[Faster R-CNN详解 2](#_Toc1477449482)

[摘要 2](#_Toc1685431271)

[一、引言 2](#_Toc56758983)

[二、相关工作 4](#_Toc1033862134)

[三、区域建议网络 5](#_Toc1311708329)

[1、平移不变的anchor 6](#_Toc1257772350)

[2、学习区域建议的损失函数 7](#_Toc80209403)

# Faster R-CNN详解

## 摘要

目前最先进的目标检测网络需要先用区域建议算法推测目标位置，像SPPnet和Fast R-CNN这些网络已经减少了检测网络的运行时间，这时计算区域建议就成了瓶颈问题。本文中，我们介绍一种区域建议网络（Region Proposal Network, RPN），它和检测网络共享全图的卷积特征，使得区域建议几乎不花时间。RPN是一个全卷积网络，在每个位置同时预测目标边界和objectness得分。RPN是端到端训练的，生成高质量区域建议框，用于Fast R-CNN来检测。通过一种简单的交替运行优化方法，RPN和Fast R-CNN可以在训练时共享卷积特征。对于非常深的VGG-16模型，我们的检测系统在GPU上的帧率为5fps（包含所有步骤），在PASCAL VOC 2007和PASCAL VOC 2012上实现了最高的目标检测准确率（2007是73.2%mAP，2012是70.4%mAP），每个图像用了300个建议框。[代码](https://github.com/ShaoqingRen/faster_rcnn)已公开。

## 一、引言

最近在目标检测中取得的进步都是由区域建议方法和基于区域的卷积神经网络（R-CNN）取得的成功来推动的。基于区域的CNN刚提出时在计算上消耗很大，幸好后来这个消耗通过建议框之间共享卷积大大降低了。最近的Fast R-CNN用非常深的网络实现了近实时检测的速率，注意它忽略了生成区域建议框的时间。现在，建议框是最先进的检测系统中的计算瓶颈。

区域建议方法典型地依赖于消耗小的特征和经济的获取方案。选择性搜索（Selective Search, SS）是最流行的方法之一，它基于设计好的低级特征贪心地融合超级像素。与高效检测网络相比，SS要慢一个数量级，CPU应用中大约每个图像2s。EdgeBoxes在建议框质量和速度之间做出了目前最好的权衡，大约每个图像0.2s。但无论如何，区域建议步骤花费了和检测网络差不多的时间。

Fast R-CNN利用了GPU，而区域建议方法是在CPU上实现的，这个运行时间的比较是不公平的。一种明显提速生成建议框的方法是在GPU上实现它，这是一种工程上很有效的解决方案，但这个方法忽略了其后的检测网络，因而也错失了共享计算的重要机会。

本文中，我们改变了算法——用深度网络计算建议框——这是一种简洁有效的解决方案，建议框计算几乎不会给检测网络的计算带来消耗。为了这个目的，我们介绍新的区域建议网络（Region Proposal Networks, RPN），它与最先进的目标检测网络共享卷积层。在测试时，通过共享卷积，计算建议框的边际成本是很小的（例如每个图像10ms）。

我们观察发现，基于区域的检测器例如Fast R-CNN使用的卷积（conv）特征映射，同样可以用于生成区域建议。我们紧接着这些卷积特征增加两个额外的卷积层，构造RPN：第一层把每个卷积映射位置编码为一个短的（例如256-d）特征向量，第二层在每个卷积映射位置，输出这个位置上多种尺度和长宽比的k个区域建议的objectness得分和回归边界（k=9是典型值）。

我们的RPN是一种全卷积网络（fully-convolutional network, FRN），可以针对生成检测建议框的任务端到端地训练。为了统一RPN和Fast R-CNN目标检测网络，我们提出一种简单的训练方案，即保持建议框固定，微调区域建议和微调目标检测之间交替进行(这句翻译的是屎).这个方案收敛很快，最后形成可让两个任务共享卷积特征的标准网络。

我们在PASCAL VOC检测标准集[4]上评估我们的方法， fast R-CNN结合RPN的检测准确率超过了作为强大基准的fast R-CNN结合SS的方法。同时，我们的方法没有了SS测试时的计算负担，对于生成建议框的有效运行时间只有10毫秒。利用[19]中网络非常深的深度模型，我们的检测方法在GPU上依然有5fps的帧率（包括所有步骤），因此就速度和准确率（PASCAL VOC 2007上是73.2%mAP，PASCAL VOC 2012上是70.4%）而言，这是一个实用的目标检测系统。代码已公开。

