Fast R-CNN

《Fast R-CNN》 Ross Girshick Microsoft Research rbg@microsoft.com
http://blog.csdn.net/u010678153/article/details/46891655
http://blog.csdn.net/wonder233/article/details/53671018

摘要

用深度卷积网络高效的分类物体候选域
比R-CNN,SPP快

1.介绍

物体检测的两个挑战：

- 许多候选域必须要被处理
- 候选域的位置不够精确，一般需要被调整

一个一步的训练算法去分类候选域和调整候选域的位置
RCNN的9倍快，SSP的3倍快，运行时，处理一张图片只要0.3s ,精确率：在PASCAL VOC 2012上mAP 66%

（说R-CNN坏话）
R-CNN的缺点：
1. **训练分为几个阶段**，先是在物体候选域上用log损失微调卷积网络，然后给卷积网络特征上加一个SVM，这些SVM替代通过微调学到的softmax分类器，第三阶段，学习Bounding-box回归
2. **训练空间时间耗费大**：SVM和Bounding-box回归训练特征要从每个候选域提取出来，写到磁盘上
3. **物体检测慢**：测试阶段，要从每个图片的每个候选域提取特征, VGG16 在GPU上，每张图片要47s

SSP是为每张图片计算一个feature map ,然后在feature map中提取 候选域特征用于分类，共享计算

（说SSP坏话）
SSP的缺点：
1. **训练分为多个阶段**：提取特征，微调网络，训练SVM，Bounding-box回归
2. **特征要被写入磁盘**

---

fast R-CNN的优点：
1. mAP比R-CNN,SSP高
2. 训练是单阶段的，用了multi-task损失
3. 训练可以更新网路的所有层
4. 特征缓存不需要磁盘

---

caffe下的python代码：https://github.com/rbgirshick/fast-rcnn

2.fast R-CNN结构和训练

网络结构：

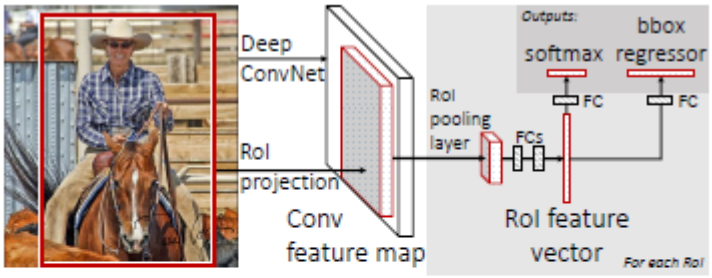


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

输入整张图片和一些物体候选域，通过一些卷基层和池化层产生整张图片的feature map，为每个RoI,池化层从feature map中提取出固定长度的feature 向量，然后把特征向量输入到一系列的全连接层，输出分为两部分，一个是softmax 输出的K+1类的的概率，另一个是为每个类输出一个四元组，用于调整bounding-box的位置

RoI池化层

用最大池把任何感兴趣非空区域转化成空间上$H \times W$的feature map.H,W是跟特定RoI独立的层的超参数。
RoI是feature map的矩形窗。（r,c,h,w）(r,c)左上角的坐标，（h,w）高和宽
RoI max pooling把 $h \times w$ 的RoI分成$H * W$个网格，每个网格的大小近似为：$h/H \times w/W$,然后在每个子窗口中做最大池化，输出跟网格大小一致
池化被独立的应用在每个feature map通道
RoI池化是SSP空间金字塔池化的一种特殊情况，池化子窗口计算

从预训练的网络初始化

用三个预训练的ImageNet网络实验，每个网络有5个池化层，5到13个卷积层，用预训练的网络初始化一个fast R-CNN要经过3个转化
1. 最后的最大池化层要替换成RoI池化层，通过设置H,W跟第一个全连接层匹配，VGG16，H=W=7
2. 网络的最后一个全连接层和1000类的softmax要替换成两个层，全连接层和K+1类的softmax；特定类的bounding-box回归
3. 网络的输入要改为：图片和图片的RoIs

微调

网络所有的参数训练都是用反向传播的

（说SSP坏话）
SSP为什不能更新空间金字塔池化层（SPP）下面的层：
????*（不懂，等下看了SSP看能不能懂）*
根本原因是：通过SPP层的反向传播是高度无效的，当每个训练样本（RoI）来自不同的图片时。每个RoI有个非常大的感受野，几乎是整张图片。

随机梯度下降的min-batch是分层采样的。
首先采样出N个图片，然后从每个图片中采样出R/N个RoI。
来自同一个图片的RoI共享前向和后向传播的计算和内存。
选择小的N可以降低min-batch的计算量。 **流线型训练过程，只有一步微调**
一步微调，联合优化Softmax分类和bounding-box回归，而不是分为三个阶段

