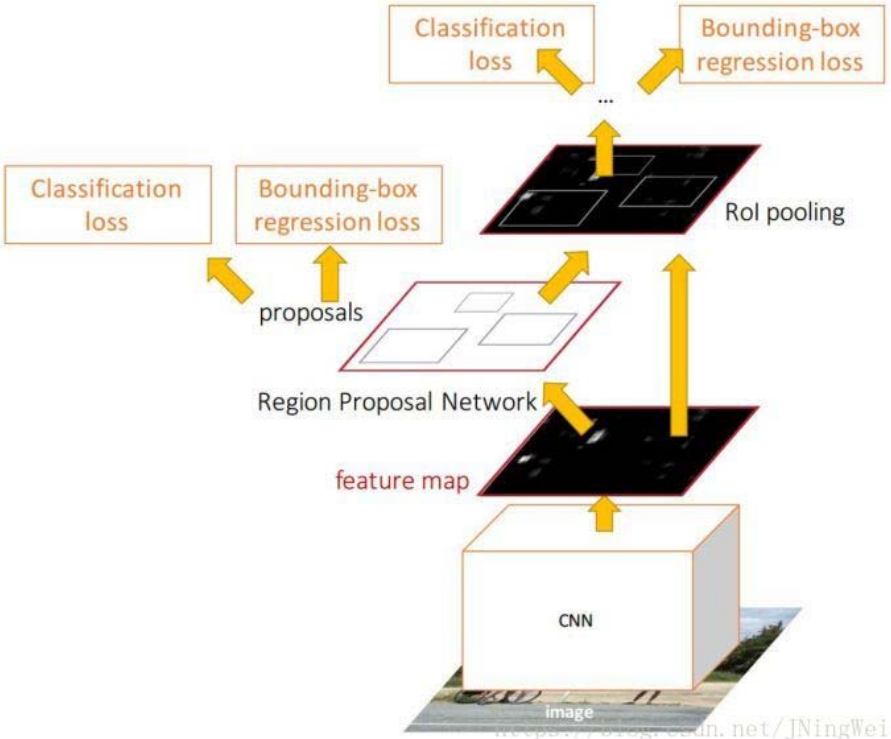
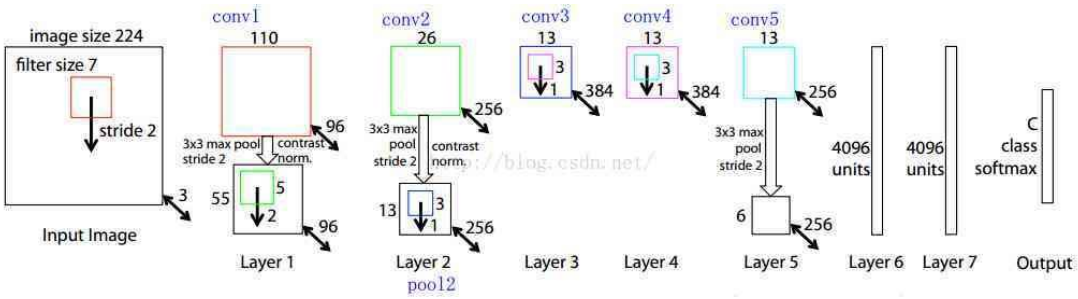
2018 年秋-第 9 周

FASTER -RCNN:

