模型分为两部分：预测部份 + 训练部分

预测部分：

1、backbone： 主干特征提取

2、neck： 加强特征提取，特征融合

3、head：利用head获得预测结果

4、将预测结果解码：①获得预测框与得分；②得分筛选与非极大抑制



训练部分：

1、计算loss需要的内容

计算loss实际上是 **网络预测结果** 和 **网络真实结果** 的对比

和网络预测结果一样，网络损失也分为三部分：reg部分、obj部分、cls部分。reg是**特征点的回归参数判断，**obj是**特征点是否包含物体的判断**，cls是**特征点包含物体的种类**

网络**最后输出的内容就是三个特征层每个网格点对应的预测框及其种类**，即**三个特征层**分别对应着图片被分为**不同size的网格**后，**每个网格点上三个先验框对应的位置、置信度及其种类。**

2、正样本的匹配过程

采用各种方式去划分正样本和负样本

3、计算loss

**先验框，预测框，真实框**

1）预测阶段，根据需求设置一大批先验框，

2）经过网络，获得三个特征层，即将原图按照三种不同大小划分成对应网格，例如（13，13，255）、（26，26，255）、（52，52，255），然后每个网格中心点建立三个先验框

3）**网络的预测结果**会判断这些框内是否包含物体，以及这个物体的种类

由于每一个网格点都具有三个先验框，所以上述的预测结果可以reshape为：(N,13,13,3,85)、(N,26,26,3,85)、(N,52,52,3,85)

其中的85可以拆分为4+1+80，其中的4代表先验框的调整参数，1代表先验框内是否包含物体，80代表的是这个先验框的种类，由于coco分了80类，所以这里是80。如果YoloV3只检测两类物体，那么这个85就变为了4+1+2 = 7。

即85包含了4+1+80，分别代表x\_offset、y\_offset、h和w、置信度、分类结果。

4）但是这个预测结果并不对应着最终的预测框在图片上的位置，还**需要解码**才可以完成。

YoloV3的解码过程分为两步：

先将**每个网格点**加上它**对应的x\_offset和y\_offset**，**加完后的结果就是预测框的中心**。

然后再利用 **先验框和h、w结合 计算出预测框的宽高**。这样就能得到整个预测框的位置了。

总结下来就是：

第1，2步根据需求设定先验框参数，获取特征层

第3步得到网络的预测结果，即这些先验框要如何移动，先验框中包不包含物体，种类是什么

第4步 **根据预测结果 对先验框进行微调，便得到预测框，此时两种框数量是一样的**

**我们需要进行得分筛选 + 非极大抑制 从而得到最终的预测框，并绘制的原图中**

而在**训练过程中才涉及到真实框 与 先验框 + 预测框之间的 纠葛**，毕竟预测阶段哪来的手工标注的真实框，哈哈哈哈哈

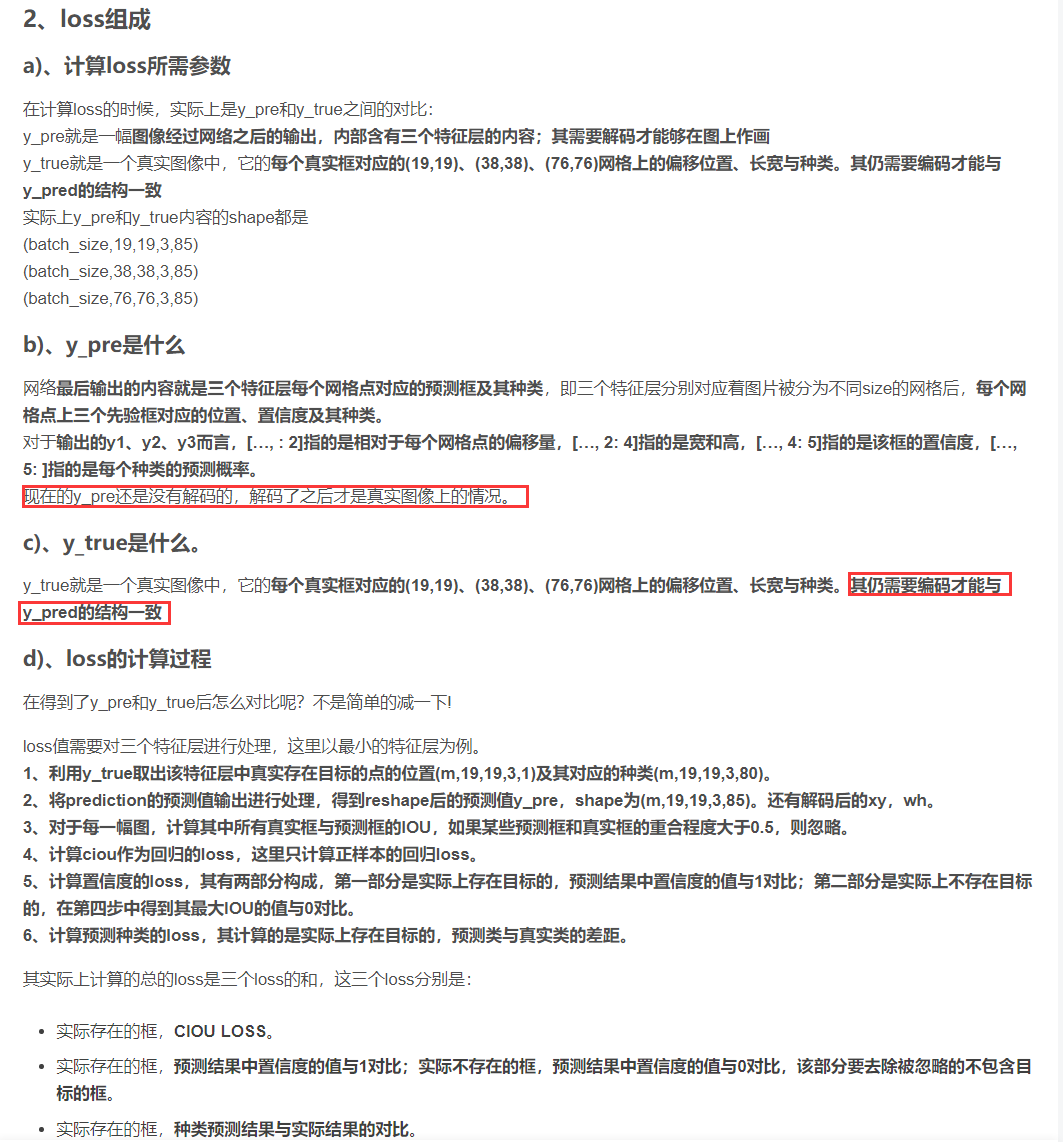
训练过程主要是通过 计算loss（实际上是 **网络预测结果** 和 **网络真实结果** 的对比），来不断的优化模型

dd

**y\_pred 与 y\_true的关系**

概括：





**对真实框的处理：**

https://blog.csdn.net/weixin\_44791964/article/details/106872072?spm=1001.2014.3001.5502

