YOLO v1 的原理及实现过程

本文链接：https://blog.csdn.net/weixin_42278173/article/details/81778217

目标检测是一件比较实际的且具有挑战性的计算机视觉任务，其可以看成图像分类与定位的结合，给定一张图片，目标检测系统要能够识别出图片的目标并给出其位置，由于图片中目标数是不定的，且要给出目标的精确位置，目标检测相比分类任务更复杂。目标检测的一个实际应用场景就是无人驾驶，如果能够在无人车上装载一个有效的目标检测系统，那么无人车将和人一样有了眼睛，可以快速地检测出前面的行人与车辆，从而作出实时决策。

本文的内容如下：

- YOLO 与其他检测方法的区别
- 基本概念的介绍
- 网络结构
- LOSS 函数
- 训练与测试

YOLO 与其他检测方法的区别：

YOLO 的全称叫做“You Only Look Once”，简单来说，YOLO 可以做到将一张图片输入，直接输出最终结果，包括框和框内物体的名称及 score(得分)。而很多其他检测方法如 Faster R-CNN 便是通过两次检测来达到目的。具体的两次检测过程如下所述：

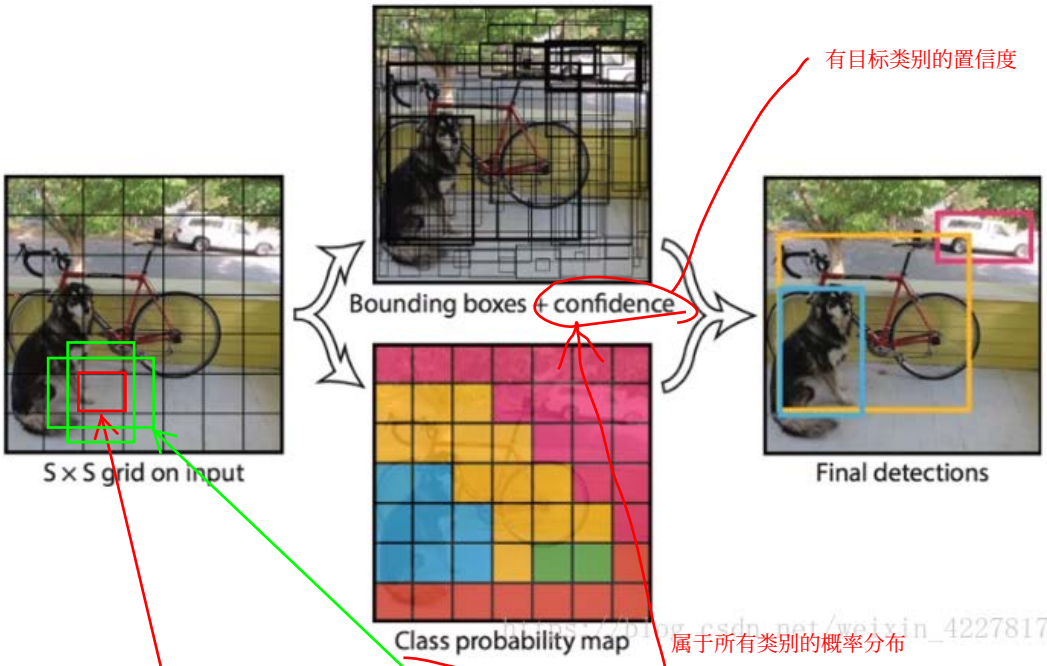
- Input Image 经过 CNN 特征提取，首先来到 Region Proposal Network 输出 classification，第一次并不是判定物体在 COCO 数据集上对应的目标类的某一类，而是输出一个 binary 的值 p,可以理解为 p 属于[0,1]，人工设定一个 threshold=0.5。判定是否有目标在范围内
- RPN 网络的意义是如果一个 Region 的 p 大于等于 threshold，那么就认为这个 Region 中可能是目标类别中的某一类，这些被选出的 region 又叫做 ROI(Region of Interest)。这一步会同时框定出 ROI 感兴趣区域的大致位置，即输出 Bounding Box。
- 然后将 ROI 输入到普通的分类网络，即可得到整个网络最终的输出 classification,这里的 class 真正对应了 COCO 数据集里 80 个类别的具体类别之一。在这一步之后，还要对 bbox 进行尺寸的微调，输出第一张图右上方的 Bounding-box regression。

