



专栏

目录视图

摘要视图

阅

个人资料



shenxiaolu1984

关注

发私信

访问：241192次

积分：2697

等级：5

排名：第12223名

【活动】2017 CSDN博客专栏评选 【评论送书】SQL优化、深度学习、数据科学家 CSDN日报20170526 ——《论程序员的时代焦虑与焦虑的缓解》 CSDN日报 | 4.19-5.19 上榜作者排行出炉

【目标检测】Fast RCNN算法详解

标签：目标检测 detection deep-learn 深度学习

2016-04-12 13:05

27383人阅读

评论(7)

收藏

举报

分类：论文解读 (20)

关闭

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE International Conference on Computer Vision. 2015.

继2014年的RCNN之后，Ross Girshick在15年推出Fast RCNN，构思精巧，流程更为紧凑，大幅提升了目标检测





百度云

云计算新用户
注册送520元代金券

立即领取

DL框架 (13)

会议概览 (7)

数学 (9)

文章存档

2017年05月 (2)

2017年04月 (1)

2017年02月 (4)

2016年12月 (1)

2016年11月 (8)

展开

阅读排行

【目标检测】Faster RCNN算法.. (32335)

【目标检测】Fast RCNN算法... (27383)

【深度学习】生成对抗网络Ge... (18478)

【目标检测】RCNN算法详解 (17645)

【目标跟踪】KCF高速跟踪详解 (17193)

循环矩阵傅里叶对角化 (9706)

【人体姿态】Convolutional Po... (7644)

【推荐系统算法】PMF(Probab... (7369)

欧拉角、四元数和旋转矩阵 (6239)

同样使用最大规模的网络，Fast RCNN和RCNN相比，训练时间从84小时减少为9.5小时，**测试时间从47秒减少为0.32秒**。在PASCAL VOC 2007上的准确率相差无几，约在66%-67%之间。

思想

基础：RCNN

简单来说，RCNN使用以下四步实现目标检测：

- a. 在图像中确定约1000-2000个**候选框**
- b. 对于每个候选框内图像块，使用**深度网络**提取**特征**
- c. 对候选框中提取出的特征，使用**分类器**判别是否属于一个特定类
- d. 对于属于某一特征的候选框，用**回归器**进一步调整其**位置**

