**解答关于R-FCN的所有疑惑（原创）**

[张磊\_0503](https://www.jianshu.com/u/67426bc5bd59) 关注

2018.01.13 13:21\* 字数 1758 阅读 5733评论 4喜欢 10

论文链接：[https://arxiv.org/pdf/1605.06409.pdf](https://link.jianshu.com/?t=https%3A%2F%2Farxiv.org%2Fpdf%2F1605.06409.pdf)  
Matlab源码：[https://github.com/daijifeng001/r-fcn](https://link.jianshu.com/?t=https%3A%2F%2Fgithub.com%2Fdaijifeng001%2Fr-fcn)

R-FCN是微软亚洲研究院的代季峰在2016年提出的一种全新的目标检测结构。它对传统的Faster R-CNN结构进行了改造，将ROI层后的卷积都移到了ROI层前，并利用一种位置敏感的特征图来评估各个类别的概率，使其在保持较高定位准确度的同时，大幅提高检测速率。  
网上很容易找到关于R-FCN的各种文章，所以不再重复介绍它的结构，只是选几个容易引起误解的点做深入解读。

**理解难点1：平移不变性和平移可变性**

作者在论文中提到了两个概念，平移不变性（translation invariance）和平移可变性（translation variance）。平移不变性比较好理解，在用基础的分类结构比如ResNet、Inception给一只猫分类时，无论猫怎么扭曲、平移，最终识别出来的都是猫，输入怎么变形输出都不变这就是平移不变性，网络的层次越深这个特性会越明显。平移可变性则是针对目标检测的，一只猫从图片左侧移到了右侧，检测出的猫的坐标会发生变化就称为平移可变性。当卷积网络变深后最后一层卷积输出的feature map变小，物体在输入上的小偏移，经过N多层pooling后在最后的小feature map上会感知不到，这就是为什么原文会说网络变深平移可变性变差。

再来看个Faster R-CNN + ResNet-101结构的例子。如果在Faster R-CNN中没有ROI层，直接对整个feature map进行分类和位置的回归，由于ResNet的结构较深，平移可变性较差，检测出来的坐标会极度不准确。如果在ResNet中间（图1 conv4与conv5间）加个ROI层结果就不一样了，ROI层提取出的proposal中，有的对应前景label，有的对应背景label，proposal位置的偏移就有可能造成label分类（前景和背景分类）的不同。偏移后原来的前景很有可能变成了背景，原来的背景很有可能变成了前景，换句话说分类loss对proposal的位置是敏感的，这种情况ROI层给深层网络带来了平移可变性。如果把ROI加到ResNet的最后一层（图1 conv5后）结果又是怎样呢？conv5的第一个卷积stride是2，造成conv5输出的feature map更小，这时proposal的一个小偏移在conv5输出上很有可能都感知不到，即proposal对应的label没有改变，所以conv5后虽然有ROI也对平移可变性没有什么帮助，识别出来的位置准确度会很差。

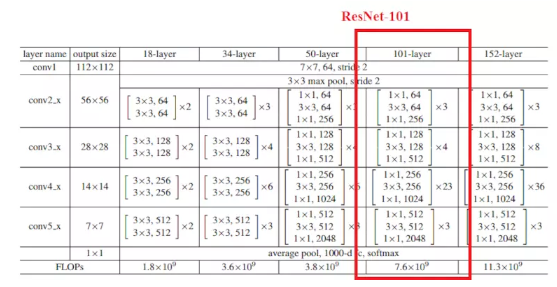


图1：ResNet-101的结构

论文中作者给了测试的数据：ROI放在ResNet-101的conv5后，mAP是68.9%；ROI放到conv5前（就是标准的Faster R-CNN结构）的mAP是76.4%，差距是巨大的，这能证明平移可变性对目标检测的重要性。

**理解难点2：R-FCN结构的由来**

R-FCN要解决的根本问题是Faster R-CNN检测速度慢的问题，速度慢是因为ROI层后的结构对不同的proposal是不共享的，试想下如果有300个proposal，ROI后的全连接网络就要计算300次，这个耗时就太吓人了。所以作者把ROI后的结构往前挪来提升速度，但光是挪动下还不行，ROI在conv5后会引起上节提到的平移可变性问题，必须通过其他方法加强结构的可变性，所以作者就想出了通过添加Position-sensitive score map来达到这个目的。