## 二、相关工作

最近几篇文章中提出了用深度网络定位类确定或类不确定的包围盒[21, 18, 3, 20] 的方法。在OverFeat方法[18]中，训练全连接（fc）层，对假定只有一个目标的定位任务预测包围盒坐标。fc层再转入卷积层来检测多个类确定的目标。MultiBox方法[3, 20]从最后一个fc层同时预测多个（如800）包围盒的网络中生成区域建议，R-CNN[6]就是用的这个。他们的建议框网络应用于单个图像或多个大图像的切割部分（如224x224）[20]。我们在后文中讲我们的方法时会更深层次地讨论OverFeat和MultiBox。

卷积的共享计算[18, 7, 2, 5]高效、精确，已经在视觉识别方面吸引了越来越多的注意。OverFeat论文[18]从图像金字塔计算卷积特征，用于分类、定位、检测。在共享的卷积特征映射上自适应大小的pooling（SPP）[7]能有效用于基于区域的目标检测[7, 16]和语义分割[2]。Fast R-CNN[5]实现了在共享卷积特征上训练的端到端检测器，显示出令人惊叹的准确率和速度。

## 三、区域建议网络

区域建议网络（RPN）将一个图像（任意大小）作为输入，输出矩形目标建议框的集合，每个框有一个objectness得分。我们用全卷积网络[14]对这个过程构建模型，本章会详细描述。因为我们的最终目标是和Fast R-CNN目标检测网络[15]共享计算，所以假设这两个网络共享一系列卷积层。在实验中，我们详细研究Zeiler和Fergus的模型[23]（ZF），它有5个可共享的卷积层，以及Simonyan和Zisserman的模型[19]（VGG），它有13个可共享的卷积层。

为了生成区域建议框，我们在最后一个共享的卷积层输出的卷积特征映射上滑动小网络，这个网络全连接到输入卷积特征映射的nxn的空间窗口上。每个滑动窗口映射到一个低维向量上（对于ZF是256-d，对于VGG是512-d，每个特征映射的一个滑动窗口对应一个数值）。这个向量输出给两个同级的全连接的层——包围盒回归层（reg）和包围盒分类层（cls）。本文中n=3，注意图像的有效感受野很大（ZF是171像素，VGG是228像素）。图1（左）以这个小网络在某个位置的情况举了个例子。注意，由于小网络是滑动窗口的形式，所以全连接的层（nxn的）被所有空间位置共享（指所有位置用来计算内积的nxn的层参数相同）。这种结构实现为nxn的卷积层，后接两个同级的1x1的卷积层（分别对应reg和cls），ReLU[15]应用于nxn卷积层的输出。

