layout: post
title: "RCNN系列文章之Faster RCNN详解"
date: 2020-07-13
description: "目标检测"

tag: 目标检测

RCNN系列的文章主要是RCNN，Fast RCNN， Faster RCNN， Mask RCNN, Cascade RCNN,这一系列的文章是目标检测two-stage算法的代表，这系列的算法精度高，效果好，是一类重要的方法。

论文地址：Faster R-CNN: Towards Real-Time Object Detectionwith Region Proposal Networks

简要介绍

论文的主要贡献在于提出了**RPN（region proposal network)。**

RPN是一个全卷积网络，能够同时预测每个位置目标的边界（object bound）和目标的得分，fast rcnn使用RPN，通过端到端的训练可以生成高质量的region proposal，将RPN与fast rcnn整合在一起构成了faster rcnn算法，类似于attention的机制，RPN可以告诉算法，where to look。

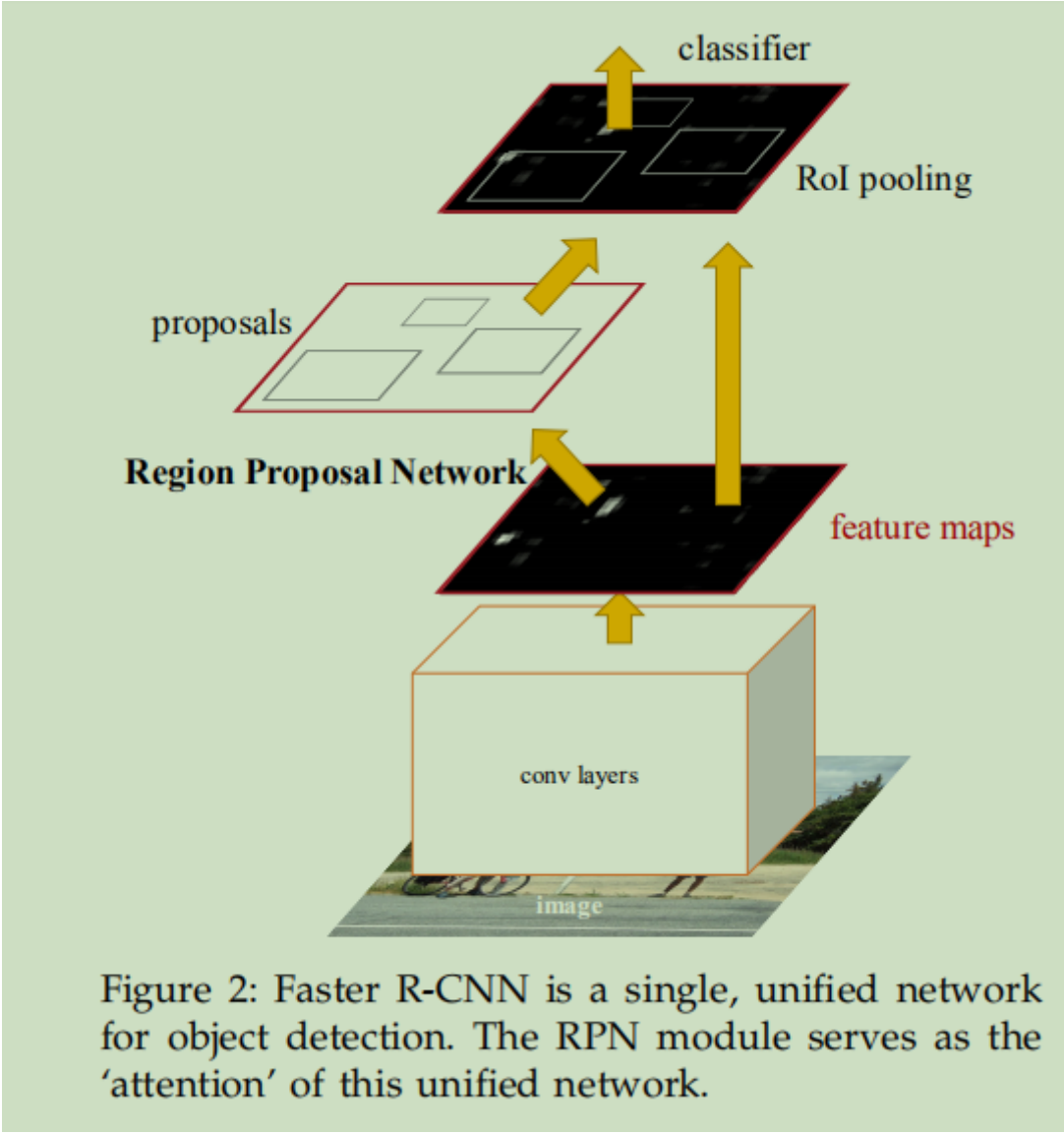


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

