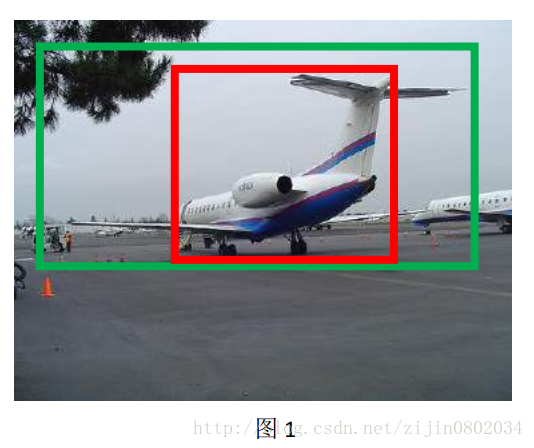
**边框回归(Bounding Box Regression)详解**

最近一直看检测有关的Paper, 从rcnn， fast rcnn, faster rcnn, yolo, r-fcn, ssd，到今年cvpr最新的yolo9000。这些paper中损失函数都包含了边框回归，除了rcnn详细介绍了，其他的paper都是一笔带过，或者直接引用rcnn就把损失函数写出来了。前三条网上解释比较多，后面的两条我看了很多paper，才得出这些结论。

1. **为什么要边框回归？**

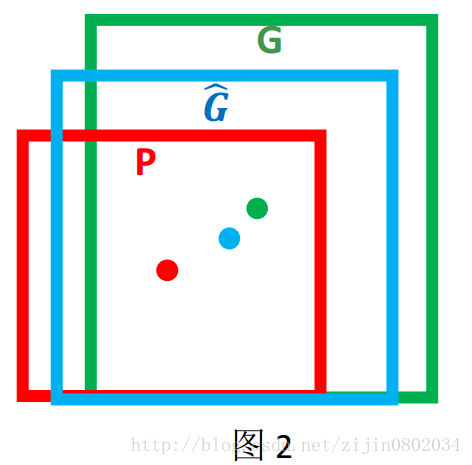
这里引用王斌师兄的理解，如下图所示：



对于上图，绿色的框（最外围）表示Ground Truth, 红色的框（内围）为Selective Search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。

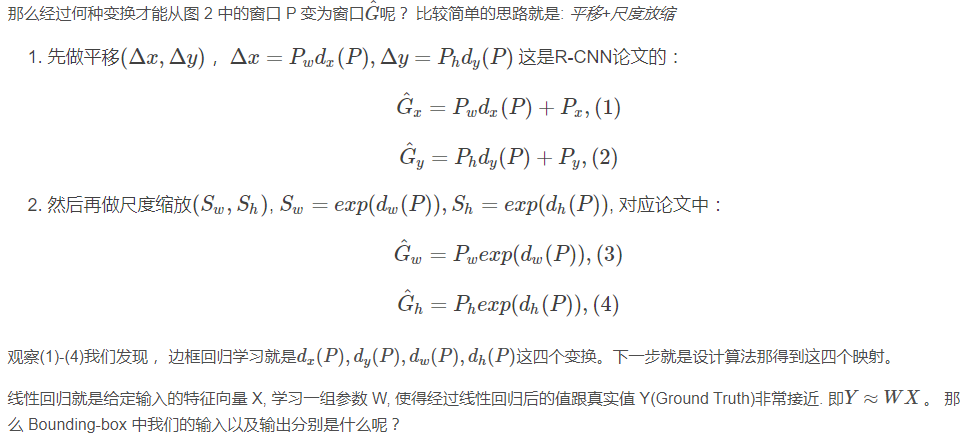
1. **边框回归是什么？**

继续借用师兄的理解：对于窗口一般使用四维向量(x,y,w,h) 来表示， 分别表示窗口的中心点坐标和宽高。 对于图 2, 红色的框 P 代表原始的Proposal, 绿色的框 G 代表目标的 Ground Truth， 我们的目标是寻找一种关系使得输入原始的窗口 P 经过映射得到一个跟真实窗口 G 更接近的回归窗口。



边框回归的目的既是：给定(Px,Py,Pw,Ph)寻找一种映射f， 使得f(Px,Py,Pw,Ph)=(,,,) 并且(,,,)≈(,,,)