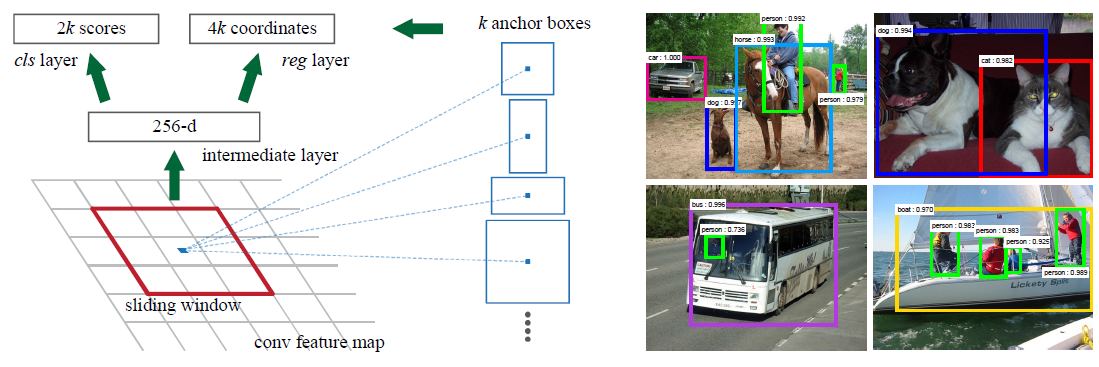


图1：左：区域建议网络（RPN）。右：用RPN建议框在PASCAL VOC 2007测试集上的检测实例。我们的方法可以在很大范围的尺度和长宽比中检测目标。

### 1、平移不变的anchor

在每一个滑动窗口的位置，我们同时预测k个区域建议，所以reg层有4k个输出，即k个box的坐标编码。cls层输出2k个得分，即对每个建议框是目标/非目标的估计概率（为简单起见，是用二类的softmax层实现的cls层，还可以用logistic回归来生成k个得分）。k个建议框被相应的k个称为anchor的box参数化。每个anchor以当前滑动窗口中心为中心，并对应一种尺度和长宽比，我们使用3种尺度和3种长宽比，这样在每一个滑动位置就有k=9个anchor。对于大小为WxH（典型值约2,400）的卷积特征映射，总共有WHk个anchor。我们的方法有一个重要特性，就是平移不变性，对anchor和对计算anchor相应的建议框的函数而言都是这样。

作为比较，MultiBox方法[20]用k-means生成800个anchor，但不具有平移不变性。如果平移了图像中的目标，建议框也应该平移，也应该能用同样的函数预测建议框。此外，因为MultiBox的anchor不具有平移不变性，所以它需要（4+1）x800－d的输出层，而我们的方法只要（4+2）x9-d的输出层。我们的建议框层少一个数量级的参数（MultiBox用GoogleLeNet[20]需要2700万vs.RPN用VGG-16需要240万），这样在PASCAL VOC这种小数据集上出现过拟合的风险较小。

### 2、学习区域建议的损失函数

为了训练RPN，我们给每个anchor分配一个二进制的标签（是不是目标）。我们分配正标签给两类anchor：（i）与某个ground truth（GT）包围盒有最高的IoU（Intersection-over-Union，交集并集之比）重叠的anchor（也许不到0.7），（ii）与任意GT包围盒有大于0.7的IoU交叠的anchor。注意到一个GT包围盒可能分配正标签给多个anchor。我们分配负标签给与所有GT包围盒的IoU比率都低于0.3的anchor。非正非负的anchor对训练目标没有任何作用。

有了这些定义，我们遵循Fast R-CNN[5]中的多任务损失，最小化目标函数。我们对一个图像的损失函数定义为：

fsdrf

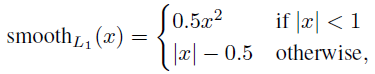
这里，i是一个mini-batch中anchor的索引，Pi是anchor i是目标的预测概率。如果anchor为正，GT标签Pi\* 就是1，如果anchor为负，Pi\* 就是0。ti是一个向量，表示预测的包围盒的4个参数化坐标，ti\* 是与正anchor对应的GT包围盒的坐标向量。分类损失\*Lcls是两个类别（目标vs.非目标）的对数损失

gnfgc

对于回归损失\*，我们用

hjg

来计算，其中R是[5]中定义的鲁棒的损失函数（smooth L1）。



Pi\* Lreg这一项意味着只有正anchor（Pi\* =1）才有回归损失，其他情况就没有（Pi\* =0）。cls层和reg层的输出分别由{pi}和{ti}组成，这两项分别由Ncls和Nreg以及一个平衡权重λ归一化（早期实现及公开的代码中，λ=10，cls项的归一化值为mini-batch的大小，即Ncls=256，reg项的归一化值为anchor位置的数量，即Nreg~2,400，这样cls和reg项差不多是等权重的。

对于回归，我们学习[6]采用4个坐标：

fhr

x，y，w，h指的是包围盒中心的（x, y）坐标、宽、高。变量x，xa，x\*分别指预测的包围盒、anchor的包围盒、GT的包围盒（对y，w，h也是一样）的x坐标。可以理解为从anchor包围盒到附近的GT包围盒的包围盒回归。

无论如何，我们用了一种与之前的基于特征映射的方法[7, 5]不同的方法实现了包围盒算法。在[7, 5]中，包围盒回归在从任意大小的区域中pooling到的特征上执行，回归权重是所有不同大小的区域共享的。在我们的方法中，用于回归的特征在特征映射中具有相同的空间大小（nxn）。考虑到各种不同的大小，需要学习一系列k个包围盒回归量。每一个回归量对应于一个尺度和长宽比，k个回归量之间不共享权重。因此，即使特征具有固定的尺寸/尺度，预测各种尺寸的包围盒仍然是可能的。