1、首先给出一个 faster rcnn 的宏观框架



2、从 image 到 feature map，经过一个 cnn，这个 cnn 可以是任意一个分类网络

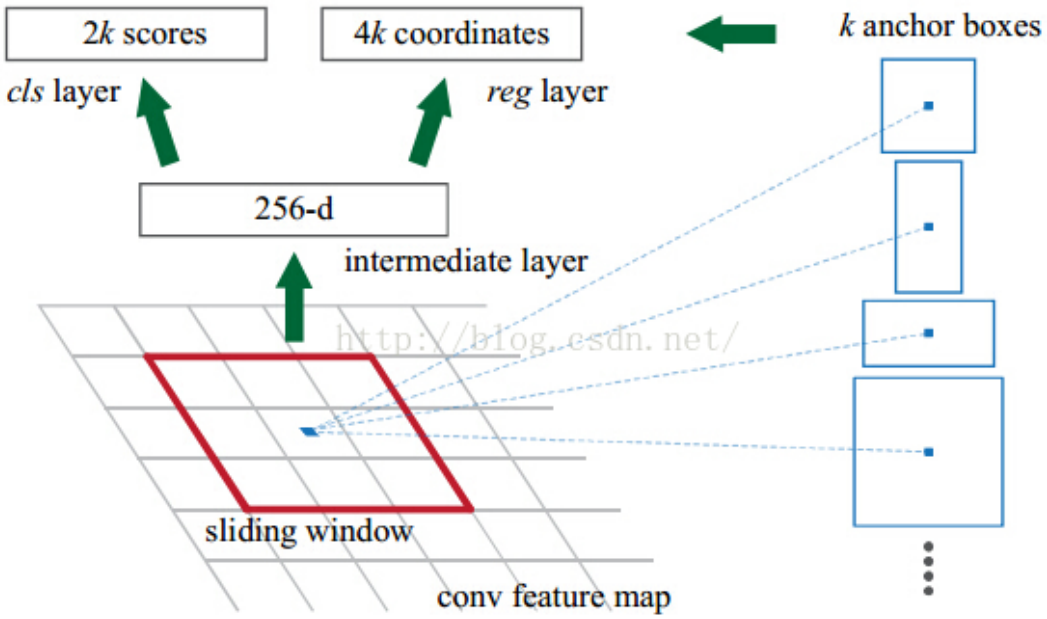


比方说上面是 ZF 网络结构，这个网络具体的卷积、池化推导部分在此省略

关注的部分：输入图片大小是 224*224*3，取 conv5 的输出，也就是 13*13*256 送给 RPN 网络

即原始图像（224*224*3）经过一个 cnn（ZF）的部分层之后的 conv5 我们把它取出来作为 feature map（13*13*256）

3、然后我们讨论 RPN 以及与之相关的 anchor 机制



上面这个图给出了 RPN 的结构

我们在上一步得到，这个 conv5 feature map 的维度是 13*13*256

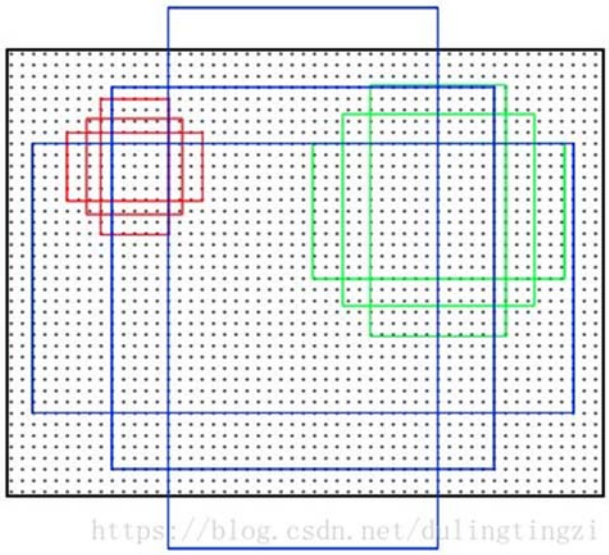
作者选用一个大小是 3*3 的 sliding window，用一个 3*3*256*256 这样的一个 4 维的卷积核，就可以将每一个 3*3 的 sliding window 卷积成一个 256 维的向量

k=9（这个放在后面解释）所以 cls layer 就是 18 个输出节点了，那么在 256-d 和 cls layer 之间使用一个 1*1*256*18 的卷积核，就可以得到 cls layer

reg layer 也是一样，reg layer 的输出是 36 个，所以对应的卷积核是 1*1*256*36，这样就可以得到 reg layer 的输出

/**********

【anchor 机制】这个是一个十分重要的部分，单独拿出来



这是网上的一个图，我不得不说这个图让我对 anchor 机制有了一些误解（当然也有可能是我还没有完全弄明白）

anchor 的本质：本质是 SPP 思想的逆向。SPP 是将不同尺寸的输入 resize 成为相同尺寸的输出。SPP 的逆向是将相同尺寸的输出，倒推得到不同尺寸的输入

anchor 的窗口尺寸，三个面积尺寸，然后在每个面积尺寸下，取三种不同的长宽比例（1:1,1:2,2:1）.这样一来得到了一共 9 种面积尺寸各异的 anchor，所以 k=9

一个 3x3 的滑动窗口，在这个 mxn 的区域上进行滑动，stride=1，padding=2，这样一来，滑动得到的就是 mxn 个 3x3 的窗口,如果跟前面对应的话是 m=n=13.

