layout: post
title: "yolo系列文章之yolov2详解"
date: 2020-07-08
description: "目标检测"

tag: 目标检测

论文地址：YOLO9000: Better, Faster, Stronger

主要介绍

YOLO9000可以检测多达9000种不同的物体，通过使用多尺度的训练方法，可以使得yolov2在多种尺度上运行，在检测速度与精度上可以达到平衡。另外利用联合训练目标检测与分类的方法，使用这种方法，我们在COCO检测数据集上与ImageNet分类数据集上同时训练YOLO9000， 这种联合训练可以使得YOLO9000能够检测未标注检测数据的目标检测数据集。在ImageNet检测数据集上验证这种方法，尽管在200类中仅仅有44类含有标注信息，依然得到19.7的map，在COCO数据集中没有的156类，我们得到了16.0map。

由于检测数据集获取标注难度较大，成本较高，通过利用已有的大量分类数据来扩大当前检测系统的范围。 提出的方法使用目标分类的分层视图，因此可以将不同的数据集组合在一起。

提出一种联合训练的方法，同时利用目标检测数据集与分类数据集一起进行训练，利用检测数据标注信息来学习精确的定位目标，利用分类图像来增加模型鲁棒性。

算法细节介绍

YOLOv1存在大量的问题，尤其是与two-stage的方法相比，产生大量的定位错误，而且召回率相对较低，因此YOLO9000主要着眼于改善召回率与定位精度。

效果更好

1. 使用**batch normalization**，改善模型收敛性，不再需要其他形式的正则化
2. 采用**High Resolution Classifier**（高分辨率分类器），一般在Imagenet图像分类任务中，模型基本上采用224x224的分辨率进行训练。使用更大的输入图像分辨率能够有效的提升算法的精度。
3. **Convolutional With Anchor Boxes**（与锚框卷积），从YOLO中移除全连接层，并使用锚框来预测边界框。 首先我们消除一个池化层，以使网络卷积层的输出具有更高的分辨率，输入图像的尺寸采用416，这样保证在需要 的特征图上有奇数个位置，这样就能保证只有一个中心单元，YOLO下采样层下降32倍得到13x13的特征图。

   引入锚框后，我们将类预测机制与空间位置分开处理，单独预测每个锚框的类及其目标。 遵循原来的YOLO的做法，目标预测依然预测了真实标签框（ground truth box）和候选框的IOU，而类别预测也是预测了当有目标存在时，该类别的条件概率。
4. **Dimension Clusters**（维度聚类），如果在YOLO中使用锚框时，锚框的尺寸是手工设计挑选的，这很影响算法的精度，与收敛效果。这里在训练集的边界框上采用K-means来进行聚类，而距离度量时，如果采用欧式距离，那么较大的框产生的误差较大，这样影响聚类效果，因此据类度量采用`1-IOU`作为距离度量。

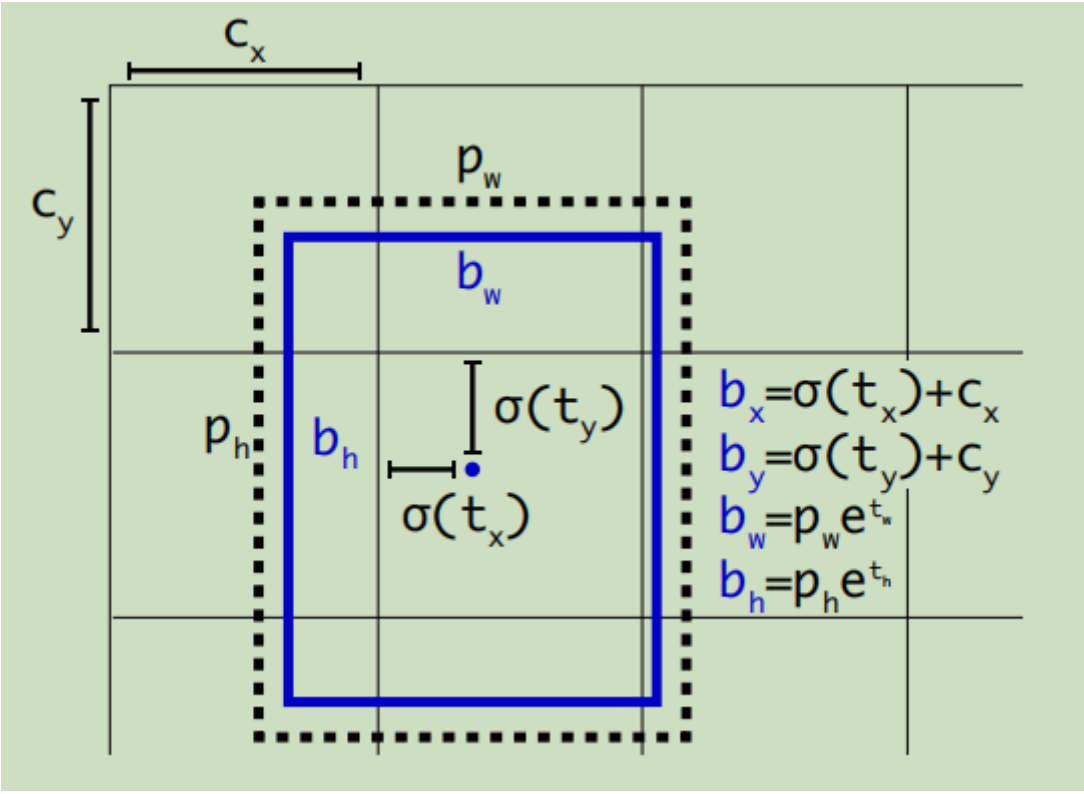
5. **Direct location prediction**（直接位置预测），当在yolo中使用锚框时，会遇到模型不稳定的问题，尤其在早期迭代的过程中，大量的不稳定主要来自于预测bbox的位置，原有的边界框预测公式为：

