### 目标检测Yolo：

与FasterRcnn不同之处：

1. FasterRcnn将目标检测分解为分类问题与回归问题：首先采用独立的rpn网络求取region proposal，然后利用bounding box regression对提取的位置进行修正，最后采用softmax进行分类。
2. YOLO将物体预测作为一个回归问题进行求解，将图像输入一次网络，便能够得到图像中所有物体的位置和其所属类别对应的置信概率

#### Yolov1

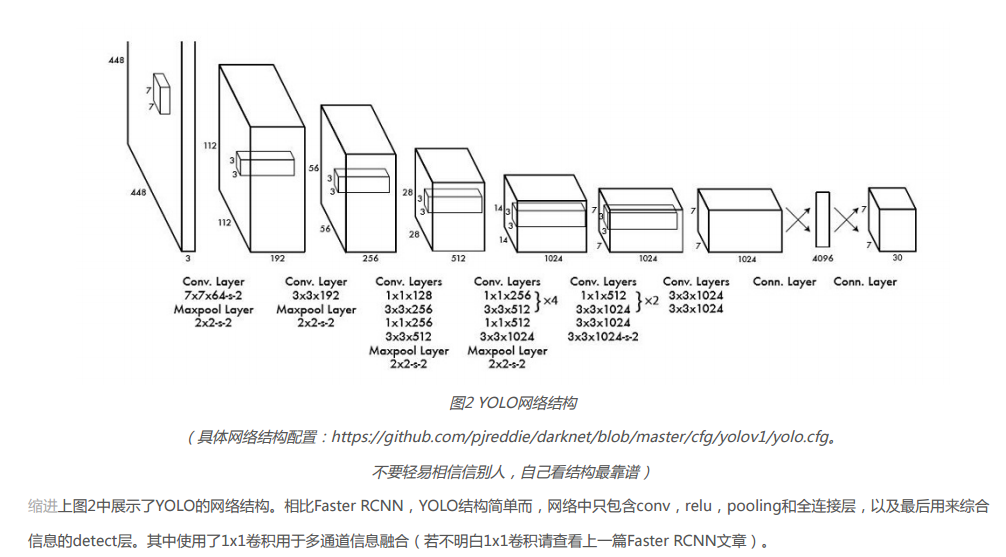
**优点：**

yolo将检测问题整合为回归问题，是网络简单，训练速度加快

**缺点：**

1.输入图像尺寸固定，因为最后一层是全联接层

2.占比较小的目标检测效果不好.虽然每个格子可以预测B个bounding box，但是最终只选择只选择IOU最高的bounding box作为物体检测输出，即每个格子最多只预测出一个物体。当物体占画面比例较小，如图像中包含畜群或鸟群时，每个格子包含多个物体，但却只能检测出其中一个。



**思想：**

1. 将原图划分为SxS的网格，如果一个目标的中心落入某个格子，这个格子就负责检测该目标

2.每个网格要预测B个bounding boxes以及c个类别的概率，在yolo中每个格子只有一个c类别，即相当于忽略了B个bounding boxes，每个格子只判断一次类别

3.每个bounding box除了要回归自身的位置之外，还要预测一个confidence值，这个值代表了所预测的box中所含有目标的置信度和这个bounding box预测有多准



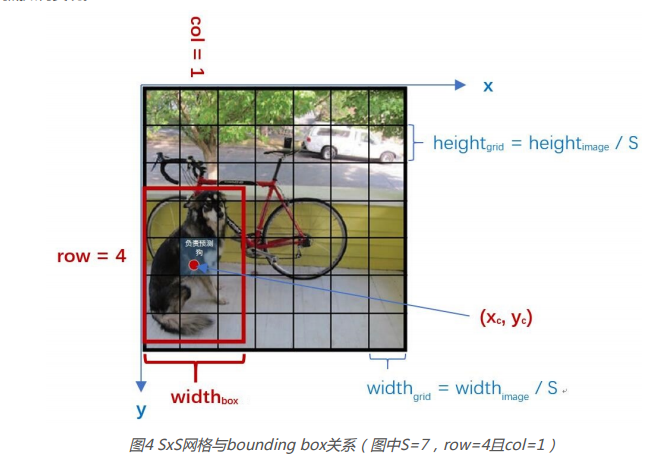
有目标落在中心格子里Pr(obiect)=1,否则Pr（object）=0,第二个是预测的框与GT之间的IOU。所以每个bounding box都包含了五个预测量：（x,y,w,h,confidence）,其中（x,y）代表预测box相对于格子的中心，（w,h）为预测的宽度相对于图片width和height的比例。