### 3、优化（训练）

RPN很自然地实现为全卷积网络[14]，通过反向传播和随机梯度下降（SGD）[12]端到端训练。我们遵循[5]中的“image-centric”采样策略训练这个网络。每个mini-batch由包含了许多正负样本的单个图像组成。我们可以优化所有anchor的损失函数，但是这会偏向于负样本，因为它们是主要的。因此，我们随机地在一个图像中采样256个anchor，计算mini-batch的损失函数，其中采样的正负anchor的比例是1:1。如果一个图像中的正样本数小于128，我们就用负样本填补这个mini-batch。

我们通过从零均值标准差为0.01的高斯分布中获取的权重来随机初始化所有新层（最后一个卷积层其后的层），所有其他层（即共享的卷积层）是通过对ImageNet分类[17]预训练的模型来初始化的，这也是标准惯例[6]。我们调整ZF网络的所有层，以及conv3\_1，并为VGG网络做准备，以节约内存[5]。我们在PASCAL数据集上对于60k个mini-batch用的学习率为0.001，对于下一20k个mini-batch用的学习率是0.0001。动量是0.9，权重衰减为0.0005[11]。我们的实现使用了Caffe[10]。

### 4、区域建议与目标检测共享卷积特征

迄今为止，我们已经描述了如何为生成区域建议训练网络，而没有考虑基于区域的目标检测CNN如何利用这些建议框。对于检测网络，我们采用Fast R-CNN[5]，现在描述一种算法，学习由RPN和Fast R-CNN之间共享的卷积层。

RPN和Fast R-CNN都是独立训练的，要用不同方式修改它们的卷积层。因此我们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。注意到这不是仅仅定义一个包含了RPN和Fast R-CNN的单独网络，然后用反向传播联合优化它那么简单。原因是Fast R-CNN训练依赖于固定的目标建议框，而且并不清楚当同时改变建议机制时，学习Fast R-CNN会不会收敛。虽然这种联合优化在未来工作中是个有意思的问题，我们开发了一种实用的4步训练算法，通过交替优化来学习共享的特征。

第一步，我们依上述训练RPN，该网络用ImageNet预训练的模型初始化，并端到端微调用于区域建议任务。

第二步，我们利用第一步的RPN生成的建议框，由Fast R-CNN训练一个单独的检测网络，这个检测网络同样是由ImageNet预训练的模型初始化的，这时候两个网络还没有共享卷积层。

第三步，我们用检测网络初始化RPN训练，但我们固定共享的卷积层，并且只微调RPN独有的层，现在两个网络共享卷积层了。

第四步，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。

## 四、实现细节

我们训练、测试区域建议和目标检测网络都是在单一尺度的图像上[7, 5]。我们缩放图像，让它们的短边s=600像素[5]。多尺度特征提取可能提高准确率但是不利于速度与准确率之间的权衡[5]。我们也注意到ZF和VGG网络，对缩放后的图像在最后一个卷积层的总步长为16像素，这样相当于一个典型的PASCAL图像（~500x375）上大约10个像素（600/16=375/10）。即使是这样大的步长也取得了好结果，尽管若步长小点准确率可能得到进一步提高。

对于anchor，我们用3个简单的尺度，包围盒面积为128x128，256x256，512x512，和3个简单的长宽比，1:1，1:2，2:1。注意到，在预测大建议框时，我们的算法考虑了使用大于基本感受野的anchor包围盒。这些预测不是不可能——只要看得见目标的中间部分，还是能大致推断出这个目标的范围。通过这个设计，我们的解决方案不需要多尺度特征或者多尺度滑动窗口来预测大的区域，节省了相当多的运行时间。图1（右）显示了我们的算法处理多种尺度和长宽比的能力。下表是用ZF网络对每个anchor学到的平均建议框大小（s=600）。