因此，Region Proposal 的作用为：

1. 避免影响网络的分类性能，如果不进行第一步的筛选，anchor 框定的 region 会有许多类似于蓝天，草地这种背景类别，对于物体分类没有益处。
2. 减少参数，增加网络分类性能。

YOLO 的优势是：不需要 region proposal，所谓的 region proposal 即需要进行多于一次的识别过程。以 Faster R-CNN 为代表的目标检测的方法就是两次检测的过程。

基本概念的介绍



有目标类别的置信度

属于所有类别的概率分布

1. 首先将原图像分成 S x S grid，在介绍 yolo 的论文里 S=7，即将图像分为 7 x 7 grid。

2. 每个 grid cell 都会生成 B 个 Bounding Box 去进行物体的框定和分类，B 在论文里给的是 2，因此在此例中有 98 个 Bounding Box。

这两个bbox的四个坐标没有固定大小，是在7*7*30的网络最后输出层直接作为预测输出，然后iou成一个作为最终输出，根据与标签的差别不断调整优化

3. Confidence 包含两方面，边界框**含有目标的可能性**以及**边界框的准确度**。也就是location，也可反映为交并比

4. 利用 **NMS（非极大值抑制）算法**得到最后的 final 的 detections。

5. 对于每一个单元格要给出 C 个类别概率值，其表征的是由该单元格负责预测的边界框其目标属于各个类别的概率。这些概率值其实是在各个边界框置信度下的条件概率，即：

$$Pr(class_i|object) * Pr(object) * \text{IOU}_{pred}^{truth} = Pr(class_i) * \text{IOU}_{pred}^{truth}$$

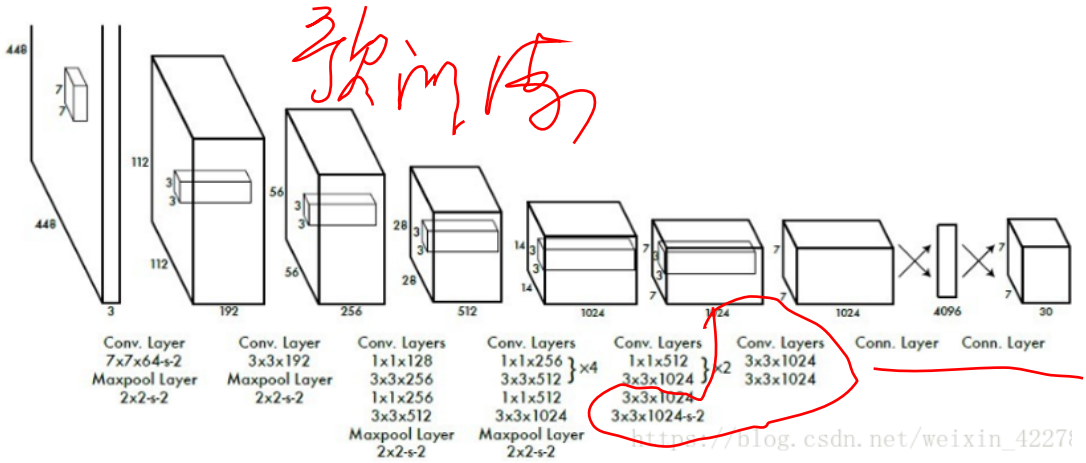
其中 $Pr(object) * IOU_{pred}^{truth}$ 为**上述 3.**中的 confidence，而 IOU_{pred}^{truth} 是指的预测框和实际框（Ground Truth）的交并比。

这边我们提到了边界框，边界框的大小和位置用 4 个值来表示：(x,y,w,h)，其中(x, y)是边界框的中心坐标，w 和 h 是边界框的宽与高。具体来说，(x, y)是相对于每个单元格左上角坐标点偏移值，并且单位是相对于单元格大小的，而 w 和 h 预测值是相对于整个图片的宽和高的比例，这样理论上 4 个元素的大小在[0, 1]范围内，而且每个边界框的预测值实际上包含 5 个元素：(x, y, w, h, c)。前四个元素表征边界框的大小和位置，最后一个值是置信度。

