目录

[一 RCNN算法](#_Toc12605282)

[1 引言 3](#_Toc12605283)

[2 模块设计 4](#_Toc12605284)

[2.1 区域推荐（region proposal） 4](#_Toc12605285)

[2.2 特征提取（Feature extraction） 5](#_Toc12605286)

[2.3 测试阶段的分类检测 7](#_Toc12605287)

[3 训练 7](#_Toc12605288)

[3.1 有监督的预训练 7](#_Toc12605289)

[3.2 特定领域的参数优化 7](#_Toc12605290)

[3.3 目标种类分类器 8](#_Toc12605291)

[4 Bounding-box回归 8](#_Toc12605292)

[5 在PASCAL VOC 2010上的结果 11](#_Toc12605293)

[6 算法存在的不足 11](#_Toc12605294)

[二 Fast-RCNN算法](#_Toc12605295)

[1 引言 13](#_Toc12605296)

[2 特征提取网络 14](#_Toc12605297)

[2.1 网络结构 14](#_Toc12605298)

[2.2 roi\_pool层详解 15](#_Toc12605299)

[3 分类与位置调整 16](#_Toc12605300)

[3.1 数据结构 16](#_Toc12605301)

[3.2 代价函数 16](#_Toc12605302)

[4 网络参数训练 17](#_Toc12605303)

[4.1 有监督预训练 17](#_Toc12605304)

[4.2 特定样本下的微调 18](#_Toc12605305)

[4.3 单一尺度(brute-force)和多尺度(image pyramids) 18](#_Toc12605306)

[5 全连接层SVD提速 19](#_Toc12605307)

[6 算法存在的不足 19](#_Toc12605308)

[三 Faster RCNN](#_Toc12605309)算法

[1 引言 21](#_Toc12605310)

[2 Faster R-CNN整体介绍 22](#_Toc12605311)

[3 Region Proposal Network 22](#_Toc12605312)

[3.1 Anchors 23](#_Toc12605313)

[3.2 RPN的损失函数 24](#_Toc12605314)

[3.3非极大值抑制(Non-maximum suppression)处理 24](#_Toc12605315)

[4 RPN网络与Fast R-CNN网络的权值共享 25](#_Toc12605316)

# 一 RCNN算法

# 1 引言

RCNN（Regions with CNN features）是将CNN方法应用到目标检测问题上的一个里程碑，借助CNN良好的特征提取和分类性能，通过RegionProposal方法实现目标检测问题的转化。

关于目标检测，其实是一个比较好理解的概念，具体来说，目标检测的任务是找出图像中所有感兴趣的目标，确定这些目标对象的语义类别，并标定出目标对象在图像中的位置，目标对象的位置一般使用边界框进行标定。在图1-1中，对目标检测进行了说明。

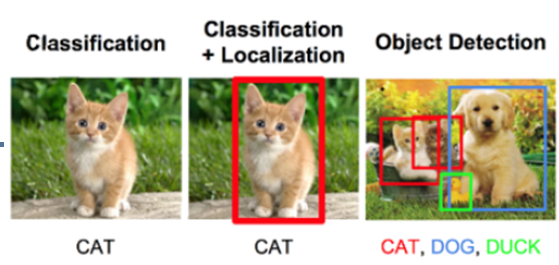


图1-1 目标检测任务

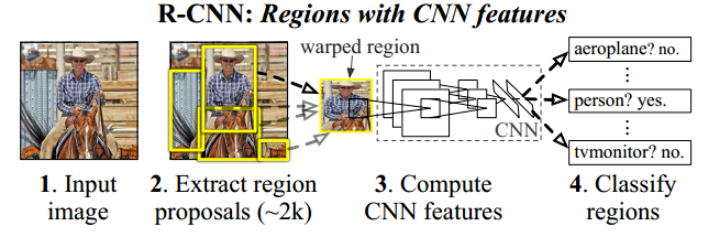
在RCNN之前，overfeat已经是用深度学习的方法做目标检测，但RCNN是第一个可以真正可以工业级应用的解决方案。可以说改变了目标检测领域的主要研究思路，紧随其后的系列文章：*Fast-RCNN* ，*Faster-RCNN*都沿袭了*R-CNN*的思路。