4.每个网格预测的类别条件概率和bounding box预测的confidence信息相乘，就得到，每个bounding box的class-specific-confidence score:



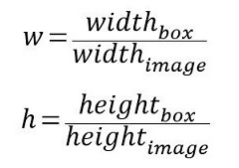
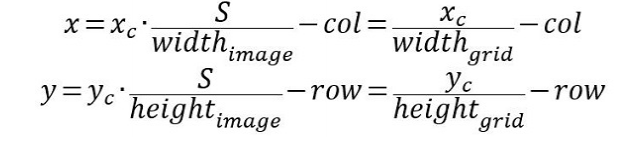
最后设置一个阈值，去掉小于阈值的值，然后对留下的进行非极大值抑制，最终得到检测框。

**YOLO中的bounding box normalization:**

对bound box的坐标（x,y,w,h）进行了normalization,以便进行回归。

中心落到了（row,clo）中，这个网格复制检测狗这个目标，那么归一

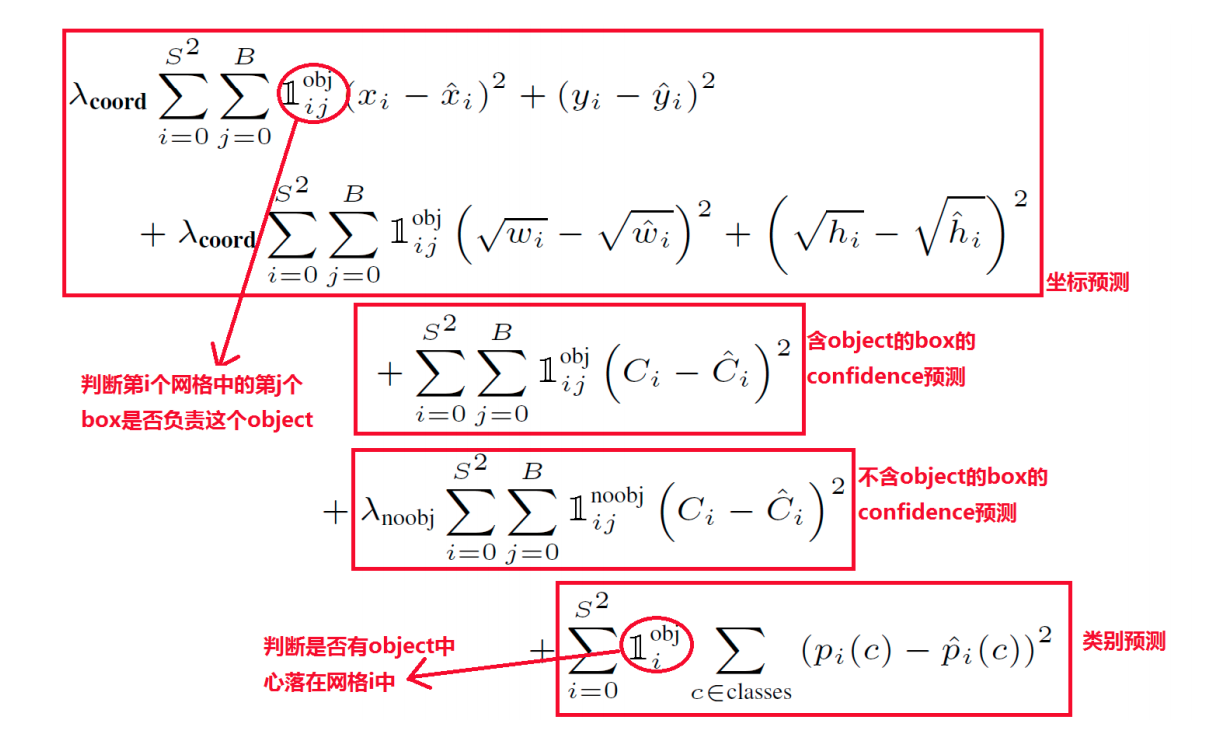
化后的坐标为：



**YOLO的损失函数：**

1. 坐标预测误差：大的框预测稍差一点还能接受，但是小的框预测差一点就不能接受了，所以这里使用开根号进行回归

2.confidence预测误差：由于大多数网格中不含目标，导致大多数confidence=0，所以同等对待函目标的box与不含目标的box是不对的，所以在不含目标的confidence预测误差中乘以权重0.5，还有同等对待4个坐标与一个confidence误差也不合理，所以作者在坐标预测误差前乘以权重5