总结一下：每个单元格需要预测(B*5+C)个值。输入的图片为 S x S 网络，因此最终的预测值为 S x S x (B x 5 + C)大小的张量，在论文里给出的例子为 S=7，B=2， 因此最终的张量大小为 7 x 7 x 30。

网络结构

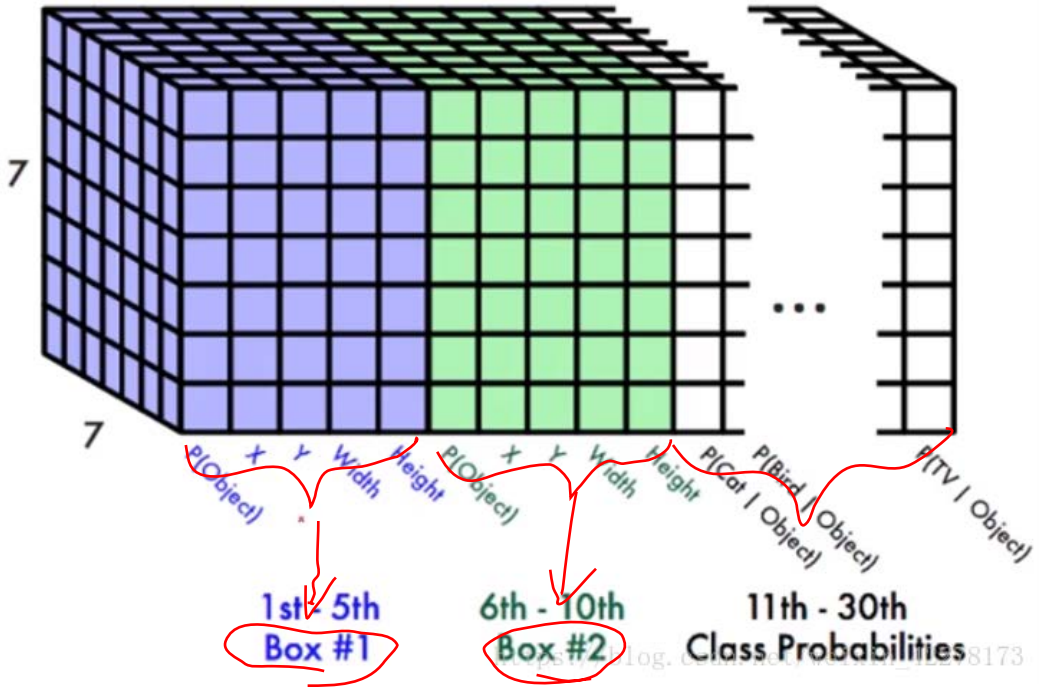
Yolo 采用卷积网络来提取特征，然后使用全连接层来得到预测值。网络结构参考 GooLeNet 模型（2014 年 Christian Szegedy 提出的一种全新的深度学习结构）。YOLO 检测层包含 24 个卷积层和 2 个全连接层。

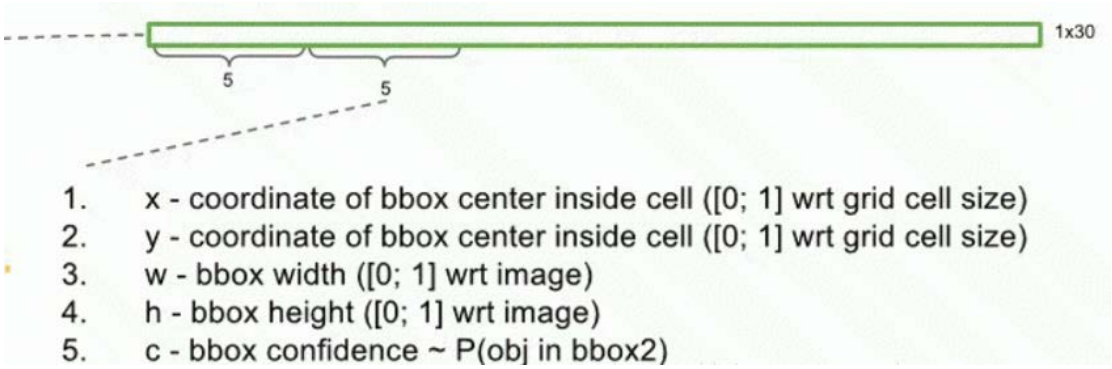


（这里一开始我数不出来为什么有 24 个卷积层，其实要仔细看图，你会发现有一些“x4”“x2”，把这些考虑进去你就会发现确实是 24 层）

最后得到的 7 x 7 x 30 代表的是最后的输出，代表一共 49 个 cell，每个 cell 拥有 30 个值，其中有 20 个值为类别概率值，即该 cell 检测出来的属于某类物体的概率。而剩下的 10 个值可以分成两部分，分表代表 cell 两个 Bounding Box 各自的参数部分。