**理解难点3：Position-sensitive score map的结构**

图1的ResNet-101应用到R-FCN时会把最后的average pool和1000-d fc全连接层都去掉了，仅保留前100层，再新加一个1x1x1024的卷积层用来降维（从2048维降到1024维），和一个很特殊的卷积来生成k2 \* (C+1)维的Position-sensitive score map。其中的C是要分类的类别数，比如PASCAL VOC类别就是20，加上1表示加上一个背景分类；k是之后的ROI Pooling中对ROI区域要划分的小格数，比如论文中k=3就是对ROI在长宽方向各三等分形成9个小区域(如图2)。Position-sensitive score map的值对小区域相对于ROI中的位置很敏感，为什么这么说后面会解释。

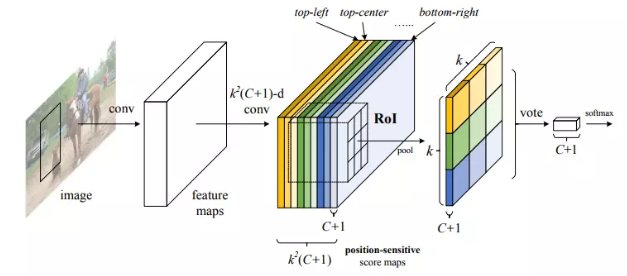


图2：Position-sensitive score map的结构

图2中最后一个特殊卷积输出Position-sensitive score map后，就要做ROI Pooling了，和Faster R-CNN中的ROI Pooling一样要对9个小区域分别进行pooling，要注意的是R-FCN中９个小区域并不是在所有k2 \* (C+1)维度上都做pooling，每个小区域只会在对应的(C+1)个维度上作pooling，比如ROI左上角的区域就在前C+1个维度上pooling，左中位置的区域就在C+２到２C+２间的维度上作pooling，以此类推。pooling后输出的是C+1维度的k\*k数据，每个维度上的k\*k个数据再加到一起(图２的vote过程)形成C+1个单点数据，就代表了C+1个类别的分类概率。  
对于目标定位的输出和上面的分类输出过程类似，只是维度不再是k2 \* (C+1)，而是k2\*4，表示９个小区域的［dx,dy,dw,dh］4个偏移坐标。

**理解难点４：Position-sensitive score map为什么会带来平移可变性**

Position-sensitive score map的概念最早来自另一篇实例分割的论文Instance-sensitive Fully Convolutional Networks ([https://arxiv.org/pdf/1603.08678.pdf](https://link.jianshu.com/?t=https%3A%2F%2Farxiv.org%2Fpdf%2F1603.08678.pdf))。图3是示意图，中间的9张图对应Position-sensitive score map的9个维度的输出。拿左上角的图说明：它的每一个点代表该点正好出现在目标左上角的概率(更准确的说应该是得分，因为还没做softmax)，也可以理解是该点右下方正好是目标的概率。要注意的是：**“目标左上角的概率”的概念并不局限于图中画的绿色框范围，而是整张图上的每一个点**，这是新学习者很容易引起误解的地方。同理其余8张图各自对应了目标正上侧、右上侧、左中侧、正中侧、右中侧、左下侧、正下侧、右下侧的概率。在训练时，一个ROI的9个小区域从每张图的对应区域去Pooling出一个结果，组成新的图(图3右侧的9宫格图)，如果ROI刚好覆盖ground truth，这个新的区域就标记为前景(label=1)。  
这里有个关键点要解释，为什么每张图都能携带相对位置信息？因为从图3提取1~9号小方格时，每个小方格在每张图上的位置并不相同，而是在上下左右方向上有偏移，当组合出来的9宫格对应ground truth时，小方格1就对应了 ground truth左上角的位置，小方格2对应了ground truth正上方的位置，依此类推，所以用这种9宫格训练目标时就有了相对目标位置的信息在里面 。

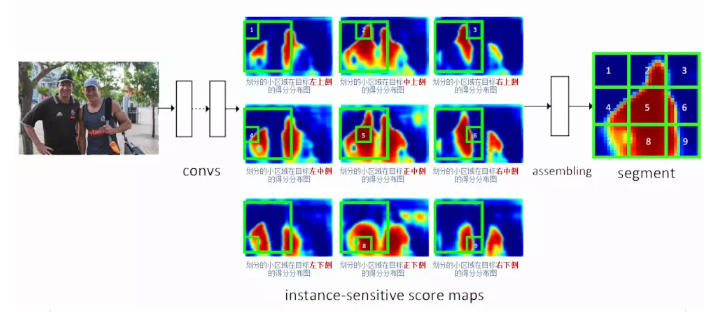


图3：Position-sensitive score map