#### YOLO V2：

对YOLO的改进：  
**1.Batch Normalization: Batch Normalization**

可以提升模型收敛速度，而且可以起到一定正则化效果，降低模型的过拟合。在YOLOv2中，每个卷积层后面都添加了Batch Normalization层，并且不再使用droput。使用Batch Normalization后，YOLOv2的mAP提升了2.4%。

**2.High Resolution Classifier:**

目前大部分的检测模型都会在先在ImageNet分类数据集上预训练模型的主体部分（CNN特征提取器），由于史原因，ImageNet分类模型基本采用大小为224\*224的图片作为输入，分辨率相对较低，不利于检测模型。所以YOLOv1在采用224\*224分类模型预训练后，将分辨率增加至448\*448，并使用这个高分辨率在检测数据集上finetune。但是直接切换分辨率，检测模型可能难以快速适应高分辨率。所以YOLOv2增加了在ImageNet数据集上使用448\*448输入来finetune分类网络这一中间过程（10 epochs），这可以使得模型在检测数据集上finetune之前已经适用高分辨率输入。使用高分辨率分类器后，YOLOv2的mAP提升了约4%。

1. **Convolutional With Anchor Boxes**

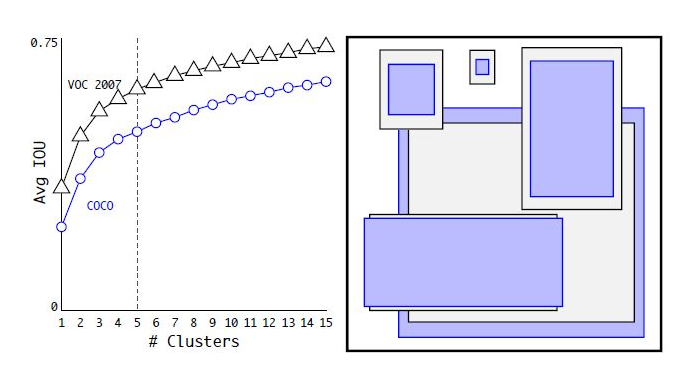
在YOLOv1中，输入图片最终被划分为7\*7网格，每个单元格预测2个边界框。YOLOv1最后采用的是全连接层直接对边界框进行预测，其中边界框的宽与高是相对整张图片大小的，而由于各个图片中存在不同尺度和长宽比（scales and ratios）的物体，YOLOv1在训练过程中学习适应不同物体的形状是比较困难的，这也导致YOLOv1在精确定位方面表现较差。

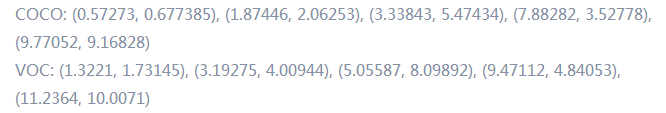
1. **Anchors box**

YOLOv2借鉴了Faster R-CNN中RPN网络的先验框（anchor boxes，prior boxes，SSD也采用了先验框）策略。RPN对CNN特征提取器得到的特征图（feature map）进行卷积来预测每个位置的边界框以及置信度（是否含有物体），并且各个位置设置不同尺度和比例的先验框，所以RPN预测的是边界框相对于先验框的offsets值,采用先验框使得模型更容易学习。所以YOLOv2移除了YOLOv1中的全连接层而采用了卷积和anchor boxes来预测边界框。为了使检测所用的特征图分辨率更高，移除其中的一个pool层。在检测模型中，YOLOv2不是采用448\*448图片作为输入，而是采用416\*416大小。因为YOLOv2模型下采样的总步长为32，对于416\*416大小的图片，最终得到的特征图大小为13\*13，维度是奇数，这样特征图恰好只有一个中心位置。对于一些大物体，它们中心点往往落入图片中心位置，此时使用特征图的一个中心点去预测这些物体的边界框相对容易些。所以在YOLOv2设计中要保证最终的特征图有奇数个位置。对于YOLOv1，每个cell都预测2个boxes，每个boxes包含5个值，前4个值是边界框位置与大小，最后一个值是置信度（confidence scores，包含两部分：含有物体的概率以及预测框与ground truth的IOU）。但是每个cell只预测一套分类概率值（class predictions，其实是置信度下的条件概率值）,供2个boxes共享。YOLOv2使用了anchor boxes之后，每个位置的各个anchor box都单独预测一套分类概率值，这和SSD比较类似（但SSD没有预测置信度，而是把background作为一个类别来处理）。