对于每个 3x3 的窗口，作者就计算这个滑动窗口的中心点所对应的原始图片的中心点。然后作者假定，这个 3x3 窗口，是从原始图片上通过 SPP 池化得到的，而这个池化的区域的面积以及比例，就是一个个的 anchor。换句话说，对于每个 3x3 窗口，作者假定它来自 9 种不同原始区域的池化，但是这些池化在原始图片中的中心点，都完全一样。这个中心点，就是刚才提到的，3x3 窗口中心点所对应的原始图片中的中心点。（而刚才那个图明显不是同一个中心点，所以造成了误解）

如此一来，在每个窗口位置，我们都可以根据 9 个不同长宽比例、不同面积的 anchor，逆向推导出它所对应的原始图片中的一个区域，这个区域的尺寸以及坐标，都是已知的。而这个区域，就是我们想要的 proposal。所以我们通过滑动窗口和 anchor，成功得到了 mxnx9 个原始图片的 proposal。

接下来，每个 proposal 我们只输出 6 个参数：每个 proposal 和 ground truth 进行比较得到的前景概率和背景概率(2 个参数)，这个对应 2k；由于每个 proposal 和 ground truth 位置及尺寸上的差异，从 proposal 通过平移放缩得到 ground truth 需要的 4 个平移放缩参数，这个对应 4k。

**********/

/**********

再插入一个 anchors 的标定方法：

1 如果 Anchor 对应的 reference box 与 ground truth 的 IoU 值最大，标记为正样本；

2 如果 Anchor 对应的 reference box 与 ground truth 的 IoU>0.7，标记为正样本.

事实上，采用第 2 个规则基本上可以找到足够的正样本，但是对于一些极端情况，例如所有的 Anchor 对应的 reference box 与 groud truth 的 IoU 不大于 0.7,可以采用第一种规则生成.

3 负样本标定规则：如果 Anchor 对应的 reference box 与 ground truth 的 IoU<0.3，标记为负样本.

4 剩下的既不是正样本也不是负样本，不用于最终训练.

**********/

最后又回到这个小节标题，在这一部分我们知道 RPN 的输入，那它最终产生的有意义的输出是什么：

Input Image 经过 CNN 特征提取，首先来到 Region Proposal 网络。由 Regio Proposal Network 输出的 Classification，这并不是判定物体在数据集上对应类中哪一类，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。

如果一个 Region 的 p 大于等于 0.5 ，则认为这个 Region 中可能是某一类，具体是哪一类现在还不清楚。到此为止，Network 只需要把这些可能含有物体的区域选取出来就可以了，这些被选取出来的 Region 又叫做 ROI，即感兴趣的区域。RPN 同时也会在 feature map 上框定这些 ROI 感兴趣区域的大致位置，即输出 Bounding-box。

4、最后是最上层的 RoI Pooling

我们可以认为上一步 RPN 可以等效于让我们在 feature map 上找到感兴趣的区域，然后我们把它截取下来，这个区域一定比 feature map 本身小，并且是不规则的，而后面有全连接层，所以要统一尺度

ROI pooling layer 实际上是 SPP-NET 的一个精简版，实现从原图区域映射到 conv5 区域最后 pooling 到固定大小的功能， 然后就是后面的分类和回归了

5、如何训练 faster rcnn

上面几部分我觉得把模型本身讲清楚了，最后就是考虑如何训练的问题。

这里我直接摘抄了一个博客的描述：

第一步：用 model 初始化 RPN 网络，然后训练 RPN，在训练后，model 以及 RPN 的 unique 会被更新。

第二步：用 model 初始化 Fast-rcnn 网络，注意这个 model 和第一步一样。然后使用训练过的 RPN 来计算 proposal，再将 proposal 给予 Fast-rcnn 网络。接着训练 Fast-rcnn。训练完以后，model 以及 Fast-rcnn 的 unique 都会被更新。

