

目标检测相关细节

叶亮

2020 年 10 月 10 日

1 Loss

1.1 IoU loss

1.2 GIoU loss

1.3 DIoU loss

Code optimization, the calculation of distance between two center points of bboxes.

$$\begin{aligned}(C_1 - C_2)^2 &= ((x1_{C1} + x2_{C1})/2 - (x1_{C2} + x2_{C2})/2)^2 \\ &= ((x1_{C1} + x2_{C1}) - (x1_{C2} + x2_{C2}))^2/4\end{aligned}\tag{1}$$

1.4 CIoU loss

1.5 BIoU loss

2 bbox & anchor

2.1 coder

在 RetinaNet、SSD、Cascade R-CNN 等网络中，网络预测的 bbox 都会进行 Delta xywh 编码，即将原始的 (xmin, ymin, xmax, ymax) 进行编码，计算相对距离，来减小回归的过拟合，提升回归的稳定性。

$$\begin{aligned}\delta_x &= (g_x - p_x)/p_w & \delta_y &= (g_y - p_y)/p_h \\ \delta_w &= \log g_w/p_w & \delta_h &= \log g_h/p_h\end{aligned}\tag{2}$$

上述公式计算得到的 δ 值通常很小，因为网络通常只对 p 进行少量微调，导致回归 loss 比分类 loss 小很多。为了提升学习的有效性， δ 通常需要经过均值和方差进行标准化。

$$\delta'_x = \frac{\delta_x - \mu_x}{\rho_x} \quad (3)$$

3 training

3.1 Optimizer

3.1.1 warmup

warmup 通常有三个方式:linear, constant, exp. 通常需要设置 warmup 的迭代数 $iter_{total}$ 和 warmup 的增加比率 ratio,

$$lr_t = lr_{constant} * ratio \quad (4)$$

$$\begin{aligned} lr_t &= lr_{const} * (1 - k), \\ k &= (1 - iter_t/iter_{total}) * (1 - ratio) \end{aligned} \quad (5)$$

$$lr_t = lr_{const} * k, \quad (6)$$

$$k = ratio^{1 - iter_t/iter_{total}} \quad (7)$$

4 YOLOv4

YOLOv4 的实现过程中,anchor 的生成以及 label 与 anchor 的对应关系的构建方法。

4.1 Anchor generation & target build

与 retinanet, rcnn 系列等 bbox 预测不同的是, yolo 模型推理得到的结果 (x,y,w,h) 需要经过如下公式进行编码:

$$b_x = \rho(t_x) + c_x \quad (8)$$

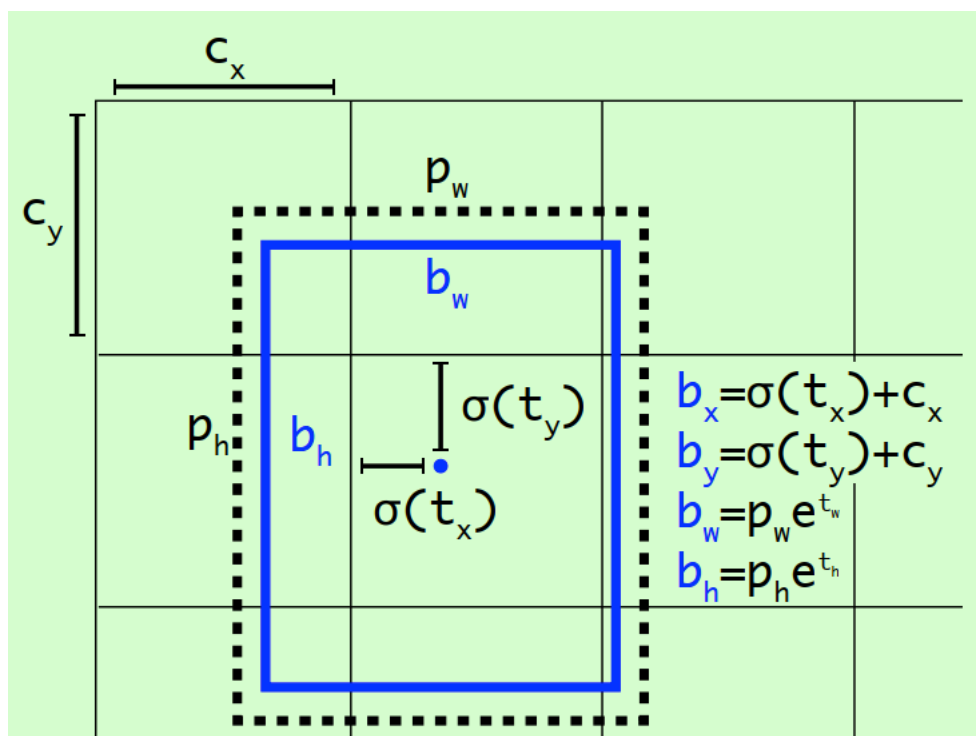


图 1: Yolo bbox prediction

5 backbone

5.1 Regularization

5.1.1 Dropout

原理,dropout 随机丢弃神经元 (全连接中输入神经元), 实现方式为 $keep_prob$, 每个神经元生成一个随机数 $k, k < keep_prob$ 即丢弃。优点, 该方法有利于分类中泛化能力的提升.

5.1.2 Drop Connect

5.1.3 Drop block

6 Refinedet

6.1 Anchor

Refine 中的 anchor 计算。对于每一个 feature map, 首先计算其 mesh grid, 然后计算每个框的中心点 $(x, y) = ((i+0.5)/featsize, (j+0.5)/featsize)$, 然后根据每个 feature map 对应的 anchor box 的大小, 计算 anchor 的长和宽 $WH_{ki} = box_k/imagesize$