RCNN结合了两个关键的因素：

（1）在候选区域上自下而上使用大型卷积神经网络(CNN)，用以定位和分割物体。

（2）当带标签的训练数据不足时，先针对辅助任务进行有监督预训练，再进行特定任务的调优，就可以产生明显的性能提升。

与图像分类不同的是目标检测需要定位一个图像内的许多物体。一个方法是将框定位看做是回归问题。另一个可替代的方法是使用滑动窗口探测器。这两种方法均有其弊端，RCNN中通过产生region proposal的方法，解决了CNN的定位问题。对这每张图片，产生了接近2000个与类别无关的region proposal，对每个CNN抽取了一个固定长度的特征向量，然后借助专门针对特定类别数据的线性SVM对每个区域进行分类。我们不考虑region的大小，使用图像变形的方法来对每个不同形状的region proposal进行归一化，得到大小相同的region作为CNN的输入。图1-2展示了RCNN的全貌。因为该方法把region proposal和CNNs结合起来，所以该方法被称为R-CNN：Regions with CNN features。

 图1-2 CNN结构图

# 2 模块设计

RCNN的目标检测系统有三个模块构成。第一个，产生与类别无关的region proposal，这些推荐定义了一个候选检测区域的集合；第二个是一个大型卷积神经网络，用于从每个区域抽取特定大小的特征向量；第三个是一个指定类别的线性SVM。本部分，将展示每个模块的设计，并介绍他们的测试阶段的用法，以及参数是如何学习的细节。

## 2.1 区域推荐（region proposal）

产生类别无关区域推荐的方法有很多种，比如: objectness（物体性），selective search（选择性搜索），category-independent object proposals(类别无关物体推荐)，constrained parametric min-cuts（受限参最小剪切, CPMC)，multi-scal combinatorial grouping(多尺度联合分组)等方法。由于R-CNN对特定区域算法是不关心的，所以RCNN采用了selective search（选择性搜索）来产生region proposal。对每张图片，产生接近2000个与类别无关的region proposal。

## 2.2 特征提取（Feature extraction）

利用大型卷积神经网络(CNNs)来对候选区域进行特征提取，CNN中采用了AlexNet网络，如图2-1所示。

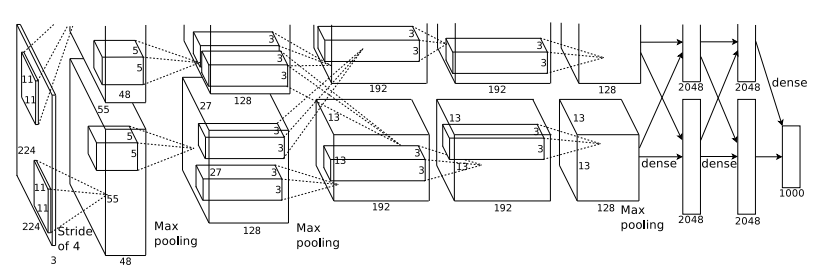


图2-1 Alexnet网络

Alexnet中共有五个卷积层和三个全连接层，每个卷积层后进行重叠的最大池化，并且为了加速计算，从从Conv2开始至Conv5, 是将像素层平均分开送至两个GPU中运算。

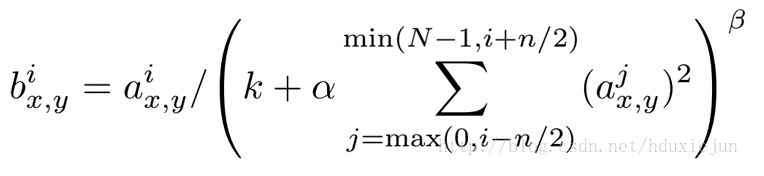
RCNN利用Alexnet网络，对每个输入的大小为277\*277的region proposal，通过五个卷积层和两个全连接层进行前向传播,得到一个4096维的特征向量。