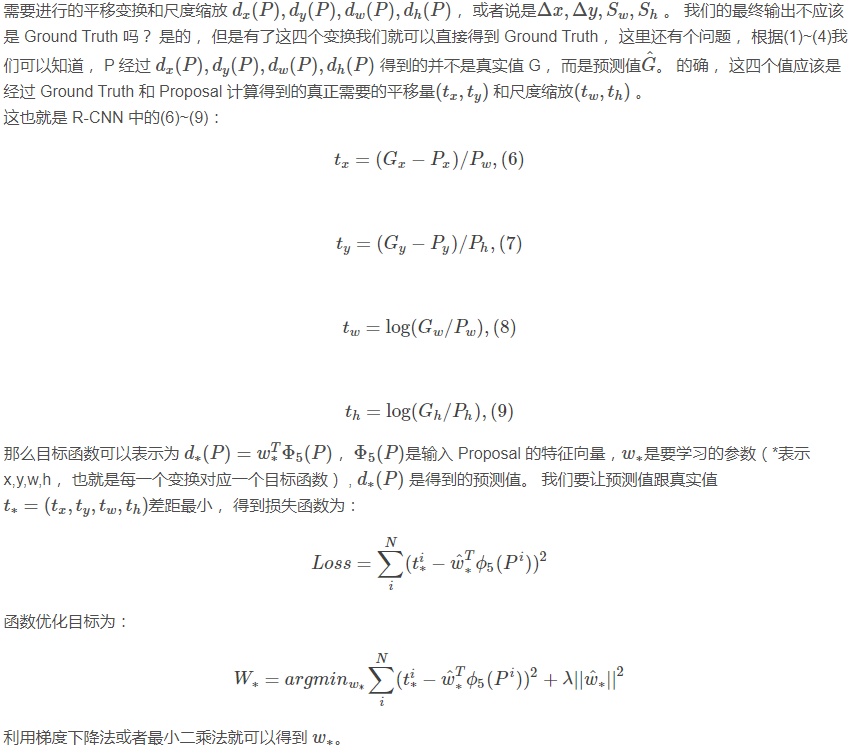
1. **边框回归怎么做的？**



**Input**:

RegionProposal→P=(Px,Py,Pw,Ph)，这个是什么？ 输入就是这四个数值吗？其实真正的输入是这个窗口对应的 CNN 特征，也就是 R-CNN 中的 Pool5 feature（特征向量）。 (注：训练阶段输入还包括 Ground Truth， 也就是下边提到的:

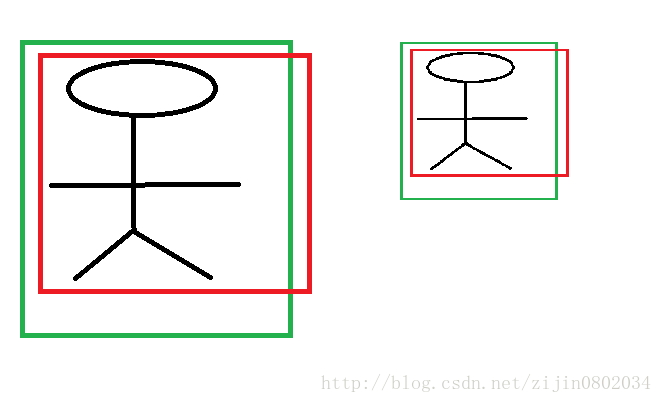
**Output**:



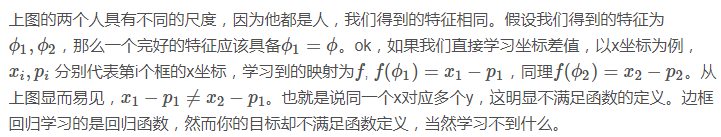
1. **为什么宽高尺度会设计这种形式？**

这边我重点解释一下为什么设计的tx,ty为什么除以宽高，为什么tw,th会有log形式！！！

首先CNN具有尺度不变性， 以图3为例：



**x,y 坐标除以宽高**



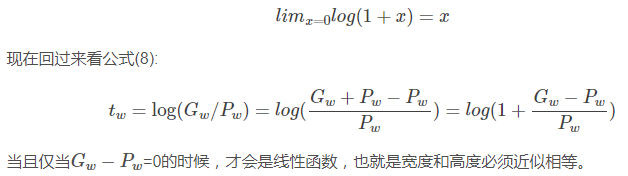
**宽高坐标Log形式**

我们想要得到一个放缩的尺度，也就是说这里限制尺度必须大于0。我们学习的tw,th怎么保证满足大于0呢？直观的想法就是EXP函数，如公式(3), (4)所示，那么反过来推导就是Log函数的来源了。

1. **为什么IoU较大，认为是线性变换？**

当输入的 Proposal 与 Ground Truth 相差较小时(RCNN 设置的是 IoU>0.6)， 可以认为这种变换是一种线性变换， 那么我们就可以用线性回归来建模对窗口进行微调， 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）。这里我来解释：

Log函数明显不满足线性函数，但是为什么当Proposal 和Ground Truth相差较小的时候，就可以认为是一种线性变换呢？大家还记得这个公式不？参看高数1。



对于IoU大于指定值这块，我并不认同作者的说法。我个人理解，只保证Region Proposal和Ground Truth的宽高相差不多就能满足回归条件。x,y位置到没有太多限制，这点我们从YOLOv2可以看出，原始的边框回归其实x，y的位置相对来说对很大的。这也是YOLOv2的改进地方。详情请参考我的博客YOLOv2。