Multi-task loss： fast-RCNN的两个输出：
对于每个RoI,属于每个类的概率 $p = (p_0, p_1, \cdots, p_K)$
bounding-box 回归的偏置 $t^k = (t^k_x, t^k_y, t^k_w, t^k_h)$, t^k表示一个尺度不变的转换和log空间宽和高的变换
u:RoI的真实类别，v:bounding-box回归的目标

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

$$L_{cls}(p, u) = -logp_u$$

$[u \geq 1]$ 当u=0时表示是背景，则为0，其他类为1，就是说背景的忽略位置损失

$$L_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L1}(t^u_i - v_i)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases}$$

L1损失对离异点不敏感，更加鲁棒
正则化bounding-box回归目标v_i均值为0，方差为1，λ设置为1

---

mini-batch采样：
SGD的mini-batch N=2,图片是统一随机选的，R=128,每张图片选64个RoI 25%的RoI是从物体的候选域中选的，这些候选域与真实的bounding-box的IoU大于等于0.5 ，这些IoU包含用前景物体标记的类别。
剩下的IoU从与真实物体的IoU在[0.1,0.5]的候选域中选最大的，这些IoU包含背景
训练时，图片会以0.5的概率水平翻转（ *？？为什么图片要翻转* ）

通过RoI池化层的反向传播：
假设N=1对每个mini-batch,x_i是输入到RoI池化层的第i个激活，y_{rj}是第r个RoI的第j个输出。

$$y_{rj} = x_{i^*(r,j)}, i^*(r,j) = argmax_{i' \in R(r,j)} x_{i'}$$

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r,j)] \frac{\partial L}{\partial y_{rj}}. \qquad (4)$$

SGD超参数：

softmax分类和bounding-box回归分别使用均值为0，方差为0.01，0.001的高斯分布初始化的，偏差初始化为0
在VOC07 or VOC12上前30k次学习率为0.001，后10k次学习率为0.0001.
使用了一个0.9的冲量和0.0005的参数衰减

尺度不变

单一尺度和通过图像金字塔的多尺度

3.Fast R-CNN检测

当使用图像金字塔时，每个被调整过尺度的RoI大概有224*224个像素
每个测试的RoI r,前向传播，输出类的后验概率分布p和针对每个类的bounding-box偏置，r属于k类的信度用概率表示，然后对每个类独立的使用非极大值抑制算法

截断SVD

具体如何实现的呢？

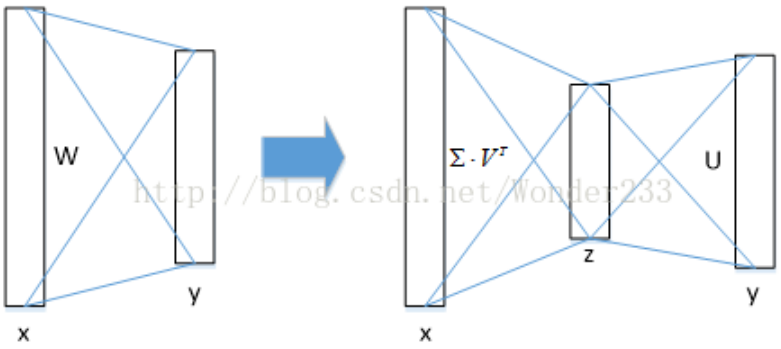
①物体分类和bbox回归都是通过全连接层实现的，假设全连接层输入数据为X，输出数据为Y，全连接层权值矩阵为W，尺寸为 u × v ，那么该层全连接计算为:

$$Y = W \times X$$

②若将W进行SVD分解（奇异值分解），并用前t个特征值近似代替，即:

$$W \approx U\Sigma_t V^T$$

U是 u × t 的左奇异矩阵，Σt 是 t × t 的对角矩阵 ，V是 v × t 的右奇异矩阵。



截断SVD将参数量由原来的 u × v 减少到 t × (u + v) ，当 t 远小于 min(u,v) 的时候降低了很大的计算量。

在实现时，相当于把一个全连接层拆分为两个全连接层，第一个全连接层使用权值矩阵 $\Sigma_t V^T$ （不含偏置），第二个全连接层使用矩阵U（含偏置）；

当RoI的数量大时，这种简单的压缩方法有很好的加速。

3 ．结果

1. 在VOC07,2010,2012上高的mAP
2. 相比R-CNN,SPP快的训练和测试

3. 在VGG16上微调提高mAP

实验设置

用到三个预训练的imageNet模型：
1. **S**: CaffeNet 本质的AlexNet R-CNN
2. **M**:VGG_CNN_M_1024,跟S深度一样，比S宽
3. **L**:VGG16