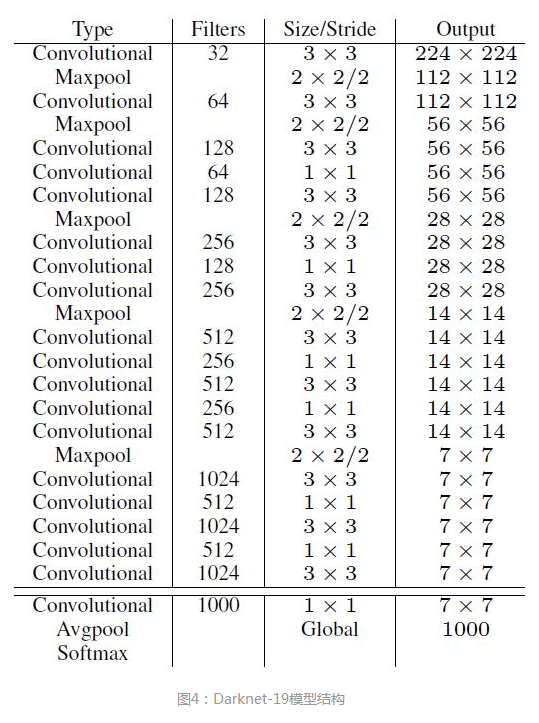
**5.Dimension Clusters**

在Faster R-CNN和SSD中，先验框的维度（长和宽）都是手动设定的，带有一定的主观性。如果选取的先验框维度比较合适，那么模型更容易学习，从而做出更好的预测。因此，YOLOv2采用k-means聚类方法对训练集中的边界框做了聚类分析。因为设置先验框的主要目的是为了使得预测框与ground truth的IOU更好，所以聚类分析时选用box与聚类中心box之间的IOU值作为距离指标。



图为在VOC和COCO数据集上的聚类分析结果，随着聚类中心数目的增加，平均IOU值（各个边界框与聚类中心的IOU的平均值）是增加的，但是综合考虑模型复杂度和召回率，作者最终选取5个聚类中心作为先验框，其相对于图片的大小如右边图所示。对于两个数据集，5个先验框的width和height如下所示

**6. New Network: Darknet-19**

****

**7.Fine-Grained Features：**

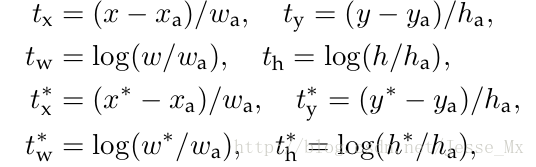
YOLOv2的输入图片大小为416\*416，经过5次maxpooling之后得到13\*13大小的特征图，并以此特征图采用卷积做预测。13\*13大小的特征图对检测大物体是足够了，但是对于小物体还需要更精细的特征图（Fine-Grained Features）。因此SSD使用了多尺度的特征图来分别检测不同大小的物体，前面更精细的特征图可以用来预测小物体。YOLOv2提出了一种passthrough层来利用更精细的特征图。YOLOv2所利用的Fine-Grained Features是26\*26大小的特征图（最后一个maxpooling层的输入），对于Darknet-19模型来说就是大小为26\*26\*512的特征图。passthrough层与ResNet网络的shortcut类似，以前面更高分辨率的特征图为输入，然后将其连接到后面的低分辨率特征图上。前面的特征图维度是后面的特征图的2倍，passthrough层抽取前面层的每个2\*2的局部区域，然后将其转化为channel维度，对于26\*26\*512的特征图，经passthrough层处理之后就变成了13\*13\*2018的新特征图（特征图大小降低4倍，而channles增加4倍，图6为一个实例），这样就可以与后面的13\*13\*1024特征图连接在一起形成13\*13\*3072大小的特征图