更多细节可以参看[这篇博客](#)。

改进：Fast RCNN

Fast RCNN方法解决了RCNN方法三个问题：

问题一：测试时速度慢

RCNN一张图像内候选框之间大量重叠，提取特征操作冗余。

本文将**整张图像**归一化后**直接送入深度网络**。在邻接时，才加入候选框信息，在末尾的少数几层处理每个候选

关闭

问题二：训练时速度慢

原因同上。

在训练时，本文先将一张图像送入网络，**紧接着送入**从这幅图像上提取出的**候选区域**。这些候选区域的前几层

问题三：训练所需空间大

RCNN中独立的分类器和回归器需要大量特征作为训练样本。



百度云

云计算新用户
注册送520元代金券

立即领取

本文把类别判断和位置精调统一用深度网络实现，不再需要额外存储。

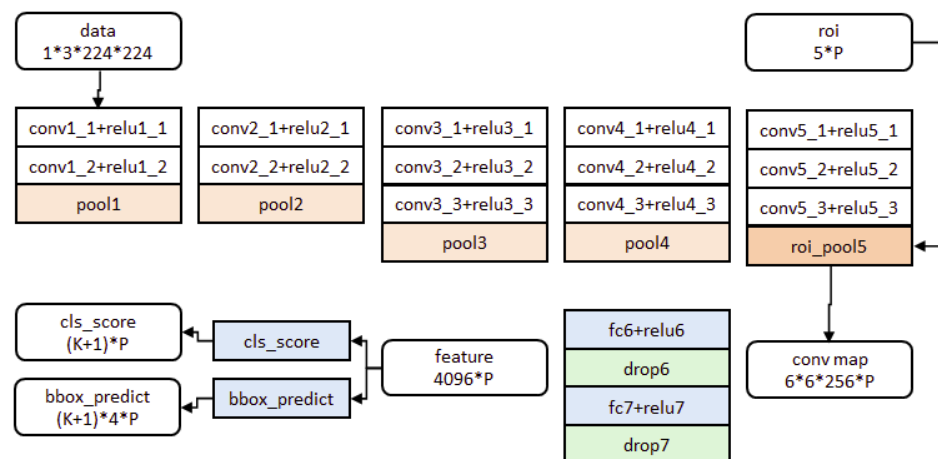
以下按次序介绍三个问题对应的解决方法。

特征提取网络

基本结构

图像归一化为 224×224 直接送入网络。

前五阶段是基础的conv+relu+pooling形式，在第五阶段结尾，输入P个候选区域（图像序号 $\times 1$ +几何位置 $\times 4$ ，序号用于训练）？。



注：文中给出了大中小三种网络，此处示出最大的一种。三种网络基本结构相似，仅conv+relu层数有差别。

roi_pool层的测试(forward)

百度云
云计算新用户
注册送520元代金券

立即领取

【目标检测】Fast RCNN算法... (7)

【深度学习】生成对抗网络Ge... (7)

【增强学习】Recurrent Visual... (7)

人脸关键点定位.Face Alignmen... (6)

推荐文章

- * 5月书讯：流畅的Python，终于等到你！
- * 【新收录】CSDN日报 —— Kotlin 专场
- * Android中带你开发一款自动爆破签名校验工具kstools
- * Android图片加载框架最全解析——深入探究Glide的缓存机制
- * Android 热修复 Tinker Gradle Plugin解析
- * Unity Shader-死亡溶解效果

最新评论

【深度学习】卷积层提速Factorized Convo...
shenxiaolu1984 : @yaogan5984:Tensorflow的f.nn.separable_conv2d()就实现...

【深度学习】卷积层提速Factorized Convo...
yaogan5984 : 博主，请问下，这个有没有开源的prototxt啊？

循环矩阵傅里叶对角化
kangqi5602 : 楼主，DFT离散Fourier变换矩阵前面为何要除以根号K。还有，这里进行离散fourier变换默认...

【TensorFlow动手玩】数据导入2
无奈的小心酸 : 受教了！感谢

【优化】梯度下降 收敛性 证明

百度云

云计算新用户
注册送520元代金券

立即领取



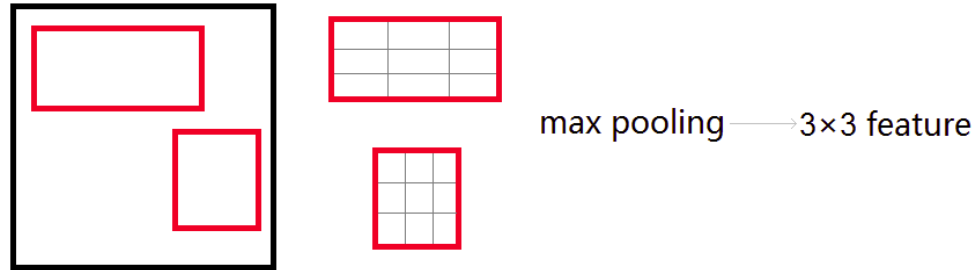
循环矩阵傅里叶对角化

baidu_38400801 : 赞同, 博主的二维推导根本就是错的, 我也验证过了。但是经过matlab验证, 一维推导的结果确实可以直接...

循环矩阵傅里叶对角化

baidu_38400801 : 赞同, 博主的二维推导根本就是错的, 我也验证过了。但是经过matlab验证, 一维推导的结果确实可以直接...

roi_pool层将每个候选区域均匀分成 $M \times N$ 块, 对每块进行max pooling。将特征图上大小不一的候选区域转变为大小统一的数据, 送入下一层。



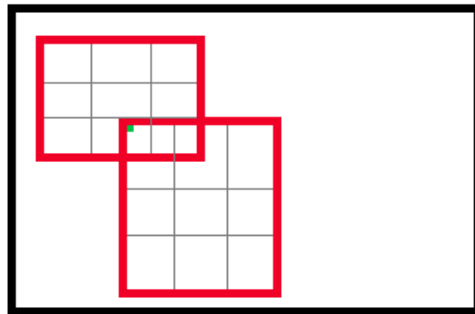
roi_pool层的训练(backward)

