Fast R-CNN阅读笔记

RGB大神独立完成的一篇文章，对比之前介绍的R-CNN和SPPnets，在速度和精度上均有提升。

论文：https://arxiv.org/pdf/1504.08083.pdf
源码：https://github.com/rbgirshick/fast-rcnn

1. 遗留问题和提升方法

如R-CNN所介绍的，作者分析了R-CNN和SPPnets的三个缺点：

1. 训练过程复杂
   - 原因：多级管道的使用
   - 解决方案: 使用多任务代替多管道
2. 训练速度慢，占用磁盘
   - 原因：卷积提取ROI的特征后存储在磁盘上，然后将特征送入SVM做分类
   - 解决方案：整张图片共同提取特征，通过计算将ROI区域的坐标映射到特征层。
3. 检测速度慢
   - 原因：图片的不同ROI区域有大量的冗余，每个ROI单独提取特征造成了大量的冗余计算。
   - 解决方案：同上

2. Fast R-CNN算法

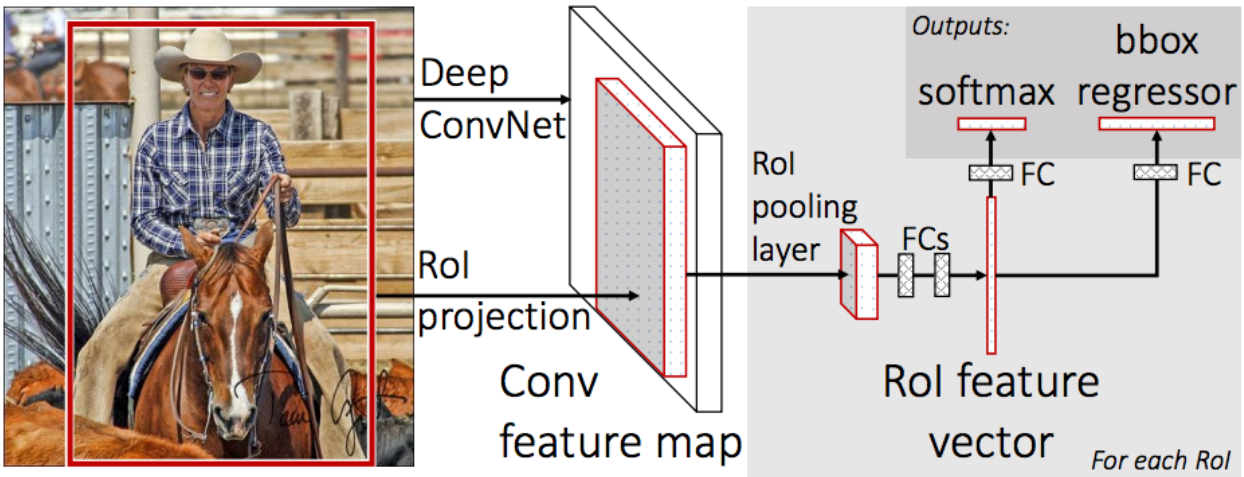

图1：Fast R-CNN网络结构

上图显示的是Fast R-CNN的网络结构，

1. 数据：图像，图像的类别（分类任务），图像ROI（通过Selective Search得到的红框，[x, y, w, h] 四个实数值，用于回归任务）；
2. 输入：原始图像（左侧）；
3. 网络结构：在特征层部分，使用卷积网络提取特征，使用SPPnets的思想提取相同大小的ROI Pooling层，两层全连接后输出图像的特征向量;
4. 输出：softmax用于识别物体类别，bbox regressor用于检测物体位置。

下文将通过网络结构详细分析Fast R-CNN算法。

2.1 数据

在物体检测领域，有很多经典的数据集，例如论文中使用的Pascal Voc，以及Microsoft提供的coco，kitti数据集等等。

Pascal Voc的图像数据并不是像Cifar-10或者MNIST等所有图像的分辨率都是固定的。其数据标签组织在.xml文件里面，下面是Pascal数据集000001.jpg的图片（图2）和其对应的Ground Truth文件000001.xml (图3). 详细内容已通过注释的形式添加到文件中