由于架构中的CNN只能接受固定大小的输入，设置为277\*277，为了计算region proposal的特征，我们首先要对图像进行转换，使得它符合CNN的输入。这个变换有很多办法，RCNN使用了最简单的一种。无论候选区域是什么尺寸和宽高比，都把候选框变形成想要的尺寸。具体的，在变形之前先在候选框周围加上16的padding,再进行各向异性缩放。

关于Alexnet网络，有两点创新值得关注一下，一个是提出了LRN技术（局部响应归一化），一个是采用了dropout技术。

（1）dropout技术

通过LRN技术，对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大，并抑制其他反馈较小的神经元，增强了模型的泛化能力。具体公式如下：

****  2-1

第i个核在位置（x,y)运用ReLU非线性化的神经元输出，可以理解成在某一张图片中的某一个通道下的某个高度和某个宽度位置的点。n是同一位置上临近的kernel map的数目，N是kernel的总数。

n/2, ,k都是自定义的，k=2,n=5,a=e-4,β=0.75，由验证机决定



（2）dropout技术

dropout技术的核心思想是在训练时以一定的概率保留一部分神经元参与工作，即在训练过程中，每个神经元都以概率 p 保持其是激活状态，而以 1-p 的概率来临时丢弃某些神经元节点不予参与工作，但在下次工作时，又可能被激活

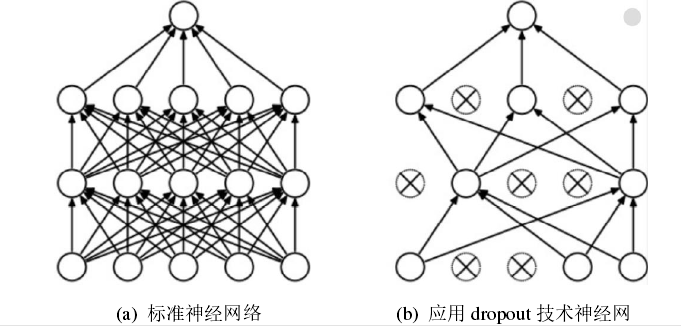
****

图2-2 dropout技术

dropout技术的优点：

1）对于每次输入到神经网络中的样本，由于其 dropout 技术的随机性，每次产生的网络结构都会有所不同，但这些结构却同时共享节点的权值，这使得不同样本对应不同的模型，故能够使网络模型学到多种相互独立的内部表征以防止过拟合现象。

2）由于一个神经元不再依赖于其它有固定关系的隐含节点共同作用而存在，所以这种技术阻止了某些特征仅仅在其它特定特征下才有效果的情况，使神经元学习得到更鲁棒的特征来提高网络的泛化能力。

目前，关于卷积神经网络模型的先进研究大都采用 ReLU+dropout 技术

## 2.3 测试阶段的分类检测

测试阶段，在测试图像上使用selective search抽取2000个推荐区域。然后变形每一个推荐区域，再通过CNN前向传播计算出特征。接着我们使用对每个类别训练出的SVM给整个特征向量中的每个类别单独打分。给出一张图像中所有的打分区域，然后使用NMS（每个类别是独立进行的），拒绝掉一些和高分区域的IOU大于阈值的候选框。

两个特性让检测变得很高效。首先，所有的CNN参数都是跨类别共享的。其次，通过CNN计算的特征向量相比其他通用方法维度是很低的。这种共享的结果就是计算推荐区域特征的耗时可以分摊到所有类别的头上（GPU：每张图13s，CPU：每张图53s）。唯一的和具体类别有关的计算是特征向量和SVM权重和点积，以及NMS。实践中，所有的点积都可以批量化成一个单独矩阵间运算。特征矩阵的典型大小是2000×4096，SVM权重的矩阵是4096xN，其中N是类别的数量。分析表明R-CNN可以扩展到上千个类别，即使有10万个类别，矩阵乘法在现代多核CPU上只需要10s而已