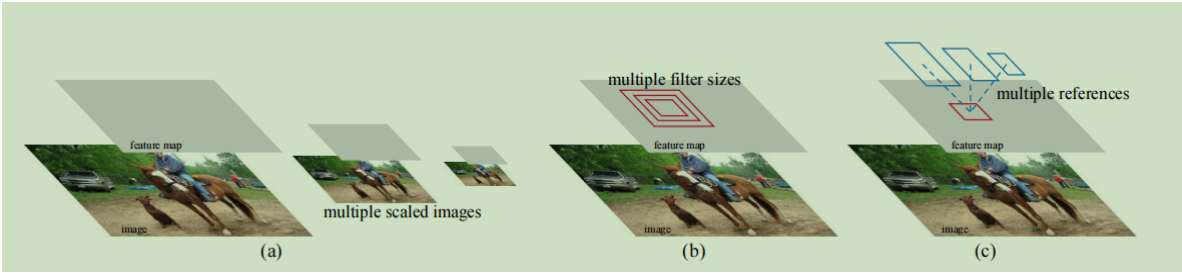
Faster RCNN算法细节介绍

在Fast rcnn中，由于需要使用selective search方法来搜索region proposal，在cpu上实现，每张图片需要2s。大量的时间耗费在region proposal的过程中。

为了解决这个问题，充分使用GPU的高效率与快速的计算，本文提出一种新型的全卷积网络（RPN）来生成region proposal，每张图片达到10ms。

具体的做法就是在基础backbone（用来提取图像特征的cnn）之后加入一些卷积层作为RPN，可以在feature map的每个位置，回归区域的边界与目标的得分。可以进行端到端的训练。

RPN主要是预测具有广泛的尺度与长宽比的region proposal，与使用图像金字塔（图a），filter金字塔（图b）不同的是，我们引入新的"anchor（锚）"盒作为多种尺度和长宽比的参考，这种方案被认为回归参考金字塔（图c），是它避免了枚举多种比例或长宽比的图像或滤波器。这个模型在使用单尺度图像进行训练和测试时运行良好，从而有利于运行速度。



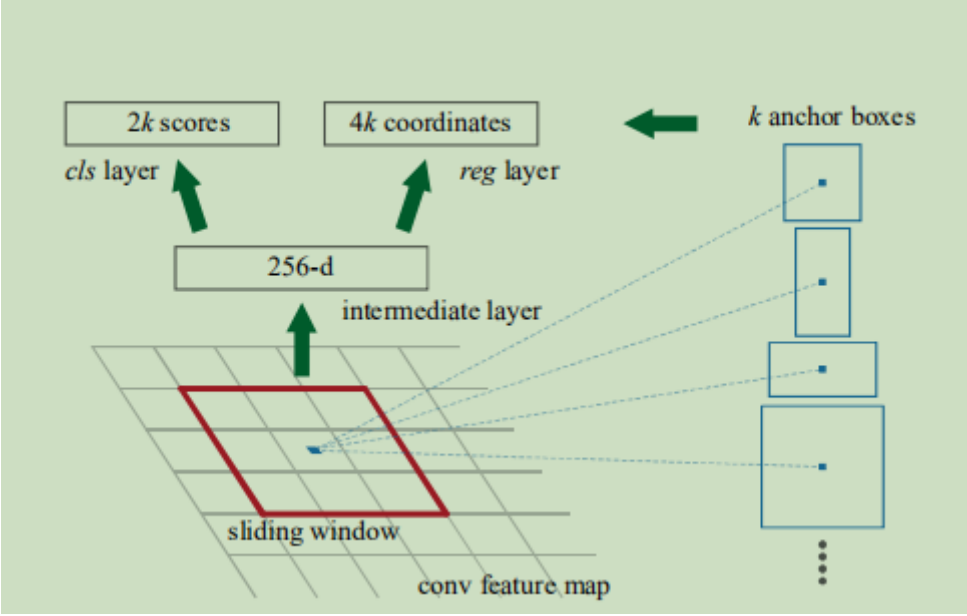
为了将RPN与Fast R-CNN目标检测网络相结合，文章提出了一种训练方案，在微调区域提议任务和保持区域提议的固定微调目标检测之间进行交替。该方案快速收敛，并产生两个任务之间共享的具有卷积特征的统一网络。

