

R-CNN 系列文章(Fast/Faster R-CNN)都训练了 Bounding-box 回归器来对窗口进行校正,以提高最终的检测精度。那么这样做的好处是什么? 具体的又该怎样去做呢? 本文对窗口回归算法进行探讨。

### 1. 问题理解(为什么要做 Bounding-box regression?)

如图 1 所示,绿色的框为飞机的 Ground Truth,红色的框是 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。

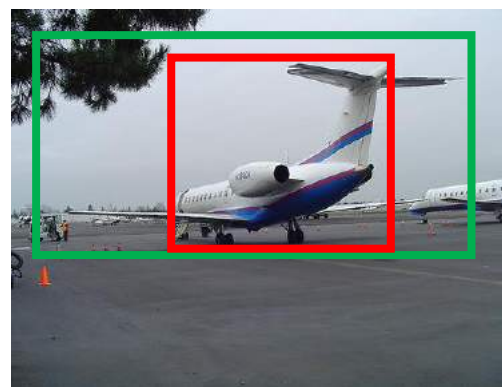


图 1

### 2. 问题数学表达(回归/微调的对象是什么?)

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  $\hat{G}$ 。

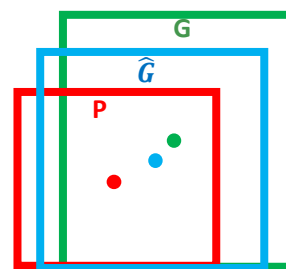


图 2

即: 给定  $(P_x, P_y, P_w, P_h)$ , 寻找一种映射  $f$ , 使得  $f(P_x, P_y, P_w, P_h) = (\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h)$ , 且  $(\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h) \approx (G_x, G_y, G_w, G_h)$

### 3. 问题解决方案(Bounding-box regression)

那么经过何种变换才能从图 2 中的窗口 P 变为窗口  $\hat{G}$  呢？比较简单的思路就是：

(1) 先做平移 ( $\Delta x, \Delta y$ ),  $\Delta x = P_w d_x(P), \Delta y = P_h d_y(P)$ 。

这实际上就是 R-CNN 论文中的

$$\hat{G}_x = P_w d_x(P) + P_x \quad (1)$$

$$\hat{G}_y = P_h d_y(P) + P_y \quad (2)$$

(2) 然后再做尺度缩放 ( $S_w, S_h$ ),  $S_w = P_w d_w(P), S_h = P_h d_h(P)$ , 对应论文中

$$\hat{G}_w = P_w \exp(d_w(P)) \quad (3)$$

$$\hat{G}_h = P_h \exp(d_h(P)) \quad (4)$$

观察(1)~(4)我们发现，我们需要学习的是  $d_x(P), d_y(P), d_w(P), d_h(P)$  这四个变换。

下一步就是设计算法得到这四个映射。当输入的 Proposal 与 Ground Truth 相差较小时(R-CNN 设置的是  $\text{IoU} > 0.6$ )，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。

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。

线性回归就是给定输入的特征向量 X，学习一组参数 W，使得经过线性回归后的值跟真实值 Y(Ground Truth)非常接近。即  $Y \approx WX$ 。那么 Bounding-box 中我们的输入以及输出分别是什么呢？

**输入：**Region Proposal  $\rightarrow P = (P_x, P_y, P_w, P_h)$ ，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  $t_* = (t_x, t_y, t_w, t_h)$ ）

**输出：**需要进行的平移变换和尺度缩放  $d_x(P), d_y(P), d_w(P), d_h(P)$ ，或者说是  $\Delta x, \Delta y, S_w, S_h$ 。我们的最终输出不应该是 Ground Truth 吗？是的，但是有了这四个变换

我们就可以直接得到 Ground Truth，这里还有个问题，根据(1)~(4)我们可以知道，P 经过  $d_x(P), d_y(P), d_w(P), d_h(P)$  得到的并不是真实值  $G$ ，而是预测值  $\hat{G}$ 。的确，这四个值应该是经过 Ground Truth 和 Proposal 计算得到的真正需要的平移量  $(t_x, t_y)$  和尺度缩放  $(t_w, t_h)$ 。

这也就是 R-CNN 中的(6)~(9)：

$$t_x = (G_x - P_x) / P_w \quad (6)$$

$$t_y = (G_y - P_y) / P_h \quad (7)$$

$$t_w = \log(G_w / P_w) \quad (8)$$

$$t_h = \log(G_h / P_h) \quad (9)$$

那么目标函数可以表示为  $d_*(P) = w_*^T \Phi_5(P)$ ， $\Phi_5(P)$  是输入 Proposal 的特征向量， $w_*$  是要学习的参数 (\*表示 x,y,w,h，也就是每一个变换对应一个目标函数)， $d_*(P)$  是得到的预测值。我们要让预测值跟真实值  $t_* = (t_x, t_y, t_w, t_h)$  差距最小，得到损失函数为：

$$\text{Loss} = \sum_i^N (t_*^i - \hat{w}_*^T \phi_5(P^i))^2$$

函数优化目标为：

$$\hat{w}_* = \underset{\hat{w}_*}{\operatorname{argmin}} \sum_i^N (t_*^i - \hat{w}_*^T \phi_5(P^i))^2 + \lambda \|\hat{w}_*\|^2. \quad (5)$$

利用梯度下降法或者最小二乘法就可以得到  $w_*$ 。

#### 4.测试阶段

根据3 我们学习到回归参数  $w_*$ ，对于测试图像，我们首先经过 CNN 提取特征  $\Phi_5(P)$ ，预测的变化就是  $d_*(P) = w_*^T \Phi_5(P)$ ，最后根据(1)~(4)对窗口进行回归。