# 3 训练

## 3.1 有监督的预训练

RCNN在大型辅助训练集ILSVRC2012分类数据集（没有约束框数据）上预训练了CNN。预训练采用了Caffe的CNN库。总体来说，训练的CNN效果十分接近krizhevsky等人的网络的性能，在ILSVRC2012分类验证集在top-1错误率上比他们高2.2%。差异主要来自于训练过程的简化。

## 3.2 特定领域的参数优化

为了让我们的CNN适应检测任务和变形后的推荐窗口。我们只使用变形后的推荐区域对CNN参数进行SGD训练。我们替换掉了ImageNet专用的1000-way分类层，换成了一个随机初始化的21-way分类层，（其中20是VOC的类别数，1代表背景），而卷积部分都没有改变。我们对待所有的推荐区域，如果其和真实标注的框的IoU>= 0.5就认为是正例，否则就是负例。SGD开始的learning\_rate为0.001（是初始化预训练时的十分之一），这使得调优得以有效进行而不会破坏初始化的成果。每轮SGD迭代，我们统一使用32个正例窗口（跨所有类别）和96个背景窗口，即每个mini-batch的大小是128。另外我们倾向于采样正例窗口，因为和背景相比他们很稀少。

## 3.3 目标种类分类器

思考一下检测汽车的二分类器。很显然，一个图像区域紧紧包裹着一辆汽车应该就是正例。同样的，没有汽车的就是背景区域，也就是负例。较为不明确的是怎样标注哪些只和汽车部分重叠的区域。我们使用IoU重叠阈值来解决这个问题，低于这个阈值的就是负例。这个阈值我们选择了0.3，是在验证集上基于{0, 0.1, … 0.5}通过网格搜索得到的。认真选择这个阈值很重要，如果设置为0.5，可以降低mAP5个点，设置为0，就会降低4个点。正例就严格的是标注的框。

一旦特征提取出来，并应用标签数据，我们优化了每个类的线性SVM。

在fine-tunning和SVM训练这两个阶段，RCNN定义得正负样例是不同的。因为fine-tunning阶段是由于CNN对小样本容易过拟合，需要大量训练数据，故对IoU限制宽松： IoU>0.5的建议框为正样本，否则为负样本； SVM这种机制是由于其适用于小样本训练，故对样本IoU限制严格：Ground Truth为正样本，与Ground Truth相交IoU＜0.3的建议框为负样本。

# 4 Bounding-box回归

基于错误分析，使用了一种简单的方法减小定位误差，训练了一个线性回归模型在给定一个选择区域的pool5特征时去预测一个新的检测窗口，利用这种方法，修复了大量的错位检测。

如图4-1红色框和绿色框所示，红色框是我们生成的Region Proposal记为R，而绿色是我们的实际要达到的检测框（Ground Truth）记为G。那么IOU就是(R∩G)/(R∪G) 。定义这个的目的就是为了在训练时为已标定的Bounding-Box寻找一个能够扩张为该Bounding-Box的Region Proposal，当IOU小于某个值得时候就必须被丢弃，这样的具体原因是因为能够进行Bounding-Box regression的两个框的是差异比较小的，这样才能进行线性模拟。