2018-03-13 15-43-07 的屏幕截图

跨越图像边界的anchor包围盒要小心处理。在训练中，我们忽略所有跨越图像边界的anchor，这样它们不会对损失有影响。对于一个典型的1000x600的图像，差不多总共有20k（~60x40x9）anchor。忽略了跨越边界的anchor以后，每个图像只剩下6k个anchor需要训练了。如果跨越边界的异常值在训练时不忽略，就会带来又大又困难的修正误差项，训练也不会收敛。在测试时，我们还是应用全卷积的RPN到整个图像中，这可能生成跨越边界的建议框，我们将其裁剪到图像边缘位置。

有些RPN建议框和其他建议框大量重叠，为了减少冗余，我们基于建议区域的cls得分，对其采用非极大值抑制（non-maximum suppression, NMS）。我们固定对NMS的IoU阈值为0.7，这样每个图像只剩2k个建议区域。正如下面展示的，NMS不会影响最终的检测准确率，但是大幅地减少了建议框的数量。NMS之后，我们用建议区域中的top-N个来检测。在下文中，我们用2k个RPN建议框训练Fast R-CNN，但是在测试时会对不同数量的建议框进行评价。

## 五、备注

**1、RCNN系列方法的介绍**

RCNN算法分为4个步骤：

（1）一张图像生成1K~2K个候选区域（采用SS方法）

（2）对每个候选区域，使用深度网络提取特征

（3）特征送入每一类的SVM 分类器，判别是否属于该类

（4）使用回归器精细修正候选框位置

**2、位置精修**

目标检测问题的衡量标准是重叠面积：许多看似准确的检测结果，往往因为候选框不够准确，重叠面积很小。故需要一个位置精修步骤。

**回归器**

对每一类目标，使用一个线性脊回归器进行精修。正则项λ=10000。 输入为深度网络pool5层的4096维特征，输出为xy方向的缩放和平移。

**训练样本**

判定为本类的候选框中，和真值重叠面积大于0.6的候选框。 可以看出该网络重复计算量很大，2K个候选框单独用CNN提取特征，再分类！

**3、Faster RCNN**

从RCNN到fast RCNN，再到本文的faster RCNN，目标检测的四个基本步骤（候选区域生成，特征提取，分类，位置精修）终于被统一到一个深度网络框架之内。所有计算没有重复，完全在GPU中完成，大大提高了运行速度。

faster RCNN可以简单地看做“区域生成网络+fast RCNN“的系统，用区域生成网络代替fast RCNN中的Selective Search方法。本篇论文着重解决了这个系统中的三个问题：