首先考虑普通max pooling层。设 x_i 为输入层的节点, y_j 为输出层的节点。

$$\frac{\partial L}{\partial x_i} = \begin{cases} 0 & \delta(i, j) = false \\ \frac{\partial L}{\partial y_j} & \delta(i, j) = true \end{cases}$$

其中判决函数 $\delta(i, j)$ 表示 i 节点是否被 j 节点选为最大值输出。不被选中有两种可能: x_i 不在 y_j 范围内, 或者 x_i 不是最大值。

对于roi max pooling, 一个输入节点可能和多个输出节点相连。设 x_i 为输入层的节点, y_{rj} 为第 r 个候选区域的第 j 个输出节点。



$$\frac{\partial L}{\partial x_i} = \sum_{r,j} \delta(i, r, j) \frac{\partial L}{\partial y_{rj}}$$

判决函数 $\delta(i, r, j)$ 表示 i 节点是否被候选区域 r 的第 j 个节点选为最大值输出。代价对于 x_i 的梯度等于所有相关的后

网络参数训练

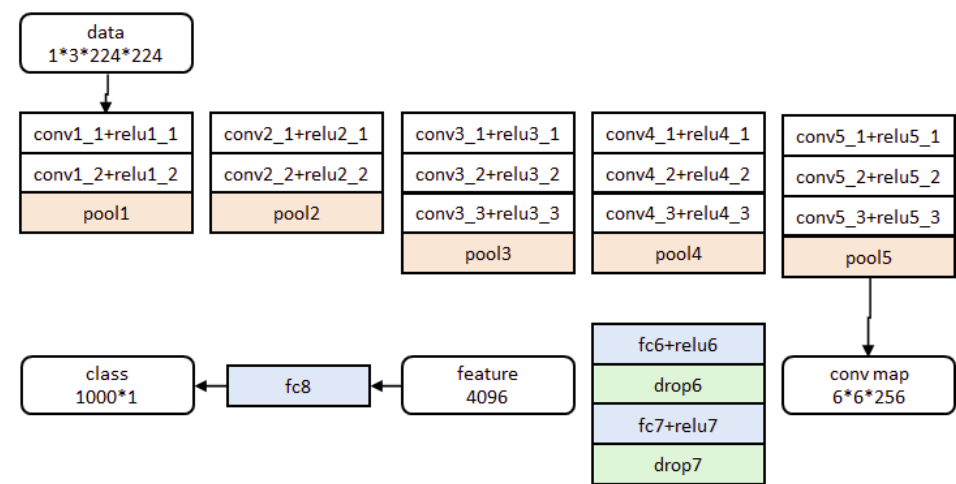
关闭





参数初始化

网络除去末尾部分如下图，在ImageNet上训练1000分类器。结果参数作为相应层的初始化参数。



其余参数随机初始化。

分层数据

在调优训练时，每一个mini-batch中首先加入N张完整图片，而后加入从N张图片中选取的R个候选框。这R个候选框可以复用N张图片前5个阶段的网络特征。

实际选择N=2，R=128。

训练数据构成

N张完整图片以50%概率水平翻转。

R个候选框的构成方式如下：

类别	比例	方式
前景	25%	与某个真值重叠在[0.5,1]的候选框



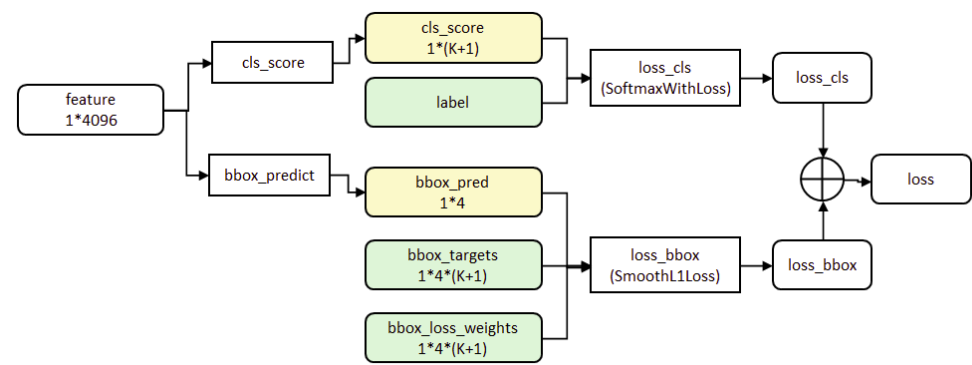


类别	比例	方式
背景	75%	与真值重叠的最大值在[0.1,0.5)的候选框

分类与位置调整

数据结构

第五阶段的特征输入到两个并行的全连层中（称为multi-task）。



cls_score层用于分类，输出K+1维数组 p ，表示属于K类和背景的概率。

bbox_prdict层用于调整候选区域位置，输出4*K维数组 d ，表示分别属于K类时，应该平移缩放参数。

代价函数

loss_cls层评估分类代价。由真实分类 u 对应的概率决定：

$$L_{cls} = -\log p_u$$

loss_bbox评估检测框定位代价。比较真实分类对应的预测参数 t^u 和真实平移缩放参数为 v 的差别：

$$L_{loc} = \sum_{i=1}^4 g(t_i^u - v_i)$$

关闭





g 为Smooth L1误差，对outlier不敏感：

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

总代价为两者加权和，如果分类为背景则不考虑定位代价：

$$L = \begin{cases} L_{cls} + \lambda L_{loc} & u \text{ 为前景} \\ L_{cls} & u \text{ 为背景} \end{cases}$$