我们取一个 cell 来看，一个 cell 有 30 个元素，如图所示：





$$\text{confidence} = Pr(object) * IOU_{pred}^{truth}$$

如果在 grid cell 里没有物体存在，则 Pr(object)=0，存在的意思是指物体的 ground truth 中心点在这个 cell 里面。另外我们发现，一个 grid cell 里面虽然有两个 Bounding Box，但是它们共享同一组分类概率，因此同一个 cell 只能识别同一个物体。

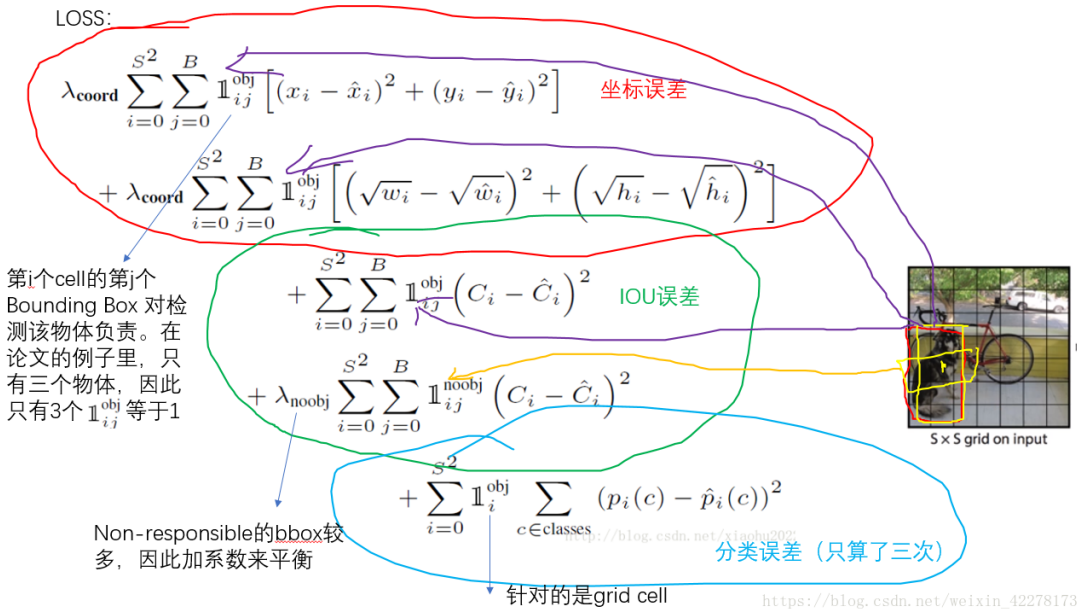
边界框的大小和位置用 4 个值来表示：(x, y, w, h)，其中(x , y)是边界框的中心坐标，w 和 h 是边界框的宽与高。(x, y)是相对于每个单元格左上角坐标点偏移值，并且单位是相对于单元格大小的，而 w 和 h 预测值是相对于整个图片的宽和高的比例，这样理论上 4 个元素的大小在[0, 1]范围内，而且每个边界框的预测值实际上包含 5 个元素：(x, y, w, h, c)。前四个元素表征边界框的大小和位置，最后一个值是置信度。

LOSS 函数

LOSS 函数和其他神经网络一样，是优化的目标，在 yolo v1 里 LOSS 函数相当复杂，大体可以分为三部分，下面具体介绍一下 LOSS 函数：

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}$$

为了便于说明，我们将各部分用不同颜色圈起来：



1. 红色圈代表的是坐标误差，极好理解。这里要注意，w 和 h 在进行误差计算的时候取的是它们的平方根，原因是对不同大小的 bbox 预测中，相比于大 bbox 预测偏一点，小

box 预测偏一点更不能忍受。而 sum-square error loss 中对同样的偏移 loss 是一样。 为了缓和这个问题，作者用了一个比较取巧的办法，就是将 box 的 width 和 height 取平方根代替原本的 height 和 width。