Faster RCNN网络包含两个主要的模块，第一个就是**生成区域建议的RPN全卷积网络**， 第二个是**利用region proposal的检测器。**

RPN(Region Proposal Network)

RPN是一个全卷积网络，输入一张图像，输出一系列的矩形目标建议框，每一个框都含有一个目标得分。

为了生成region proposal，我们在最后的共享卷积层输出的卷积特征映射上滑动一个小网络。这个小网络将输入卷积特征映射的n×n空间窗口作为输入。每个滑动窗口映射到一个低维特征（ZF为256维，VGG为512维，后面是ReLU）。这个特征被输入到两个子全连接层——一个边界框回归层（reg）和一个边界框分类层（cls）。在faster rcnn中，n=3，注意输入图像上的有效感受野是大的（ZF和VGG分别为171和228个像素）。下图显示了这个小型网络的一个位置。请注意，因为小网络以滑动窗口方式运行，所有空间位置共享全连接层。这种架构通过一个n×n卷积层，后面是两个子1×1卷积层（分别用于reg和cls）自然地实现。



anchors

在每个滑动窗口位置，我们同时预测多个region proposal，其中每个位置可能提议的最大数目表示为k。因此，*reg*层具有4k个输出，编码k个边界框的坐标，*cls*层输出2k个分数，估计每个proposal是目标或不是目标的概率。锚点位于所讨论的滑动窗口的中心，并与一个尺度和长宽比相关。默认情况下，我们使用3个尺度和3个长宽比，在每个滑动位置产生k=9个锚点。对于大小为W×H（通常约为2400）的卷积特征映射，总共有WHk个锚点。

Translation-Invarient Anchors(平移不变的anchors)

anchor的一个重要特性是它是**平移不变**的，无论是在锚点还是计算相对于锚点的region proposal的函数。如果在图像中平移目标，proposal也应该平移，并且同样的函数应该能够在任一位置预测提议。平移不变特性是由这种anchor的方法保证的。与multiBox比较，MultiBox方法使用k-means生成800个锚点，这不是平移不变的。所以如果平移目标，MultiBox不保证会生成相同的提议。

平移不变特性也减小了模型的大小。MultiBox有(4+1)×800维的全连接输出层，而当k=9个锚点的情况下有(4+2)×9维的卷积输出层。因此，对于VGG-16，输出层具有2.8e4个参数（对于VGG-16为512×(4+2)×9，比MultiBox输出层的6.1e6个参数少了两个数量级（对于MultiBox中的GoogleNet为1536×(4+1)×800）。如果考虑到特征投影层，proposal层仍然比MultiBox少一个数量级。因此这个算法在PASCAL VOC等小数据集上有更小的过拟合风险。

多尺度的锚点作为回归参考

锚点设计提出了一个新的方案来解决多尺度（和长宽比）。多尺度预测有两种流行的方法。第一种方法是基于图像/特征金字塔，例如DPM和基于CNN的方法。图像在多个尺度上进行缩放，并且针对每个尺度计算特征映射（HOG或深卷积特征），这种方法通常是有用的，但是非常耗时。第二种方法是在特征映射上使用多尺度（和/或长宽比）的滑动窗口。例如，在DPM中，使用不同的filter大小（例如5×7和7×5）分别对不同长宽比的模型进行训练。如果用这种方法来解决多尺度问题，可以把它看作是一个“filter金字塔”。第二种方法通常与第一种方法联合采用。

基于锚点方法建立在锚点金字塔上，这是更具成本效益的。这种方法参照多尺度和长宽比的anchor box来分类和回归边界框。它只依赖单一尺度的图像和特征映射，并使用单一尺寸的filter（特征映射上的滑动窗口）

由于这种基于锚点的多尺度设计，可以类似于Fast R-CNN检测器简单地使用在单尺度图像上计算的卷积特征。多尺度锚点设计是共享特征的关键组成部分。

损失函数

为了训练RPN，我们为每个锚点分配一个二值类别标签（含有目标或不含目标）。我们给两种锚点分配一个正标签：（i）具有与实际边界框的重叠最高交并比（IoU）的锚点，或者（ii）具有与实际边界框的重叠超过0.7 IoU的锚点。（注意，单个真实边界框可以为多个锚点分配正标签。通常第二个条件足以确定正样本；但我们仍然采用第一个条件，因为在一些极少数情况下，第二个条件可能找不到正样本，因为可能所有的IOU值均小于0.7）。对于所有的真实边界框，如果一个锚点的IoU低于0.3，我们给这个锚点分配一个负标签（表示背景）。那些既不是正标签也不是负标签的锚点不会有助于训练目标函数。

