本周主要在看 faster rcnn 和 ssd 网络的两篇论文，结合一些博客对两种网络中不容易理解的点做了一些总结。Faster rcnn 主要参考了 https://antkillerfarm.github.io/dl/2017/09/06/Deep_Learning_15.html，https://antkillerfarm.github.io/dl/2017/09/09/Deep_Learning_16.html，ssd 参考了 https://blog.csdn.net/wfei101/article/details/78176322。

对 Faster Rcnn 的一些总结：

Single-scale image：文中指以短边长为 600 进行缩放，对比于金字塔形状的图片多个 re-scale 为 multiple scales。

RPN 部分：进行 reshape 是为了单独“腾空”出来一个维度以便 softmax 分类(caffe 框架下)，之后再 reshape 回复原状。 在训练过程中 RPN 会随机取 256 个 anchor（不同形状的 anchor 算作不同的 anchor）作为一个 mini-batch，尽量保证其中正(IoU>0.7)负(IoU<0.3)样本各一半，

Fine-tune:使用预训练过的网络在自己的数据集进行训练

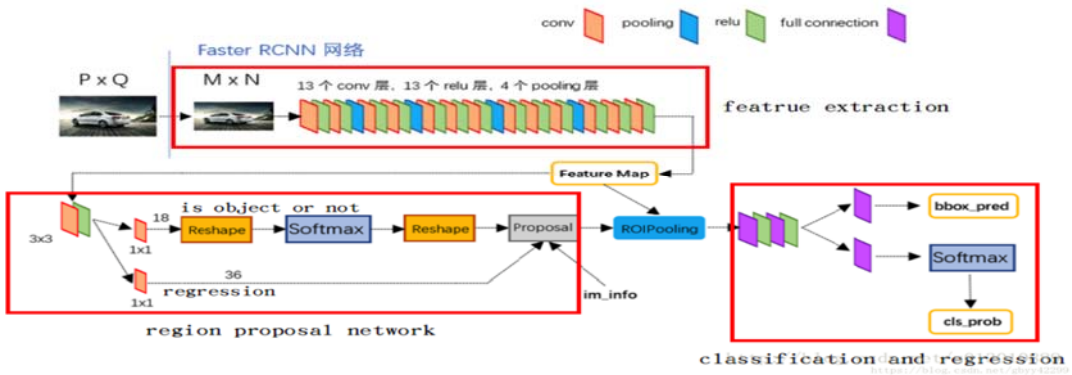
Sharing features for RPN and Fast R-CNN:通过四步训练网络，来使 RPN 和 Fast R-CNN 共用部分 features。1、单独训练 RPN 网络 2、使用第一步得到的 proposal 数据训练 Fast R-CNN 网络（相当于不改变 RPN 模块的参数） 3、固定其他部分参数只通过训练微调 RPN 模块的参数 4、固定其他部分参数，只微调 classification and regression 部分参数。 其中 1,2 步骤的 VGG 部分使用 ImageNet 预训练的数据。

mAP 计算:AP 是一个类的 average precision，mAP 是所有类平均 AP。AP：有一种算法是每个 recall 中最大的 precision 求平均。

Anchor 作用:每个滑框都能生成多个不同 scale 和纵横比的可能区域，由此来确保尽可能的找到更多的目标出现区域。得到大致区域后通过 fast rcnn 模块进行进一步修正来取得更好的结果。

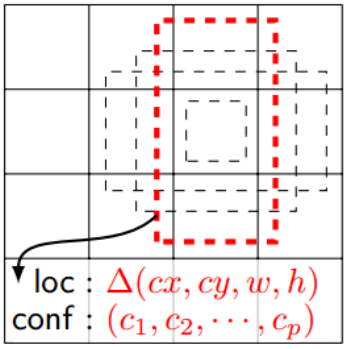
Roi pooling:把不同大小的输入映射到一个固定尺度的特征向量。根据输入的 image，将 Roi 映射到 feature map 对应的位置；将映射后的区域划分为相同大小的 sections（sections 数量和输出的维度相同）；对每个 section 进行 max pooling 操作。