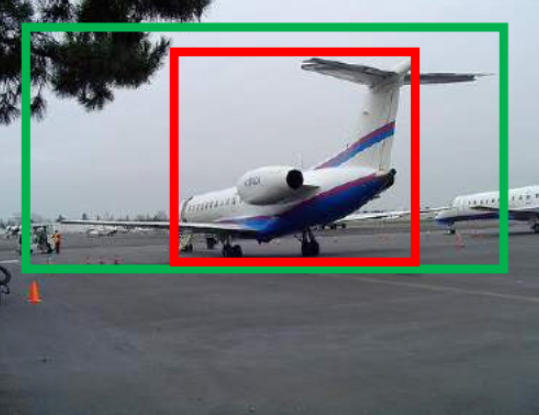


图4-1 Region Proposal和Ground Truth

对于窗口一般使用思维向量（x,y,w,h）来表示，分别表示窗口的中心点坐标和宽高。在图4-2中，对于图，红色框P代表原始的proposal，绿色的框G代表目标Ground Truth,我们的目标是寻找一种关系使得输入原始的窗口P经过映射得到一个跟真实窗口G更接近的回归窗口。

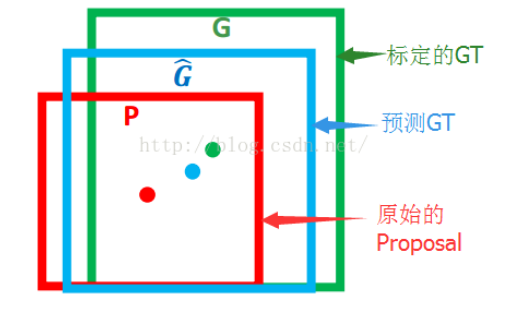


图4-2 Bounding-box回归

即：给定，寻找一种映射，使得：

 4-1

要从P得到，需要经过平移和缩放。

平移我们可以表示为：

 4-2

 4-3

缩放可以表示为：

 4-4

 4-5

观察得到，Bounding-box回归学习就是这四个变化，下一步就是设法得到这四个映射。线性回归就是给定输入的特征向量 X, 学习一组参数 W, 使得经过线性回归后的值跟真实值 Y(Ground Truth)非常接近. 即。 那么 Bounding-box 中我们的输入以及输出分别是什么呢？

Input: 这个窗口对应的 CNN 特征，也就是 R-CNN 中的 Pool5 feature（特征向量），记做；

Ouput: 输出的是需要进行的平移变换和尺度缩放，虽然我们是最终想要得到Ground Trude，但是有了这四个变换我们就可以通过转换得到Ground Truth。所以，这四个值应该是经过 Ground Truth 和 Proposal 计算得到的真正需要的平移量和尺度缩放 。

我们要让预测值跟真实值差距最小，用\*表示 x,y,w,h， 也就是每一个变换对应一个目标函数，得到损失函数为：

 4-6

其中：

 4-7

函数优化后为：

 4-8

最后利用梯度下降法或者最小二乘法就可以得到， 所以根据这Loss，输入是pool5的feature map的特征向量， 输出是近似于真实的t∗=(tx,ty,tw,th)的值所对应的。所以应用时需要再转换到(Gx^,Gy^,Gw^,Gh^)。

# 5 在PASCAL VOC 2010上的结果

按照PASCAL VOC的最佳实践步骤，在VOC2007的数据集上验证了所有的设计思想和参数处理，我们在VOC2012上训练和优化了SVMs，表1展示了在VOC2010的结果，将RCNN方法同四种先进基准方法作对比，其中包括SegDPM，UVA等实验结果表明，RCNN方法的mAP达到一个更大的提升，从35.1%提升至53.7%，而且速度更快。