应用Fast rcnn中的多任务损失函数：

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

默认情况下，我们设置λ=10（权重因子）

对于边界框的回归，采用如下坐标的参数化：

$$\begin{aligned} t_{\mathrm{x}} &= (x - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}} = (y - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}} &= \log(w/w_{\mathrm{a}}), \quad t_{\mathrm{h}} = \log(h/h_{\mathrm{a}}), \\ t_{\mathrm{x}}^* &= (x^* - x_{\mathrm{a}})/w_{\mathrm{a}}, \quad t_{\mathrm{y}}^* = (y^* - y_{\mathrm{a}})/h_{\mathrm{a}}, \\ t_{\mathrm{w}}^* &= \log(w^*/w_{\mathrm{a}}), \quad t_{\mathrm{h}}^* = \log(h^*/h_{\mathrm{a}}), \end{aligned}$$

其中，x，y，w和h表示边界框的中心坐标及其宽和高。变量x，x_a和x^*分别表示预测边界框，锚盒和实际边界框。这可以被认为是从锚盒到邻近的实际边界框的回归。

训练RPN

RPN可以通过反向传播和随机梯度下降（SGD）进行端对端训练[35]。使用"以图像为中心"的采样策略来训练这个网络。每个小批量数据（mini-batch）都从包含许多正标签于负标签的实例锚点的单张图像中产生。对所有锚点的损失函数进行优化是可能的，但是这样会偏向于负样本，因为它们是占主导地位的。取而代之的是，我们在图像中随机采样256个锚点，计算一个小批量数据的损失函数，其中采样的正锚点和负锚点的比率可达1:1。如果图像中的正样本少于128个，我们使用负样本填充mini-batch。

RPN 和 Fast R-CNN特征共享

以上描述了如何训练 RPN 网络，而没有考虑基于 region proposal 的目标检测网络如何利用这些 proposals。对于检测网络，作者采用了 Fast R-CNN。RPN 和、 Fast R-CNN 如果都独立训练，需要单独修改它们的卷积层。因此有需要开发一种允许两个网络间共享卷积层的技术，而不是分别学习两个网络。论文中作者描述了 3 种方法用来实现特征共享：

1. **交替训练。** 首先训练 RPN，用 RPN 输出的 proposals 训练 Fast R-CNN。Fast R-CNN 微调后用于初始化 RPN 网络参数，如此循环迭代。（本文所有实验中使用的解决方案）
2. **近似联合训练。** RPN 和 Fast R-CNN 整合到一个网络里一起训练。在每次SGD迭代中，前向传递生成region proposal，在训练Fast R-CNN检测器将这看作是固定的、预计算的proposal。反向传播像往常一样进行，其中对于共享层，组合来自RPN损失和Fast R-CNN损失的反向传播信号。这个解决方案很容易实现。但是这个解决方案忽略了关于proposal边界框的坐标（也是网络响应）的导数，因此是近似的。在我们的实验中，我们实验发现这个求解器产生了相当的结果，与交替训练相比，训练时间减少了大约25−50%25−50%。（**大多github上开源代码，均采用这种端到端训练方式**）
3. **非近似联合训练。** 如上所述，由RPN预测的边界框也是输入的函数。Fast R-CNN中的RoI池化层接受卷积特征以及预测的边界框作为输入，所以理论上有效的反向传播求解器也应该包括关于边界框坐标的梯度。在上述近似联合训练中，这些梯度被忽略。在一个非近似的联合训练解决方案中，我们需要一个关于边界框坐标可微分的RoI池化层。这是一个重要的问题，可以通过"RoI扭曲"层给出解决方案，这超出了本文的范围。

基于**交替训练法**，作者描述了一种4-step的训练方式：

step 1. 训练 RPN 网络，采用 ImageNet 预训练的模型进行初始化，并进行微调
step 2. 利用 step1 的RPN生成 proposal，由 Fast R-CNN 训练一个单独的检测网络，该网络同样由ImageNet预训练模型进行初始化。此时，两个网络还未共享卷积层
step 3. 用检测网络初始化 RPN 训练，固定共享的卷积层，只微调 RPN 独有的层，现在两个网络实现了卷积层共享
step 4. 保持共享的卷积层固定，微调 Fast R-CNN 的 fc 层。这样，两个网络共享相同的卷积层，构成一个统一的网络