$$x = (t_x * w_a) + x_a$$
$$y = (t_y * h_a) + y_a$$

其中xa,ya,wa,ha分别为锚框的位置信息，tx，ty是要学习的参数。由于 tx，ty 的取值没有任何约束，因此预测边框的中心可能出现在任何位置，训练早期阶段不容易稳定。因此在YOLOv2中作者更改了这个公式，将预测边界框的中心约束在特定的grid cell中。

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$Pr(object) * IOU(b, object) = \sigma(t_o)$$

其中，b_x, b_y, b_w, b_h 是预测边框的中心和宽高。$Pr(object) * IOU(b, object)$ 是预测边框的置信度，YOLO1是直接预测置信度的值，这里对预测参数 t_o 进行σ变换后作为置信度的值。c_x, c_y 是当前网格左上角到图像左上角的距离，要先将网格大小归一化，即令一个网格的宽=1，高=1。p_w, p_h 是先验框的宽和高。 σ是sigmoid函数。 t_x, t_y, t_w, t_h, t_o 是要学习的参数，分别用于预测边框的中心和宽高，以及置信度。



如图所示，tx，ty被限制在（0，1）之中，模型更容易学习，更容易稳定。

6. **Fine-Grained Features**（细粒度特征），为了解决小目标检测比较困难，准确率较低的问题，yolov2中添加了一个passthrough层，将26x26x512的特征，转换为13x13x2048与最后特征层输

出13x13x1024相叠加为13x13x3072作为最终预测的特征层。

7. **Multi-Scale Training**（多尺度训练），yolov2的模型仅仅使用了卷积层与池化层，因此在训练时可以实时更换输入图像的尺度，运用多尺度及逆行模型的训练。每10个epoch网络会随机选择一个输入尺度进行训练，每次从{320，352..., 608}（均为32的倍数，因为网络经过下采样缩放32倍）。

速度更快

没有采用VGG网络作为backbone，而是采用自定义的DarkNet作为backbone，精度不弱于vgg，但是浮点运算量减少到原来的1/5，速度更快。

YOLO2的训练主要包括三个阶段。

第一阶段就是先在ImageNet分类数据集上预训练Darknet-19，此时模型输入为 224*224 ，共训练160个epochs。

然后第二阶段将网络的输入调整为 448*448 ，继续在ImageNet数据集上finetune分类模型，训练10个epochs，此时分类模型的top-1准确度为76.5%，而top-5准确度为93.3%。

第三个阶段就是修改Darknet-19分类模型为检测模型，移除最后一个卷积层、global avgpooling层以及softmax层，并且新增了三个 3*3*1024卷积层，同时增加了一个passthrough层，最后使用 1*1 卷积层输出预测结果，输出的channels数为：**num_anchors*(5+num_classes)**，和训练采用的数据集有关系。由于anchors数为5，对于VOC数据集（20种分类对象）输出的channels数就是125，最终的预测矩阵T的shape为 (batch_size, 13, 13, 125)，可以先将其reshape为 (batch_size, 13, 13, 5, 25)，其中 T[:, :, :, :, 0:4] 为边界框的位置和大小 (t_x, t_y, t_w, t_h)，T[:, :, :, :, 4] 为边界框的置信度，而 T[:, :, :, :, 5:] 为类别预测值。

更强

提出了一个联合训练分类和检测数据的机制。 用检测的图像来学习检测算法的特定信息，如边界框坐标预测和目标以及如何对常见目标进行分类。通过使用仅具有类标签的图像来扩展其可检测类别的数量。

在训练的过程中，混合图像分类与检测数据集，如果是检测数据样本，那么训练时其loss包括分类误差与检测定位误差，如果是分类样本，那么在训练的过程中，loss只包含分类误差。

但是这种方法也有很多的条件，检测数据集只有常用的目标和通用的标签，分类数据集含有更多的标签类别信息。如果想要在这两种数据集上训练，需要整合这两种数据集并且混合label信息。

大多数分类方法使用涵盖所有可能类别的softmax层来计算最终概率分布。使用softmax，意味着类是相互排斥的。这给组合数据集带来了问题，例如，你不能用这个模型来组合ImageNet和COCO，因为类Norfolk terrier和dog不是互斥的。

相反，我们可以使用多标签模型来组合不会互相排斥的数据集。这个方法忽略了我们所知道的关于数据的所有结构，例如所有的COCO类都是相互独立的。

通过构建word tree来进行标签的处理。

参考文章：https://zhuanlan.zhihu.com/p/47575929