源码中bbox_loss_weights用于标记每一个bbox是否属于某一个类

全连接层提速

分类和位置调整都是通过全连接层(fc)实现的，设前一级数据为 x 后一级为 y ，全连接层参数为 W ，尺寸 $u \times v$ 。一次前向传播(forward)即为。

$$y = Wx$$

计算复杂度为 $u \times v$

将 W 进行SVD分解，并用前 t 个特征值近似：

$$W = U\Sigma V^T \approx U(:, 1:t) \cdot \Sigma(1:t, 1:t) \cdot V(:, 1:t)^T$$

原来的前向传播分解成两步：

$$y = Wx = U \cdot (\Sigma \cdot V^T) \cdot x = U \cdot z$$

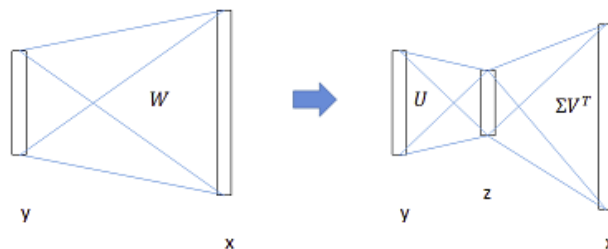
计算复杂度变为 $u \times t + v \times t$

关闭





在实现时，相当于把一个全连接层拆分成两个，中间以一个低维数据相连。



在github的源码中，这部分似乎没有实现。

实验与结论

实验过程不再详述，只记录结论

- 网络末端**同步训练**的分类和位置调整，提升准确度
- 使用**多尺度**的图像金字塔，性能几乎没有提高
- **倍增训练数据**，能够有2%-3%的准确度提升
- 网络直接输出各类概率(**softmax**)，比SVM分类器性能略好
- **更多候选窗**不能提升性能

同年作者团队又推出了Faster RCNN，进一步把检测速度提高到准实时，可以参看[这篇博客](#)。

关于RCNN, Fast RCNN, Faster RCNN这一系列目标检测**算法**，可以进一步参考作者在15年ICCV上的讲座[Traini](#)

关闭

顶
16

踩
1

• [上一篇](#) [【目标检测】RCNN算法详解](#)



- [下一篇](#) [【目标检测】Faster RCNN算法详解](#)

相关文章推荐

- [【目标检测】Fast RCNN算法详解](#)
- [【目标检测】Fast RCNN算法详解](#)
- [【目标检测】Fast RCNN算法详解](#)
- [GTX1080+Cuda8.0+Cudnnv5+caffe+faster-rcnn](#)
- [GTX1080+Cuda8.0+Cudnnv5+caffe+faster-rcnn](#)
- [【目标检测】Object Detection Fast RCNN 算法解析](#)
- [【目标检测】RCNN算法详解](#)
- [【目标检测】Faster RCNN算法详解](#)
- [【目标检测】Faster RCNN算法详解](#)
- [【目标检测】Faster RCNN算法详解](#)



山水文园二手



澳洲房价



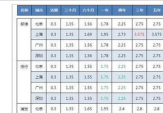
女找男征婚



美国房价



ui设计师



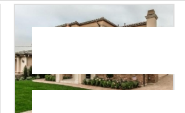
存款利率



app开发报价单



工作流



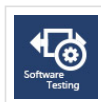
...