图2：Pascal 数据集000001.jpg

2.2 输入

和SPPnets类似，Fast R-CNN的输入是未经过resize的原始图像，从而避免图像形变或者加边带来的负面影响

2.3 网络结构

表示层也由AlexNet换成效率更高，结构更复杂的VGG网络，此处不再赘述。

Fast R-CNN非常重要的的创新点是ROI Pooling layer的提出。下面结合源码详细介绍一下ROI Pooling Layer。

每个ROI区域是卷积层的一个矩形区域，有Selective Search在图像提取的ROI区域映射而来，由一个四元组$v = (v_x, v_y, v_w, v_h)$确定的，其中(v_x, v_y)表示ROI左上角，(v_h, v_w)表示ROI区域的长和宽。给定ROI层的输出矩阵的长宽的超参数(H, W), 将ROI区域分成$(v_h/H \times v_w/W)$ 的grid, 论文采用的是max pooling, 即将每个grid的最大值作为输出。另外，u表示该ROI矩形区域对应的K+1类物体类别。

ROI特征层后接两个全连接后输出的便是ROI特征向量。网络的输出是softmax的用于判断检测物体类别的分类任务和Smooth L1 Loss用于确定物体位置的回归任务。

```
<annotation>
    <folder>VOC2007</folder>
     <filename>000001.jpg</filename>
     <source>
       <database>The VOC2007 Database</database>
       <annotation>PASCAL VOC2007</annotation>
       <image>flickr</image>
       <flickrid>341012865</flickrid>
     </source>
     <owner>
       <flickrid>Fried Camels</flickrid>
       <name>Jinky the Fruit Bat</name>
     </owner>
     <size> <!--image shape-->
       <width>353</width>
       <height>500</height>
       <depth>3</depth>
     </size>
     <segmented>0</segmented> <!--是否有分割label-->
     <object>
       <name>dog</name> <!--类别-->
       <pose>Left</pose> <!--物体的姿态-->
       <truncated>1</truncated> <!--物体是否被部分遮挡（>15%）-->
       <difficult>0</difficult> <!--是否为难以辨识的物体， 主要指要结体背景才能判断出类别的物体。虽有标注， 但一般忽略这类物体-->
       <bndbox> <!--bounding box-->
              <xmin>48</xmin>
              <ymin>240</ymin>
              <xmax>195</xmax>
              <ymax>371</ymax>
       </bndbox>
     </object>
     <object>
       <name>person</name>
       <pose>Left</pose>
       <truncated>1</truncated>
       <difficult>0</difficult>
       <bndbox>
              <xmin>8</xmin>
              <ymin>12</ymin>
              <xmax>352</xmax>
              <ymax>498</ymax>
       </bndbox>
     </object>
</annotation>
```

图3: 00001.xml

2.4 输出

上文中我们已有介绍, 输出层有分类和回归两个任务，所以在Fast R-CNN中使用的是多任务的损失函数。针对每个ROI区域，$p = (p_0, \ldots, p_K)$表示K+1个不同的类别（K类物体+背景），针对每个非背景类别的ROI，四元组$t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$表示物体的预测位置。在Fast R-CNN中，损失函数表示为：

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

其中，$L_{cls}(p, u) = -logp_u$.
L_{loc}表示为：

$$L_{loc}(t^u, v) = \sum_{i \in (x,y,w,h)} smooth_{L_1}(t_i^u - v_i)$$

其中：

$$smooth_{L_1}(x) = \begin{cases} o.5x^2 & if\,|x| < 1 \\ |x| - 0.5 & otherwise, \end{cases}$$

smooth l1的使用可以增强模型对异常数据的鲁棒性。

在实验中，v_i均归一化到0均值，单位方差，λ=1.

3. 训练细节

3.1. 迁移学习

网络的卷积层使用的是在ImageNet上预训练的网络结构

3.2 Mini-batch 抽样

在Fast R-CNN中，每次抽样的样本batch是n，其中每次随机选m张图片，再每张图片中再通过selective search在原始图片提取n/m个ROI区域，实验中，n=128, m=2，ROI区域正样本的比例在25%。

所谓正样本，是指ROI区域和Ground Truth区域重合的比例在50%以上，计算过程在utils/bbox.pyx文件中实现。

4. 检测

4.1 检测过程

1. Selective Search提取2000个ROI区域;
2. 预测每个ROI区域的类别和相对坐标；
3. 使用nms (non-maximum suppression) 合并ROI，得到最终结果.

4.2 Truncated SVD用于提速

在Fast R-CNN的网络中，存在若干个全连接网络，可以表示为Y = WX, SVD相当于将将W分解成两个小矩阵，中间用一个低维矩阵连接。

Reference

[1]. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[2]. http://blog.csdn.net/weixin_35653315/article/details/71028523

[3]. Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 38(1): 142-158.

[4]. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)

[5]. http://blog.csdn.net/shenxiaolu1984/article/details/51036677

[6]. https://zhuanlan.zhihu.com/p/24780395?refer=xiaoleimlnote

[7]. http://shartoo.github.io/RCNN-series/