2. 绿色圈我们命名为 IOU 误差，主要是针对 confidence 的误差，由于 confidence 的计算方式（如上所述），主要受到 IOU 的影响，因此我们叫做 IOU 误差。

3. 蓝色圈称为分类误差，与 bounding box 无关，只和 cell 有关，是针对每个 cell 做概率的差值计算。

4. $\mathbb{1}_{ij}^{obj}$ 符号代表第 i 个 cell 的第 j 个 Bounding Box 对检测该物体负责。在论文的例子里，只有三个物体，因此只有 3 个该项为 1，凡是不负责的均为 $\mathbb{1}_{ij}^{noobj}$

除此之外，针对各项的系数，还有一些 tips：

1. 损失函数的设计目标就是让坐标（x,y,w,h），confidence，classification 这个三个方面达到很好的平衡。

2. 8 维的 localization error 和 20 维的 classification error 同等重要显然是不合理的。更重视 8 维的坐标预测，给这些损失前面赋予更大的 loss weight。

3. 如果一个网格中没有 object（一幅图中这种网格很多），那么就会将这些网格中的 box 的 confidence push 到 0，相比于较少的有 object 的网格，这种做法是会导致网络不稳定甚至发散。

4. 对没有 object 的 bbox 的 confidence loss，赋予小的 loss weight。

训练的最终结果是得到一个网络，网络参数是根据不停地优化 LOSS 函数而得到的。

训练与测试：

训练：训练集的图片都进行各个物体的人工标注（4 个顶点的坐标），通过运算，得到 x y w h，然后图片被分成 S x S 个小格，针对每个小格有 b 个 BBOX，然后经过优化 LOSS，不停地使得 BBOX 逼近 Ground Truth。 BBOX难道在不停调整变大？