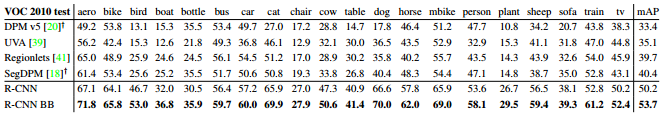


图5-1 实验结果

# 6 算法存在的不足

RCNN算法虽然在目标检测领域做出了巨大贡献，但仍然存在一些不足，例如：

（1）多个候选区域对应的图像需要预先提取，占用较大的磁盘空间；

（2）针对传统CNN需要固定尺寸的输入图像，crop/warp（归一化）产生物体截断或拉伸，会导致输入CNN的信息丢失；

（3）每一个ProposalRegion都需要进入CNN网络计算，上千个Region存在大量的范围重叠，重复的特征提取带来巨大的计算浪费。

针对这些不足，在RCNN的基础，后来又有学者相继提出了Fast RCNN以及Fast RCNN等算法，减少了训练和测试时间，并提高了检测精度，未来将继续学习这些算法，努力提升自己。

# 二 Fast-RCNN算法

# 1 引言

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的速度。Fast-RCNN具有以下创新点：

（1）规避R-CNN中冗余的特征提取操作，只对整张图像全区域进行一次特征提取；

（2）用 RoI pooling 层取代最后一层 max pooling层，同时引入候选框信息，提取相应候选框特征；

（3）Fast R-CNN 网络末尾采用并行的不同的全连接层，可同时输出分类结果和窗口回归结果，实现end-to-end的多任务训练【候选提取除外】，也不需要额外的特征存储空间【R-CNN中这部分特征是供SVM和Bounding-box regression进行训练的】；

（4）采用SVD对Fast R-CNN网络末尾并行的全连接层进行分解，减少计算复杂度，加快检测速度。

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。图1-1展示了Fast-RCNN的全貌。

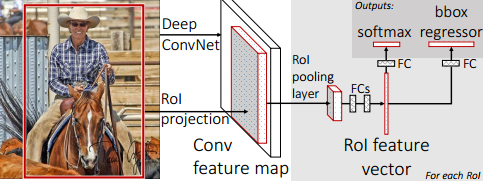


图1-1 Fast-RCNN的全貌

# 2 特征提取网络

## 2.1 网络结构

文中采用AlexNet、VGG\_CNN\_M\_1024、VGG-16这三种网络分别进行训练测试，下面仅以VGG-16举例。如图2-1所示。

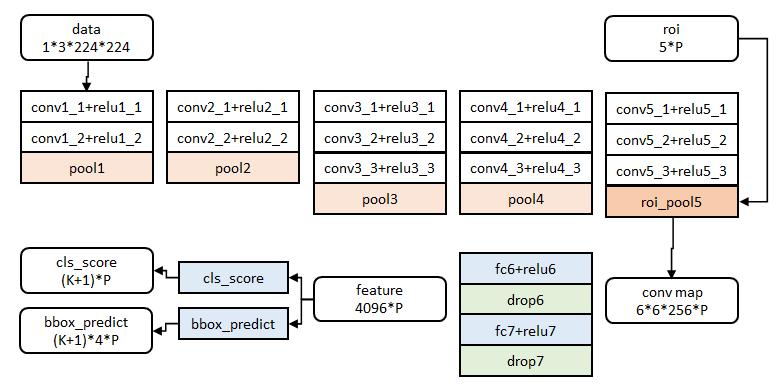


图2-1 VGG-16下的Fast-RCNN网络

整个测试过程为什么可以只进行一次CNN特征提取操作？

先看R-CNN网络，它首先采用selective search算法提取约2k个候选框，并对所有候选框都进行了CNN特征提取操作，会出现重叠区域多次重复提取特征，这些操作非常耗时、耗空间。事实上我们并不需要对每个候选框都进行CNN特征提取操作，只需要对原始的整张图片进行1次CNN特征提取操作即可，因为selective search算法提取的候选框属于整张图片，因此对整张图片提取出特征图后，再找出相应候选框在特征图中对应的区域，这样就可以避免冗余的特征提取操作，节省大量时间。

为什么要将每个候选框对应的特征框池化到H×W 的size？如何实现？

像AlexNet CNN等网络在提取特征过程中对对输入图像的大小并无要求，只是在提取完特征进行全连接操作的时候才需要固定特征尺寸，利用这一点，Fast R-CNN可输入任意size图片，并在全连接操作前加入RoI池化层，将候选框对应特征图中的特征框池化到H×W 的size，以便满足后续操作对size的要求；