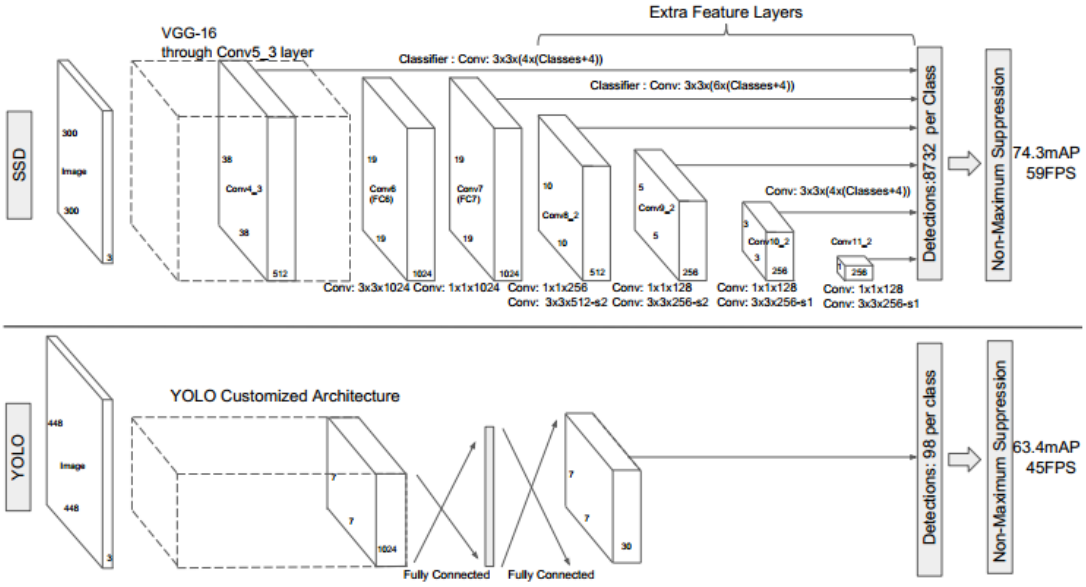
说明：第一和第二步，用同样的 model 初始化 RPN 网络和 Fast-rcnn 网络，然后各自独立地进行训练，所以训练后，各自对 model 的更新一定是不一样的（论文中的 different ways），因此就意味着 model 是不共享的（论文中的 dont share convolution layers）。

第三步：使用第二步训练完成的 model 来初始化 RPN 网络，第二次训练 RPN 网络。但是这次要把 model 锁定，训练过程中，model 始终保持不变，而 RPN 的 unique 会被改变。
说明：因为这一次的训练过程中，model 始终保持和上一步 Fast-rcnn 中 model 一致，所以就称之为着共享。

第四步：仍然保持第三步的 model 不变，初始化 Fast-rcnn，第二次训练 Fast-rcnn 网络。其实就是对其 unique 进行 finetune，训练完毕，得到一个文中所说的 unified network。

SSD：

1、首先给出 SSD 结构图



上图是原论文中的 SSD 300 网络结构图，SSD 采用特征金字塔结构进行检测，检测时利用了 conv4-3，conv-7（FC7），conv6-2，conv7-2，conv8_2，conv9_2 这些大小不同的 feature maps，在多个 feature maps 上同时进行 softmax 分类和位置回归

SSD 的突出贡献：SSD 使用低层 feature map 检测小目标，使用高层 feature map 检测大目标

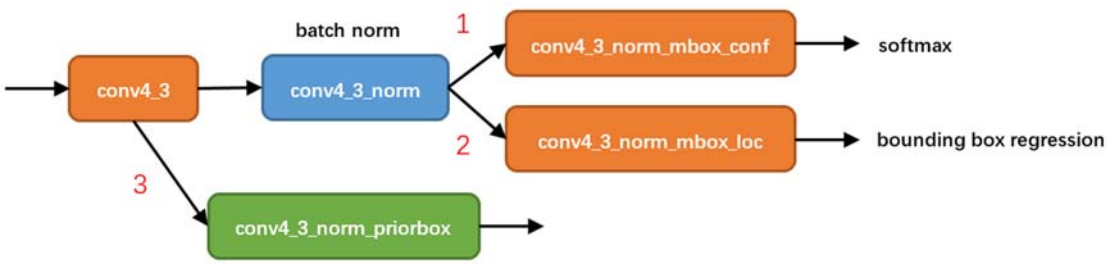
2、Prior Box（从对比角度分析）

SSD 中引入了 Prior Box，实际上与 anchor 非常类似，就是一些目标的预选框，后续通过 softmax 分类+bounding box regression 获得真实目标的位置。