单一尺度训练

VOC 2010和2012上的结果

VOC2012上跟BabyLearning,R-CNN BB,NUS NIN c2000相比，Fast-RCNN的mAP最大：65.7% 68.4（加额外的数据）

VOC2010上SegDeepM(67.2%)的mAP比Fast-RCNN(66.1%)大.SegDeepM用到的训练数据是VOC12和分割注释。当Fast-RCNN的训练集为VOC07 训练，测试集，VOC12训练集时 mAP为68.8%

VOC07上的结果

比较Fast R-CNN， R-CNN ，SPPnet.
都是用VGG16预训练的， bounding-box回归 SSPnet在训练和测试时使用5个尺度

07\diff：在07数据集中去除困难的样本 12++07 ：训练集为VOC07 训练，测试集，VOC12训练集

方法	训练集	测试集	mAP
SPPnet	07\diff	07	63.1
R-CNN	07	07	66.0
Fast R-CNN	07	07	66.9
Fast R-CNN	07 \ diff	07	68.1
Fast R-CNN	07+ 12	07	70.0
R-CNN	12	10	62.9
Fast R-CNN	12	10	66.1
Fast R-CNN	12++07	10	68.8
R-CNN	12	12	62.4
Fast R-CNN	12	12	65.7

方法	训练集	测试集	mAP	
Fast R-CNN	07++12	12	68.4	

训练和测试时间

	Fast R-CNN			R-CNN			SPPnet
	S	M	L	S	M	L	†L
train time (h)	**1.2**	2.0	9.5	22	28	84	25
train speedup	**18.3**×	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	**0.06**	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	**213**×	-	-	-	-
VOC07 mAP	57.1	59.2	**66.9**	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

Table 4. Runtime comparison between the same models in Fast R-CNN, R-CNN, and SPPnet. Fast R-CNN uses single-scale mode. SPPnet uses the five scales specified in [11]. †Timing provided by the authors of [11]. Times were measured on an Nvidia K40 GPU.

阶段SVD:
VGG16网络，fc6的矩阵为25088×4096,使用前1024个奇异值
fc7的矩阵为4096×4096,使用前256个奇异值

那个层微调？

只在全连接层微调对于很深的网络行不通
VGG网络结构：http://blog.csdn.net/zhyj3038/article/details/52448102

	layers that are fine-tuned in model L			SPPnet L
	≥ fc6	≥ conv3_1	≥ conv2_1	≥ fc6
VOC07 mAP	61.4	66.9	**67.2**	63.1
test rate (s/im)	**0.32**	**0.32**	**0.32**	2.3

VGG16从cov3_1开始微调，mAP和时间均衡最好
S 和M网络从cov2开始微调

---

5 .设计评估

评估实验都在VOC07数据集进行

多任务训练有用吗？

	S				M				L			
multi-task training?		✓		✓		✓		✓		✓		✓
stage-wise training?			✓				✓				✓	
test-time bbox reg?			✓	✓			✓	✓			✓	✓
VOC07 mAP	52.2	53.3	54.6	**57.1**	54.7	55.5	56.6	**59.2**	62.6	63.4	64.0	**66.9**

第一列：用L_{cls}，只用到分类损失
第二列：用多任务损失，$\lambda = 1$
第三列：用L_{loc}和bounding-box 回归
第四列：用多任务损失和bounding-box 回归

尺度不变性

单一尺度和图像金字塔多尺度哪个更好？
图像的短边　s =600像素,短边少于６００像素的，就把长边设为１０００像素，保持纵横比不变　　ＲOI池化层的步长为１０像素
多尺度中用到五个尺度，$s \in \{480; 576; 688; 864; 1200\}$

	SPPnet ZF		S		M		L
scales	1	5	1	5	1	5	1
test rate (s/im)	0.14	0.38	**0.10**	0.39	0.15	0.64	0.32
VOC07 mAP	58.0	59.2	57.1	58.4	59.2	60.7	**66.9**

Table 7. Multi-scale vs. single scale. SPPnet **ZF** (similar to model **S**) results are from [11]. Larger networks with a single-scale offer the best speed / accuracy tradeoff. (**L** cannot use multi-scale in our implementation due to GPU memory constraints.)

多尺度只会提高mAP一点，但时间会变慢，所以单尺度更好

是否需要更多的训练数据？

扩大训练数据集，可以提高mAP 3到４点

ＳＶM比softmax好吗？

softmax比ＳＶM好+0.1 到+0.8 mAP

更多的候选域是不是更好?

物体候选域稀疏集：selective　search
物体候选域稠密集：ＤＰM
分类稀疏集候选域的一个方法是级联，先拒绝大量的候选域
使用候选域级联分类的方法可以提高fast-RCNN的精确度
候选域不是越多越好 稀疏的候选域更好