**8.Multi-Scale Training**

由于YOLOv2模型中只有卷积层和池化层，所以YOLOv2的输入可以不限于416\*416大小的图片。为了增强模型的鲁棒性，YOLOv2采用了多尺度输入训练策略，具体来说就是在训练过程中每间隔一定的iterations之后改变模型的输入图片大小。由于YOLOv2的下采样总步长为32，输入图片大小选择一系列为32倍数的值：{320,353,…,608}，输入图片最小为320\*320，此时对应的特征图大小为10\*10（不是奇数了，确实有点尴尬），而输入图片最大为608\*608，对应的特征图大小为19\*19。在训练过程，每隔10个iterations随机选择一种输入图片大小，然后只需要修改对最后检测层的处理就可以重新训练。

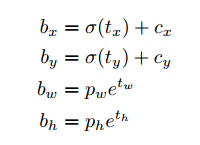
**9.Location prediction**

在采用anchors后发现模型不收敛，所以采用了类似与faster-rcnn中的方法



x 是坐标预测值，xa 是anchor坐标（预设固定值），x∗ 是坐标真实值（标注信息），其他变量 y，w，h 以此类推，t 变量是偏移量

Yolov2对每个bounding box预测四个坐标值(tx, ty, tw, th)，对于预测的cell（一幅图划分成S×S个网格cell），根据图像左上角的偏移(cx, cy)，以及设定的bounding box的宽和高pw, ph可以对bounding box按如下的方式进行预测：

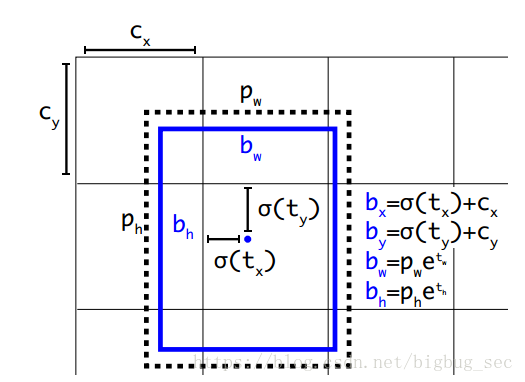


(tx,ty,tw,th)为预测的bounding box 坐标偏移量，tx,ty 经sigmod函数处理过，取值限定在了0~1，实际意义就是使anchor只负责周围的box，有利于提升效率和网络收敛

（cx.cy）是划分的网格根据图像左上角的偏移

(pw,ph）是初始化bounding box的宽高

计算如图所示

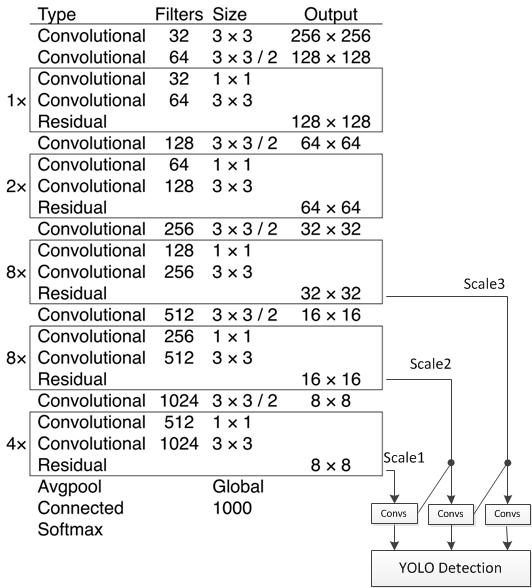


#### YOLOV3:

##### 改进之处：

**1.多尺度预测**

yolov3在三个不同的尺度预测boxes，类似于FPN，网络结构见下图。



**2.更好的基础分类网络（与ResNet相似）和分类器**

·不再使用softmax进行类别预测：Softmax使得每个框分配一个类别（score最大的一个），而对于某些数据集，目标可能有重叠的类别标签，因此Softmax不适用于多标签分类。  
 ·Softmax可被独立的多个logistic分类器替代，且准确率不会下降

yolov3对每个bounding box通过逻辑回归预测一个物体的得分，如果预测的这个bounding box与真实的边框值大部分重合且比其他所有预测的要好，那么这个值就为1.如果overlap没有达到一个阈值（yolov3中这里设定的阈值是0.5），那么这个预测的bounding box将会被忽略，也就是会显示成没有损失值。