关于边框回归:使预测框更接近于 ground truth 框，调整的变量为中心点的偏移量以及长宽的缩放比共四个，通过反向传播减少四个值所构成的 loss 来对 proposal 位置进行优化。



Feature extraction 是 VGG16 网络 conv 部分，对 conv5_3 后得到的 feature map 进行处理。 RPN 部分将得到的 512 层的 feature map 裁剪出一个 3*3*512 的滑框，这个滑框经过 3*3 的卷积可得到 1*1*512 的 feature maps，再通过 1*1 卷积层进行降维，上方 18 表示 2（两类：是/否为有效框）*9（3 个 scale*3 个纵横比）层 feature map，下方 36 表示 4（左上点和右下点的坐标）*9 层 feature map，相当于 fc 层。最终得到带有正负标签的 proposal 框。 将 proposal 框映射到 conv5_3 后的 feature map 上，经过 ROI Pooling 得到一个固定尺度的特征向量，经过两个 FC 层后分成两个输出层：第一个是是针对每个 ROI 区域坐标的偏移进一步优化，第二个针对每个 ROI 区域的分类概率预测（softmax），两个输出层合并可得到一个统一的 loss，用此信息进行参数的训练。

对 SSD 的总结：

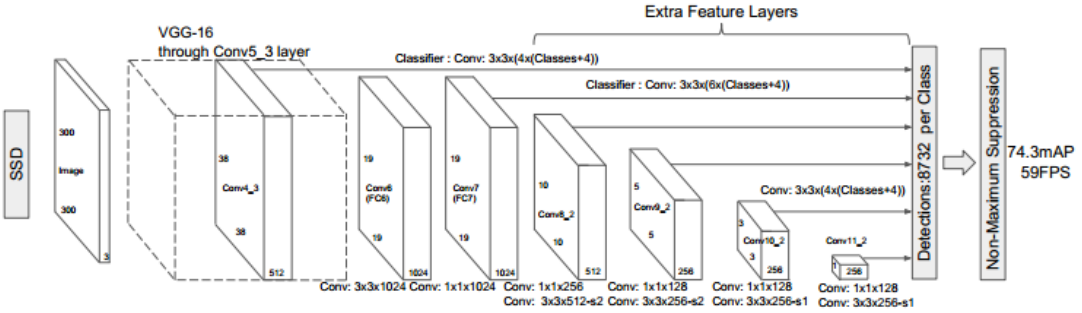


feature map cell: feature map 中的每个小格就是一个 feature map cell，上图就有 16 个 feature map cell，每个 feature map cell 都有一系列的默认规格 box，这些 box 映射到原图上就会生成不同位置不同规格的 bounding box。

Matching strategy: 将与 ground truth box 交并比大于 0.5 的所有 default box 均看做匹配。

损失函数: 在计算置信度的损失函数时，考虑了正样本和负样本（目标和背景）二者的置信度，将第 0 类设为负样本，和 p 类正样本共同做 softmax。

Hard negative mining: 对每一个 default box 求置信度损失进行排序，对于候选正样本集：选择 loss 最高的几个 default box 与正样本集匹配，匹配不成功则删除这个正样本（因为这个正样本不在难例里已经很接近 ground truth box 了，不需要再训练了）；对于候选负样本集：选择 loss 最高的几个 prior box 与候选负样本集匹配，匹配成功则作为负样本。最终确保正、负样本的比例在 1：3 左右。



通过对 VGG16 的网络进行修改来构造 SSD 网络。类似于 faster rcnn 中 anchor，将六个不同的 feature maps 中每一个 cell 做 3*3 的卷积，一共可以得到 38*38*4+19*19*6+10*10*6+5*5*6+3*3*4+1*1*4=8732 个分类器（前两个数表示 feature map 的高宽，第三个数表示每一层中选取的不同 scale 的 feature map），最后通过非最大值抑制得到预测结果。

对网络结构的理解结合 pipeline 的图来说会变得更容易，能搞清楚每层网络的输入输出很重要，由于 RPN 那部分的结构没有想对，网上也有一些博文对结构的描述也有错误，花了很久才明白，看懂了 faster rcnn 后 SSD 就比较容易理解了。