我的更多文章

[【目标检测】RCNN算法详解 \(2016-04-05 23:10:36\)](#)[【目标检测】Faster RCNN算法详解 \(2016-04-21 15:08:06\)](#)

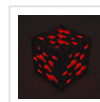
关闭

参考知识库



软件测试知识库

4775 关注 | 318 收录



算法与数据结构知识库

16473 关注 | 2320 收录

猜你在找

[使用决策树算法对测试数据进行分..](#)[C语言系列之 递归算法示例与 Win...](#)



百度云

云计算新用户
注册送520元代金券

立即领取

使用决策树算法对测试数据进行分..

C语言系列之 数组与算法实战

C++ 单元测试（GoogleTest）

《C语言/C++学习指南》加密解密...

C语言系列之 字符串压缩算法与结..

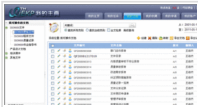
C语言系列之 字符串相关算法

C语言系列之 快速排序与全排列算..

数据结构基础系列(1)：数据结构...

59049	唐城路2号附3号	15-11
59386	唐城路2号附4号	10-12
59224	唐城路2号附3号	30-6
59365	唐城路2号附3号	33-11
59812	唐城路2号附5号	13-6
59566	唐城路2号附4号	25-12
59487	唐城路2号附4号	19-5
59343	唐城路2号附4号	7-5
59356	唐城路2号附4号	8-6

公租房摇号



文档管理系统

银行	期限	利率	币种	利率	币种	利率
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35
央行	一年	4.35	人民币	4.35	人民币	4.35

存款利率



程序员怎么样



推荐系统



大数据学习路线



一年拿本科文凭

查看评论



Cyiano

6楼 2017-04-12 17:25发表

看完了感觉很有用。对着这个看论文很快就理解了~



rimless

5楼 2017-01-19 02:57发表

您好，在“第五阶段的特征输入到两个并行的全连层中”下面的那个图中，bbox_predict的大小是不是应该是1*4*(K+1)？



ChenJoya

4楼 2017-01-06 21:11发表

楼主您好！博客写的非常棒！有一个地方不明白：在模型测试时，是不是先让网络输出bbox的四个变换，然后在conv5的特征图上得到区域，再用这个区域pooling一次得到feature来分类呢？因为一开始给定的区域不一定是真实区域，可能会有偏移，是不是要精细化之后再提取一次呢？谢谢！



sinat_34231517

楼主你好，我想问一下，bbox_targets，bbox_loss_weights，代表的是什么



h_jlwg6688

谢谢楼主的回复，再请教下，除了rcnn系列，还有什么方法可以用来做bounding box regression吗？

h_jlwg6688

关闭



百度云

云计算新用户
注册送520元代金券

立即领取



楼主你好，我想请问下 "loss_bbox评估检测框定位代价。比较真实分类对应的预测参数 t_u 和真实平移缩放参数为 v 的差别：“这里的 t 和 v 分别是什么？



shenxiaolu1984

Re: 2016-07-22 15:34发表

回复h_jlwg6688：平移缩放参数 - 从候选框到检测框的变换参数，包括xy方向的平移+xy方向的缩放，共四个， $i=1,2,3,4$ 。

t 和 v 分别是该参数的“预测结果”和“真实值”。预测结果 t 带有上标 u ，是因为在检测问题中，可能预测出不同类别的结果， u 是类别的序号。

例如，预测此处是一只猫，变换参数为 t_1 ，预测此处是一条狗，变换参数为 t_2 。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack VPN Spark ERP IE10
Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery BI HTML5 Spring Apache .NET API HTML
SDK IIS Fedora XML LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringS
aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr Angular Cloud Fou
Bootstrap

关闭

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司
京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