测试：利用训练好的网络，输入图片，直接给出 box 和对应的物体。

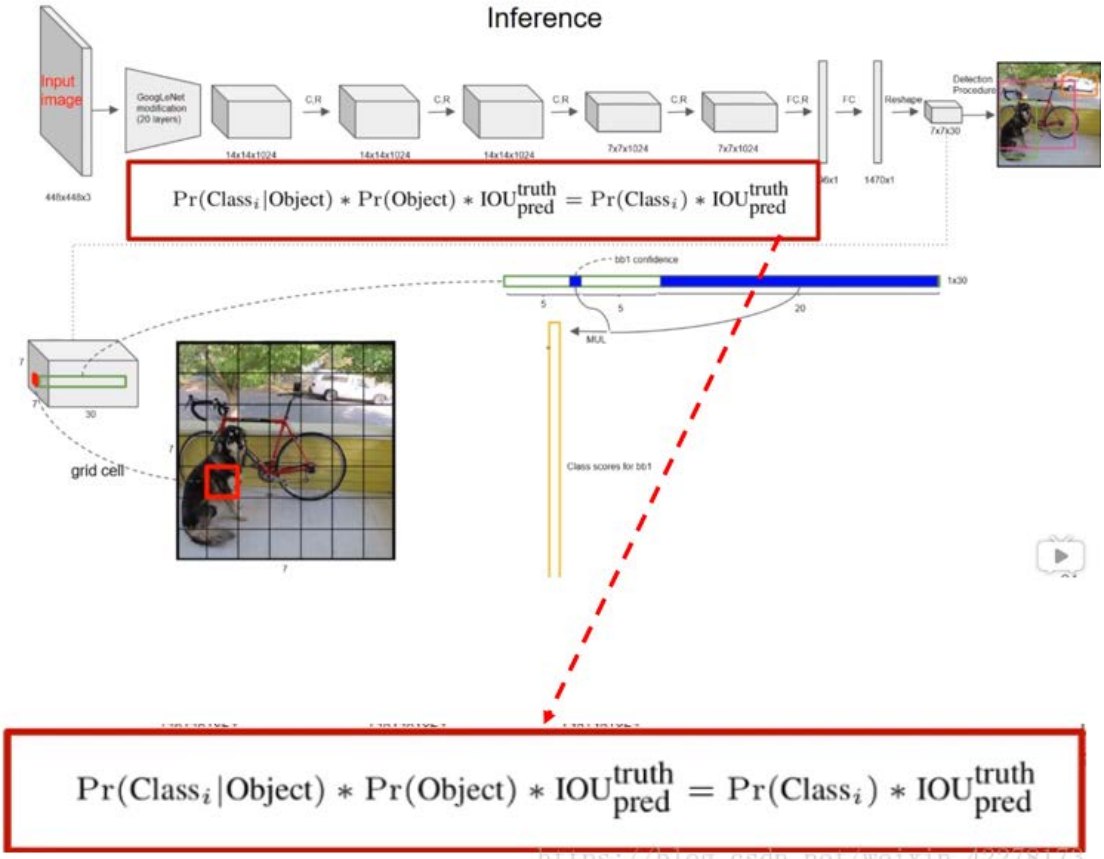
训练过程：

1. 预训练分类网络 :在 ImageNet 1000-class competition dataset 上预训练一个分类网络，这个网络是 Figure3 中的前 20 个卷积网络+average-pooling layer+ fully connected layer（此时网络输入是 224*224）。

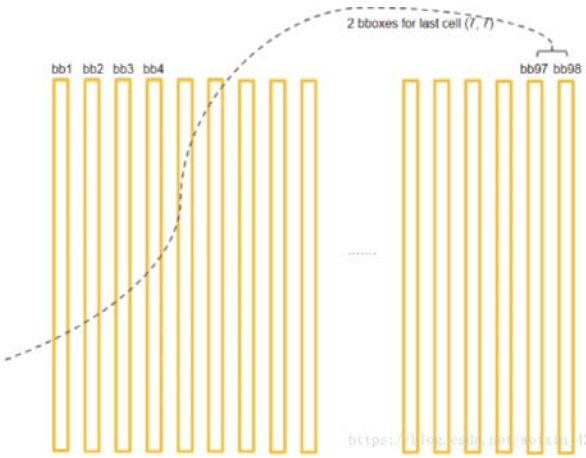
2. 训练检测网络：预训练之后，在预训练得到的 20 层卷积层之上加上随机初始化的 4 个卷积层和 2 个全连接层。由于检测任务一般需要更高清的图片，所以将网络的输入从 224x224 增加到了 448x448。 448/7=64

测试过程：

输入图片 ,网络会按照与训练时相同的分割方式将测试图片分割成 S x S 的形状 ,因此，划分出来的每个网格预测的 class 信息和 Bounding box 预测的 confidence 信息相乘，就得到了每个 Bounding box 的 class-specific confidence score，即得到了每个 Bounding box 预测具体物体的概率和位置重叠的概率。

