# 一、算法和数据结构

排序

# 二、图像处理

# 三、机器学习&深度学习

## 3.1 模型评估

### 错误率

分类错误的样本数占总样本的比例

### 精度

分类正确的样本数占总样本的比例

表：测试样本分类

|  |  |  |
| --- | --- | --- |
| 真实情况 | 预测为正 | 预测为反 |
| 正 | TP（真正） | FN（假反） |
| 反 | FP（假正） | TN（真反） |

### 准确率

分类正确的正样本个数 占 分类器判定为正样本的样本个数的比例。即：预测的结果中，多少是正确的



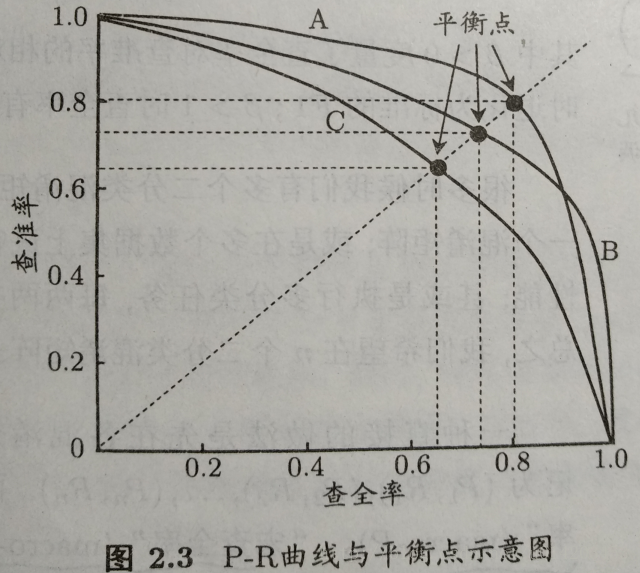
### 召回率

分类正确的正样本个数 占 真正的正样本个数的比值。即：有多少个正确的样本被预测出来 

### P-R曲线

一般来说，我们希望准确率和召回率都是越来越高才好，但是这两个量是一对相互矛盾的量，一个高时，另一个就会偏低。如下图

那如何根据P-R曲线来评价模型的好坏呢？



（P-R曲线上的点代表：在某一阈值下，模型将大于该阈值的结果判定为正样本，小于该阈值的样本定位负样本，此时返回结果对应的召回率和精确率）

最直接的----如果一个模型的P-R曲线被另一个模型的P-R曲线包围，则可以断言，后面的模型要好一些。

但是如果两个曲线有交叉，那就很难说清楚了。

这样，一个合理的判据就是去比较两个曲线下面的面积大小，面积在一定程度上能反映出P和R“双高”的比例，但问题是，这个面积太不容易估算了。因此人们设计了一些综合考虑查全率和查准率的性能度量