一些细节

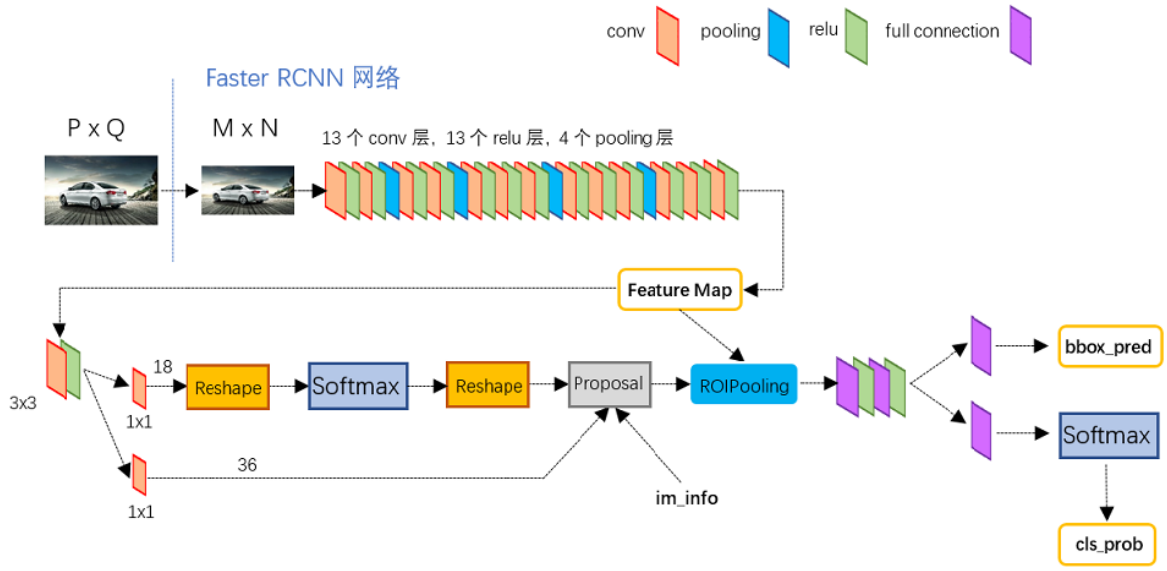
在单尺度图像上训练和测试RPN和目标检测网络。我们重新缩放图像，使得它们的短边是s=600像素。多尺度特征提取（使用图像金字塔）可能会提高精度，但不会表现出速度与精度的良好折衷。在重新缩放的图像上，最后卷积层上的ZF和VGG网络的总步长为16个像素，因此在调整大小（∽500×375）之前，典型的PASCAL图像上的总步长为∽10个像素。即使如此大的步长也能提供良好的效果，尽管步幅更小，精度可能会进一步提高。

对于锚点，使用3个尺度，边界框面积分别为128，256和512个像素，以及1:1，1:2和2:1的长宽比。这些超参数不是针对特定数据集仔细选择的，如上所述，我们的解决方案不需要图像金字塔或滤波器金字塔来预测多个尺度的区域，节省了大量的运行时间。

跨越图像边界的anchor box需要小心处理。在训练过程中，我们忽略了所有的超出图像边界锚点，所以不会造成损失。对于一个典型的1000×600的图片，总共将会有大约20000（≈60×40×9）个锚点。跨界锚点被忽略，每张图像约有6000个锚点用于训练。如果跨界异常值在训练中不被忽略，则会在目标函数中引入大的，难以纠正的误差项，且训练不会收敛。但在测试过程中，我们仍然将全卷积RPN应用于整张图像。这可能会产生跨边界的提议边界框，我们剪切到图像边界。

一些RPN proposal互相之间高度重叠。为了减少冗余，我们在proposal区域根据他们的*cls*分数采取非极大值抑制（NMS）。我们将NMS的IoU阈值固定为0.7，这就给每张图像留下了大约2000个提议区域。正如我们将要展示的那样，NMS不会损害最终的检测准确性，但会大大减少proposal的数量。在NMS之后，我们使用前N个proposal区域来进行检测。接下来，我们使用2000个RPN proposal对Fast R-CNN进行训练，但在测试时评估不同数量的proposal。

附上一张图，不知道在哪个地方找的，我找不到链接注明了，侵删



[更多技术文章请点击查看]