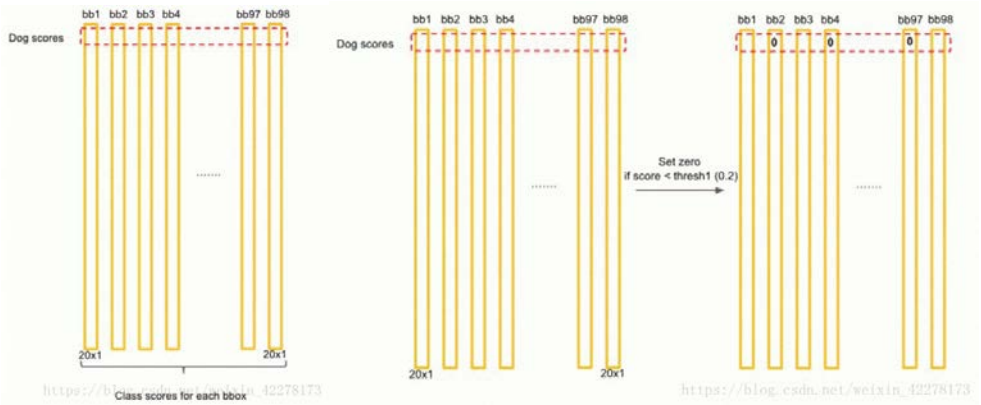


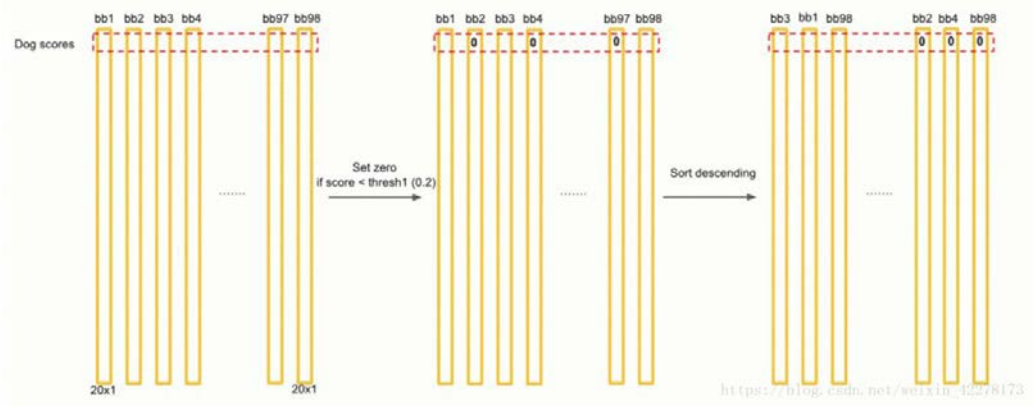
对于 98 个 Bounding Box 都这么运算，最后可以得到：

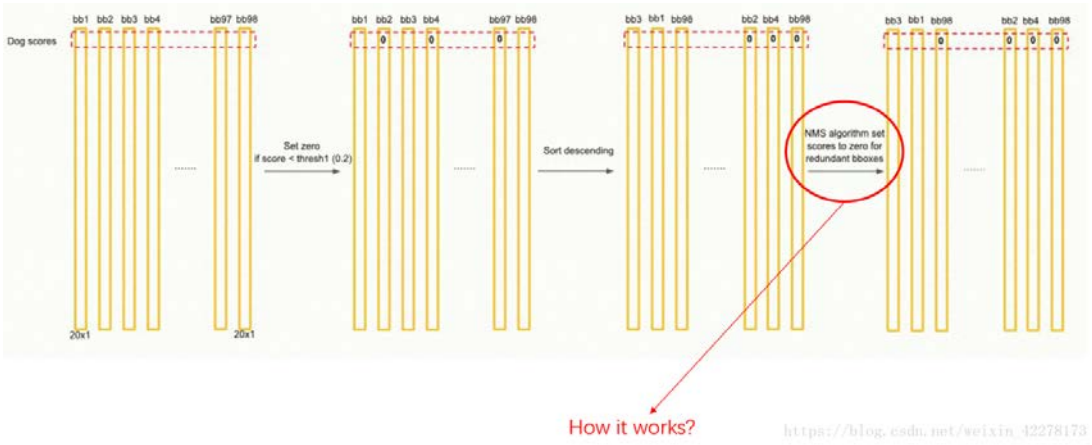


每个“条”一共有 20 个值，分别是 20 个物体的得分，因此一共有 98*20 个值，我们按照类别把它们分为 20 类。之后的过程如下（先以第一类假设为“dog”举例）：

每个BBOX共20位，一个cell是30位







直观来感受一下非极大值抑制的过程：

非极大值抑制：抑制的过程是一个迭代-遍历-消除的过程。

1.将所有框的得分排序，选中最高分及其对应的框。

2.遍历其余的框，如果和当前最高分框的重叠面积(IOU)大于一定阈值，我们就将框删除。

3.从未处理的框中继续选一个得分最高的，重复上述过程。（具体可见其他博主整理，或者 bilibili yolov1 讲解）

针对其他类别，比如 cat，vehicle 等，均采用这种方式，直到将所有的 score 值得到。然后，针对每一个 Bounding Box，查看其最大的值，如果在 20 个值里面最大值非 0，那么我们就把这个 bounding box 认定为是识别对应物体的框。

当然，yolo v1 有相当的缺陷：

1. YOLO 对相互靠的很近的物体，还有很小的群体 检测效果不好，这是因为一个网格中只预测了两个框，并且只属于一类。

2. 对测试图像中，同一类物体出现的新的不常见的长宽比和其他情况是。泛化能力偏弱。