[NIPS 2015]

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

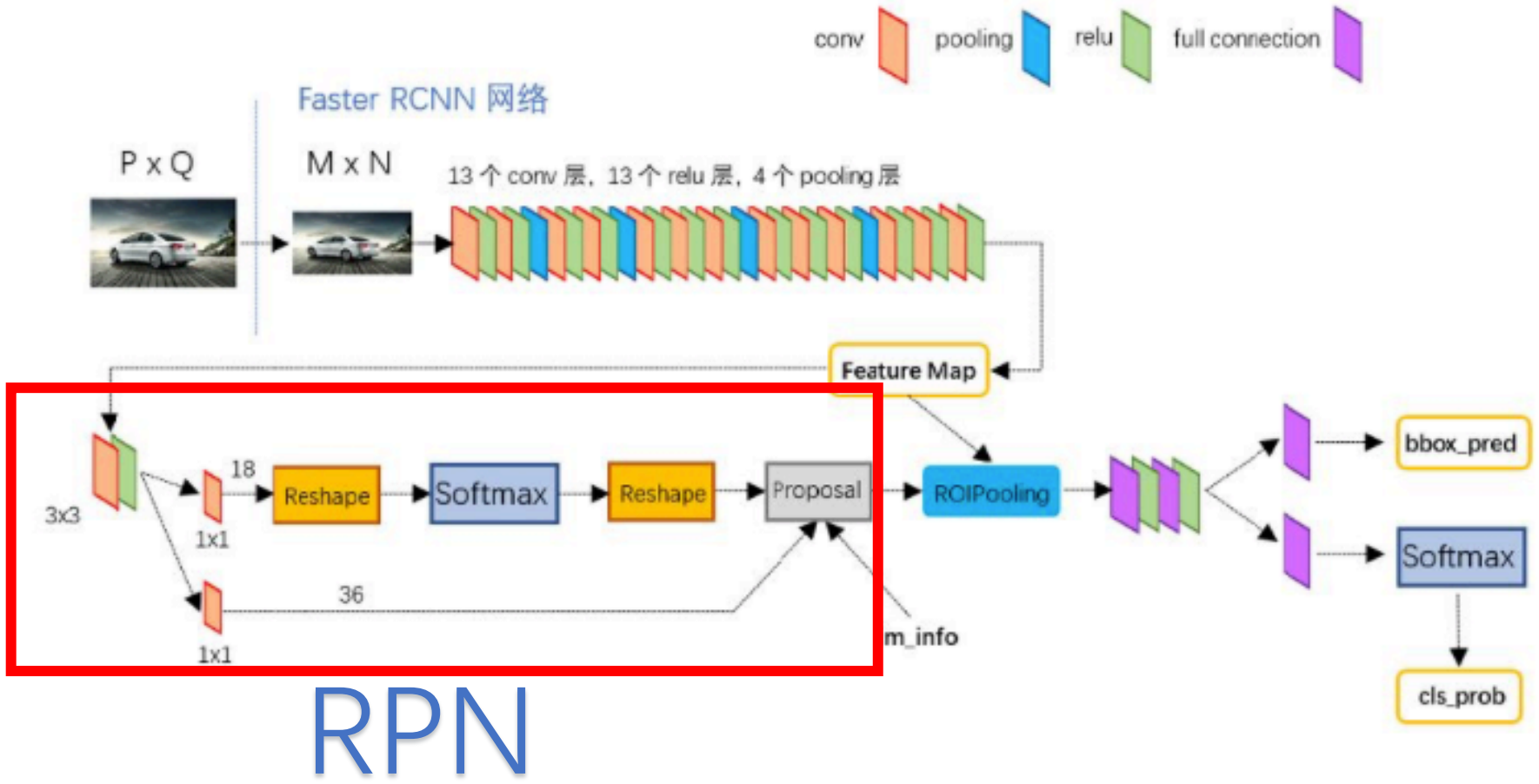
Shaoqing Ren*. Kaiming He Ross Girshick. Jian Sun.
Microsoft Research.

Abstract

- 最新的物体检测网络采用region proposal 算法来假设目标的位置。(指当时)
- 提出RPN网络，与检测网络共享整个图片的卷积特征，花费很小。
- 全卷积RPN，可以在特征图的每个位置预测目标框的位置和得分。
- 采用VGG-16模型，在GPU上检测帧率达到5fps。
- 精度：VOC 2007（73.2%mAP） VOC 2012 (70.4%mAP)。

网络结构

卷积层提取feature map->RPN网络做定位->ROIPooling->分类



Region Proposal Networks(RPN)

- 输入：任何尺寸
- 输出：一系列矩形目标->proposals
- 网络特点：全卷积网络
- 解释：以上述网络结构为例，RPN网络前接四层pooling层，均1/2下采样，所以特征图大小为M/16 * N/16，通道数为512-d（VGG）。RPN网络的3*3卷积核不改变特征图大小和通道数（也就是文章中的滑动窗，其实就是接一层卷积层），分别通过两个1*1卷积层，卷积核个数分别为18（后接了softmax层）和36，其中18=（2个概率（对象/非对象的概率）*9个anchor boxes），36=（4个coordinates（tx ty th tw） * 9 个anchor boxes）。

Anchor box

- 针对每个滑动窗口（换言之3*3卷积后的特征图每个点），选取k=9个anchors（3个不同尺寸，每个尺寸下3个不同长宽比）。每个anchor均位于对应滑动窗口中心。
- 具有平移不变性。因此每个滑动窗的输出只需要（4+2）*9输出即可。相比k聚类的MultiBox则需要（4+1）*800的输出。