具体的 Prior Box 生成规则截图如下：

- 以feature map上每个点的中点为中心（offset=0.5），生成一些列同心的prior box（然后中心点的坐标会乘以step，相当于从feature map位置映射回原图位置）
- 正方形prior box最小边长为min_size，最大边长为：$\sqrt{min\_size * max\_size}$
- 每在prototxt设置一个aspect ratio，会生成2个长方形，长宽为：$\sqrt{aspect\_ratio} * min\_size$ 和 $1/\sqrt{aspect\_ratio} * min\_size$

Prior Box 的使用：



在 conv4_3 feature map 网络 pipeline 分为了 3 条线路：

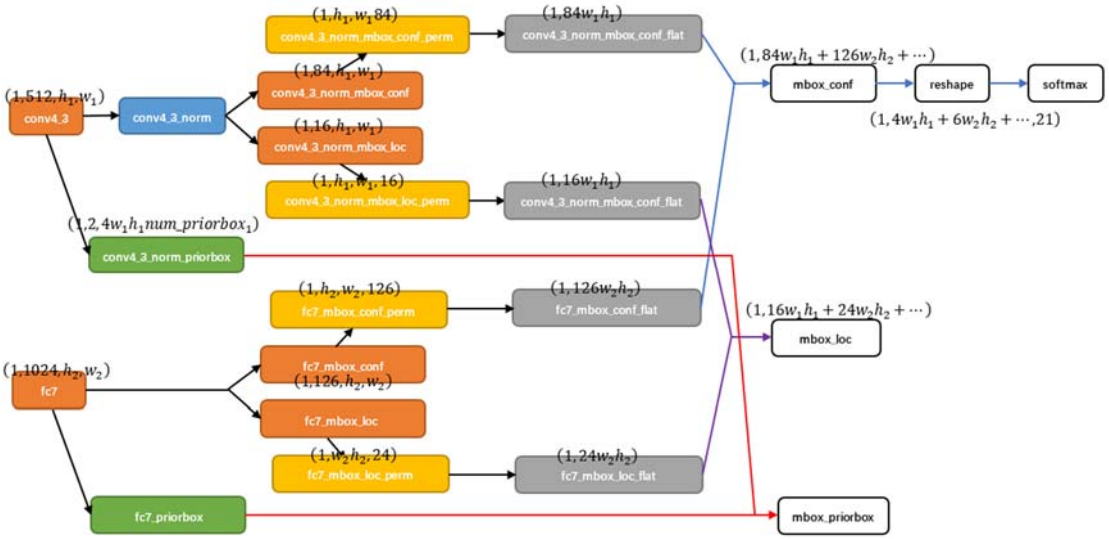
经过一次 batch norm+一次卷积后，生成了[1,num_class*num_priorbox, layer_height, layer_width]大小的 feature 用于 softmax 分类目标和非目标（其中 num_class 是目标类别，SSD 300 中 num_class = 21)

经过一次 batch norm+一次卷积后，生成了[1,4*num_priorbox, layer_height, layer_width]大小的 feature 用于 bounding box regression(即每个点一组[dxmin,dymin, dxmax, dymax])

生成了[1, 2, 4*num_priorbox]大小的 prior box blob，其中 2 个 channel 分别存储 prior box 的 4 个点坐标和对应的 4 个 variance

后续通过 softmax 分类+bounding box regression 即可从 priox box 中预测到目标

3、多个 feature maps 如何协同工作



4、SSD 训练

Matching strategy:

在训练时，groundtruth boxes 与 default boxes（就是 prior boxes） 按照如下方式进行配对：

首先，寻找与每一个 ground truth box 有最大的 jaccard overlap 的 default box，这样就能保证每一个 groundtruth box 与唯一的一个 default box 对应起来

SSD 之后又将剩余还没有配对的 default box 与任意一个 groundtruth box 尝试配对，只要两者之间的 jaccard overlap 大于阈值，就认为 match（SSD 300 阈值为 0.5）。

配对到 GT 的 default box 就是 positive，没有配对到 GT 的 default box 就是 negative。

Hard negative mining:

一般情况下 negative default boxes 数量>>positive default boxes 数量，直接训练会导致网络过于重视负样本，从而 loss 不稳定。所以 SSD 在训练时会依据 confidience score 排序 default box，挑选其中 confidience 高的 box 进行训练，控制 positive:negative=1:3