1. 如何设计区域生成网络？

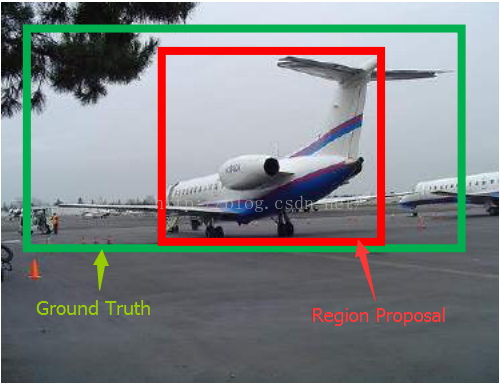
2. 如何训练区域生成网络？

3. 如何让区域生成网络和fast RCNN网络共享特征提取网络？

**4、R-CNN中的boundingbox回归**

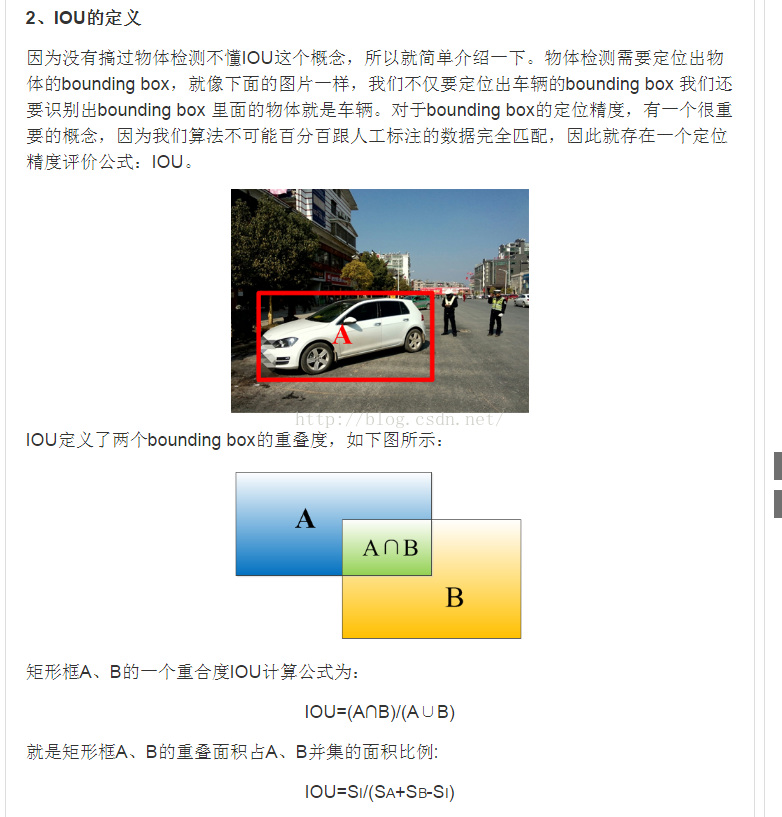
下面先介绍R-CNN和Fast R-CNN中所用到的边框回归方法。

（1）为什么要做Bounding-box regression？

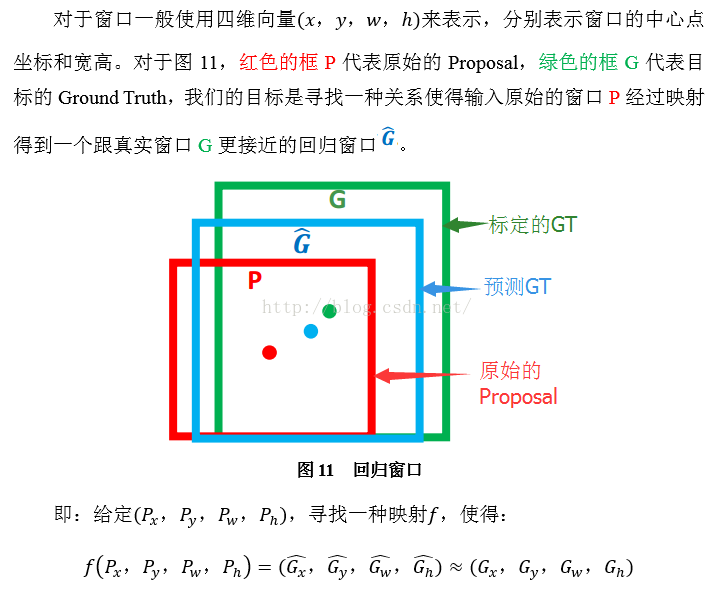


如上图所示，绿色的框为飞机的Ground Truth，红色的框是提取的Region Proposal。那么即便红色的框被分类器识别为飞机，但是由于红色的框定位不准(IoU<0.5)，那么这张图相当于没有正确的检测出飞机。如果我们能对红色的框进行微调，使得经过微调后的窗口跟Ground Truth更接近，这样岂不是定位会更准确。确实，Bounding-box regression 就是用来微调这个窗口的。