RPN Loss Function

- 对每个anchor分配一个标签（是否存在目标）

下面两种情况为正标签（即存在目标）：
1. 与ground truth的iou最高的anchor
2. 与ground truth的iou超过阈值的anchor（文中为0.7）
（一个ground truth可能给多个anchor贴上正标签）

下面情况为负标签（即不存在目标）：
Iou小于阈值的anchor（文中为0.3）

既不是正标签也不是负标签的anchor对训练不产生影响（不计算损失）。

RPN Loss Function

表示只对真实正样本的anchor(i)计算坐标损失，同yolo类似

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

Smooth L1 loss

- i: anchor索引
- pi : anchor（i）是正标签的概率
- Pi* ：anchor(i) ground truth是正样本概率（1 or 0）
- ti : 预测的四个坐标参数（tx ty th tw）
- ti* : ground truth的坐标参数(tx* ty* th* tw*)
- Ncls Nreg 归一化参数，λ 两项损失的参数

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \quad t_w = \log(w/w_a), \quad t_h = \log(h/h_a),$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \quad t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a),$$

- x (预测坐标x) xa(anchor的坐标x) x*(ground truth坐标x)

优化

- SGD
- 采样256个anchors（正负样本各128）->正负样本平衡

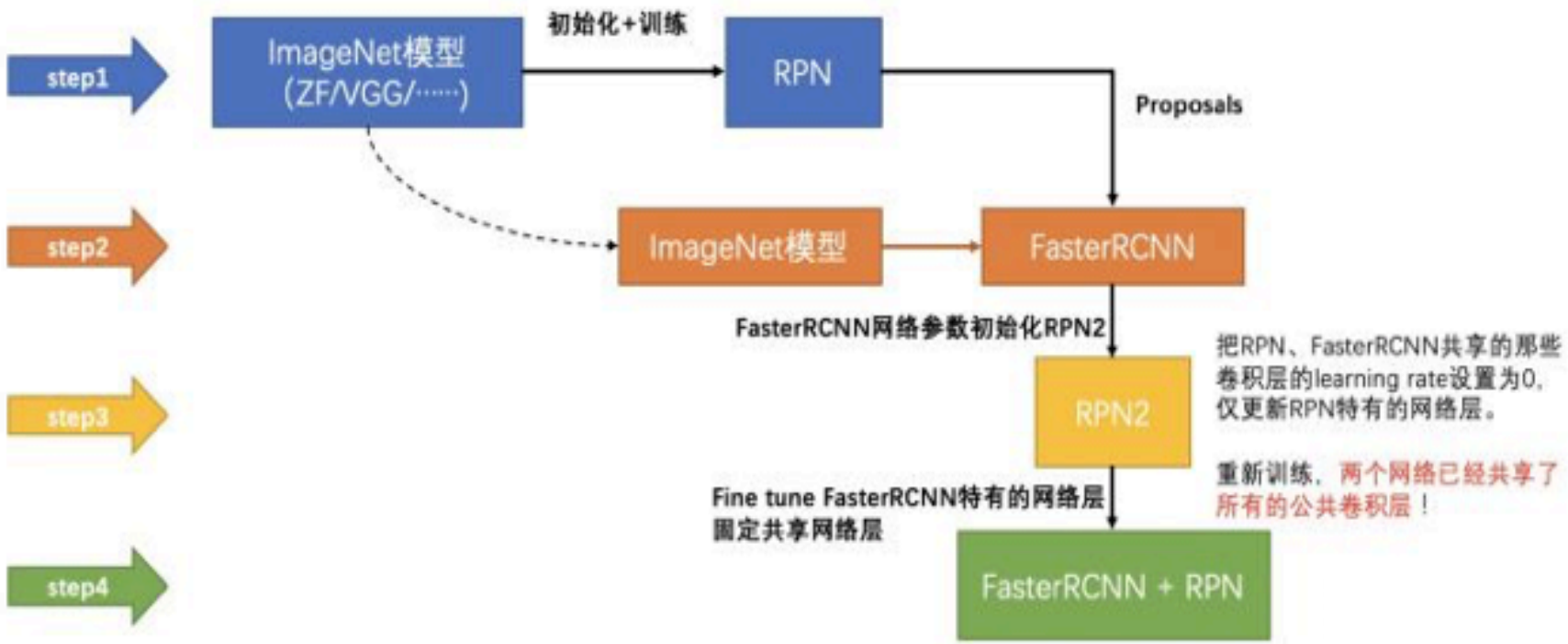
优化

- SGD
- 采样256个anchors（正负样本各128）->正负样本平衡

ROIPooling

- Proposals输出对应原图M*N的尺寸框，大小各异
- Feature map 对应下采样16倍后的特征图
- 先将proposal输出框映射到特征图尺寸上，然后将框分成pool_w*pool_h尺寸的网格，在特征图上最大池化，变成固定尺寸pool_w*pool_h大小（即无论回归后的框多大，都池化到固定尺寸）然后送到全连接层做分类。同时送到全连接层，将proposal再次回归，得到更高精度的rect box。

训练过程


step1
ImageNet模型
(ZF/VGG/……)
初始化+训练
RPN
Proposals
step2
ImageNet模型
FasterRCNN
FasterRCNN网络参数初始化RPN2
把RPN、FasterRCNN共享的那些
卷积层的learning rate设置为0，
仅更新RPN特有的网络层。
step3
RPN2
重新训练，两个网络已经共享了
所有的公共卷积层！
Fine tune FasterRCNN特有的网络层
固定共享网络层
step4
FasterRCNN + RPN

References

- R. Girshick. Fast R-CNN. arXiv:1504.08083, 2015.
- K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.

Code

- Python: https://github.com/rbgirshick/py-faster-rcnn
- Matlab: https://github.com/ShaoqingRen/faster_rcnn