### 平衡点

Break-Even Point，简称BEP，平衡点是“查准率=查全率”是的取值，基于此标准，则上图中，模型A优于模型B。

### F1度量

BEP 度量过于简化了，更常用的是F1度量



其中，样本总数=TP+FP+FN+TN

F1度量是基于查准率和查全率的调和平均定义的



在一些应用中，我们对P和R的重视程度又是不同的，因此F1度量的一般形式是：



其中β＞0，度量了查全率对查准率的相对重要性

β=1，退化为标准的F1；

β>1，查全率有更大的影响；

β<1，查准率有更大的影响；

## 3.2 现有流行的网络及其特点？

（1）AlexNet包括5层卷积层和三层全连接层，其中一个万层的

# 关于YOLO

## 一、YOLO背景

### 1.1 YOLO系列算法的基本思想

（1）首先通过特征提取网络对输入图像提取特征，得到一定size的feature map，比如13\*13；

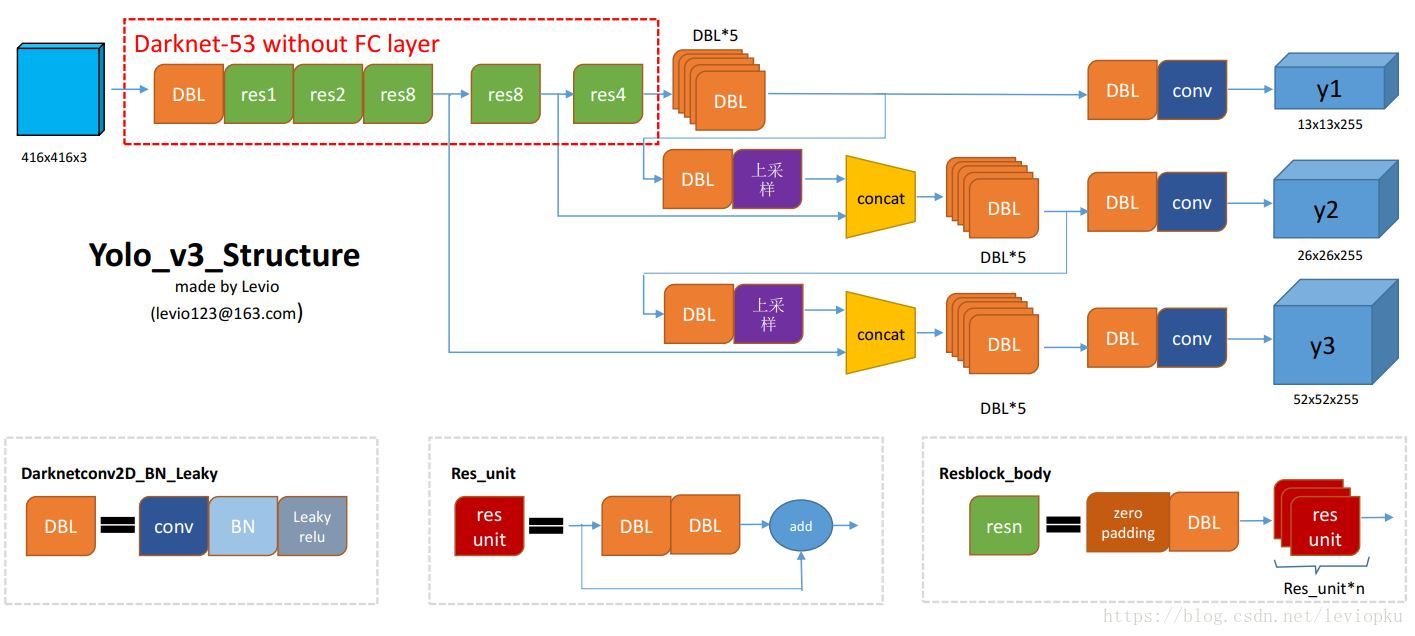
（2）然后将输入图像分成13\*13个grid cell，接着如果ground truth中某个object的中心坐标落在哪个grid cell中，那么就由该grid cell来预测该object；

需要注意的是每个grid cell都会预测固定数量的bounding box（YOLO v1中是2个，YOLO v2中是5个，YOLO v3中是3个，这几个bounding box的初始size是不一样的），那么这几个bounding box中最终是由哪一个来预测该object？答案是：这几个bounding box中只有和ground truth的IOU最大的bounding box才是用来预测该object的。

有上面的分析可得，在最后的feature map上面，比如13\*13，每一个cell的输出为：B**\*（5+C）**，

注：YOLO v1中是（B\*5+C），其中B表示每个grid cell预测的bounding box的数量，比如YOLO v1中是2个，YOLO v2中是5个，YOLO v3中是3个，C表示bounding box的类别数（没有背景类，所以对于VOC数据集是20），5表示4个坐标信息和一个置信度（objectness score/confidence score）

## 三、YOLOv3 网络架构



（1）DBL: 如图左下角所示，也就是代码中的Darknetconv2d\_BN\_Leaky，是yolo\_v3的基本组件。就是卷积+BN+Leaky\_relu。对于v3来说，BN和leaky\_relu已经是和卷积层不可分离的部分了(最后一层卷积除外)，共同构成了最小组件。

**(而BN就是通过一定的规范化手段，把每层神经网络任意神经元这个输入值的分布强行拉回到均值为0方差为1的标准正态分布，其实就是把越来越偏的分布强制拉回比较标准的分布，这样使得激活输入值落在非线性函数对输入比较敏感的区域，这样输入的小变化就会导致损失函数较大的变化，意思是这样让梯度变大，避免梯度消失问题产生，而且梯度变大意味着学习收敛速度快，能大大加快训练速度)**

（2）resn：n代表数字，有res1，res2, … ,res8等等，表示这个res\_block里含有多少个res\_unit。这是yolo\_v3的大组件，yolo\_v3开始借鉴了ResNet的残差结构，使用这种结构可以让网络结构更深(从v2的darknet-19上升到v3的darknet-53，前者没有残差结构)。对于res\_block的解释，其基本组件也是DBL，DBL也是其中的一个小组件。

（3）concat：张量拼接。将darknet中间层和后面的某一层经过上采样之后进行拼接。拼接的操作和残差层add的操作是不一样的，拼接会扩充张量的维度，而add只是直接相加不会导致张量维度的改变。

至于为什么要进行拼接，这里在前面的改进地方已经给出了解释，就是为了对不同尺寸的目标进行特征提取，从而增大对(不同尺寸的)目标的识别能力，这也是YOLO V3相较于V2版本的一大改进。

## 四、YOLO系列和RCNN系列的简单对比

（1）统一网络结构

YOLO没有求取region proposal的过程。

Faster R-CNN中尽管RPN与fast rcnn共享卷积层，但是在模型训练过程中，需要反复训练RPN网络和fast rcnn网络。相对于R-CNN系列的"两个阶段"(候选框提取与分类，图示如下),YOLO只需要一步完成。

（2）YOLO统一为一个回归问题

而R-CNN将检测结果分为两部分求解：物体类别（分类问题），物体位置即bounding box（回归问题）