（2）什么是IOU

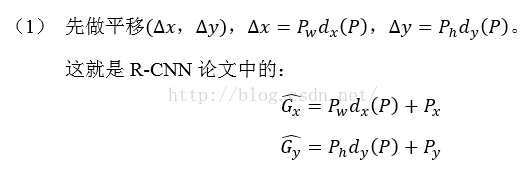


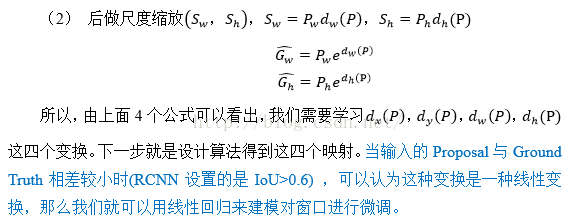
（3）回归/微调的对象是什么？



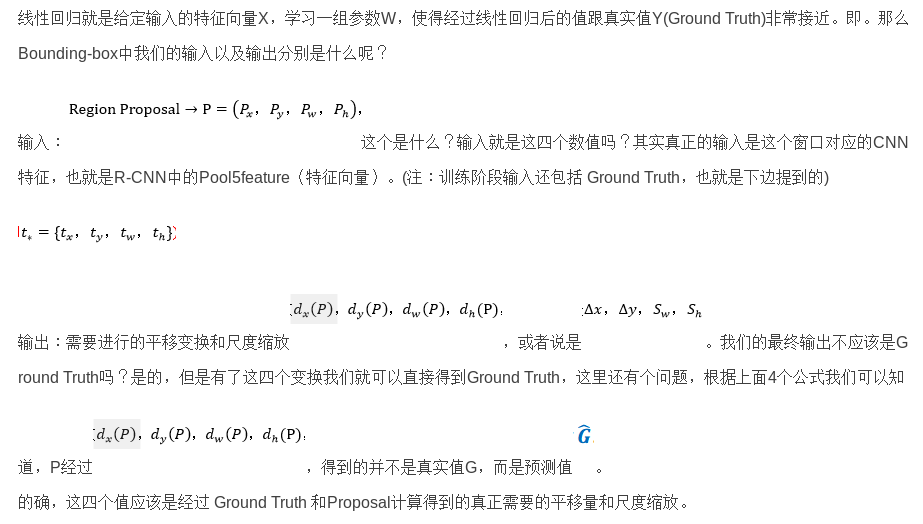
（4）Bounding-box regression（边框回归）

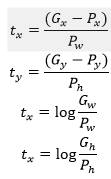
那么经过何种变换才能从图11中的窗口P变为窗口呢？比较简单的思路就是：

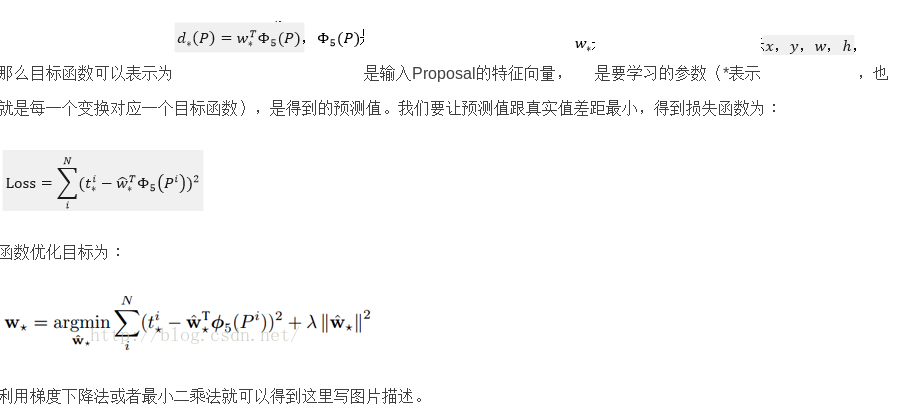




注意：只有当Proposal和Ground Truth比较接近时（线性问题），我们才能将其作为训练样本训练我们的线性回归模型，否则会导致训练的回归模型不work（当Proposal跟GT离得较远，就是复杂的非线性问题了，此时用线性回归建模显然不合理）。这个也是G-CNN: an Iterative Grid Based Object Detector多次迭代实现目标准确定位的关键。







5、anchor机制

anchor是根据nxn区域的中心点，推算出对应原图在这个中心点的k种不同大小的可能区域。

之前的理解错了，用anchor机制只是为了得到k个区域的监督信息（区域的坐标和前景背景标签）

从nxn提出的256d特征是被这k种区域共享的，在clc layer和reg layer计算损失的时候，用这共享的256d特征 加上 anchor推算出k种区域的坐标和前景、背景的标签，便可以对这k种区域同时计算loss。

clc layer和reg layer同时预测k个区域的前景、背景概率（1个区域2个scores，所以是2k个scores），以及bounding box（1个区域4个coordinates，所以是4k个coordinates），具体的说：

clc layer输出预测区域的2个参数，即预测为前景的概率pa和pb，损失用softmax loss（cross entropy loss）（本来还以为是sigmoid，这样的话只预测pa就可以了？）。需要的监督信息是Y=0,1，表示这个区域是否ground truth

reg layer输出预测区域的4个参数：x,y,w,h，用smooth L1 loss。需要的监督信息是 anchor的区域坐标{xa,ya,wa,ha} 和 ground truth的区域坐标{x\*,y\*,w\*,h\*}