proposal 了：



后话

RPN 之后，proposal 成为 RoI (感兴趣区域) ，被输入 RoIPooling 或 RoIAlign 中进行 size 上的归一化。当然，这些都是 RPN 网络 之后 的操作了，严格来说并 不属于 RPN 的范围了。

图中 绿框内 为 RPN ，红圈内 为 RoI 以及其对应的 Pooling 操作：



note

但是如果只在最后一层 feature map 上映射回原图像，且初始产生的 anchor 被限定了尺寸下限，那么低于最小 anchor 尺寸的小目标虽然被 anchor 圈入，在后面的过程中依然容易被漏检。

但是 FPN 的出现，大大降低了小目标的漏检率，使得 RPN 如虎添翼。