## 2.2 roi\_pool层详解

（1）roi\_pool层的测试（forward）

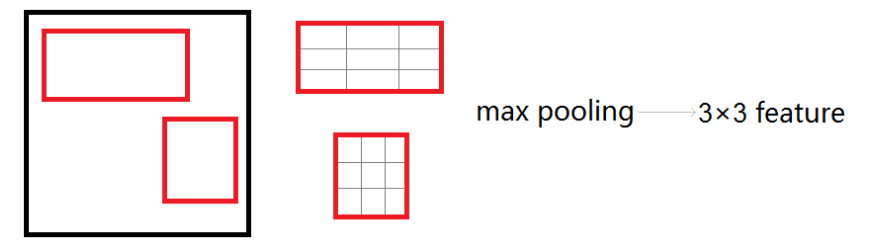


图2-2 roi\_pool层

roi\_pool层将每个候选区域对应的特征框均匀分成M×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。

（2）roi\_pool层的训练（backward）

首先考虑普通max pooling层。设为输入层的节点，为输出层的节点。

 2-1

其中判决函数表示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：不在范围内，或者不是最大值。

对于roi max pooling，一个输入节点可能和多个输出节点相连，如图2-3所示。设为输入层的节点，为第r个候选区域的第j个输出节点。

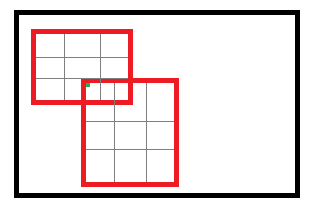


图2-3 roi\_pool层节点连接表示

梯度计算公式为：

 2-2

判决函数表示i节点是否被候选区域r的第j个节点选为最大值输出。对于的梯度等于所有相关的后一层梯度之和。

# 3 分类与位置调整

## 3.1 数据结构

经过VGG的fc6和fc7后得到的特征向量输入到两个并行的全连接层中（称为multi-task），如图3-1所示。黄色框表示训练数据，绿色框表示输入目标，源码中bbox\_loss\_weights用于标记每一个bbox是否属于某一个类。

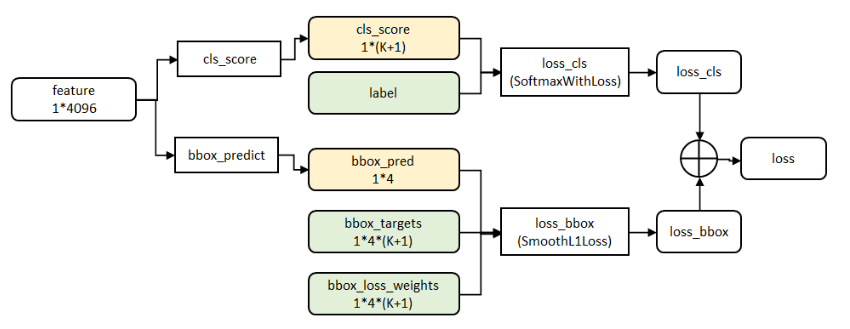


图3-1 分类与位置调整

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tt，表示分别属于K类时，应该平移缩放的参数。

## 3.2 代价函数

针对一个RoI即一类物体说明。

loss\_cls层评估分类代价。由真实分类u对应的概率决定：

 3-1

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

 3-2

g为Smooth L1误差，对outlier不敏感：

 3-3

总代价为两者加权和，如果分类为背景则不考虑定位代价：

 3-4

# 4 网络参数训练

## 4.1 有监督预训练

在ILSVRC 20XX上训练1000类分类器，训练得到的参数作为相映层的初始化参数。ILSVRC 20XX样本只有类别标签，有1000种物体。预训练网络结构如图4-1所示。

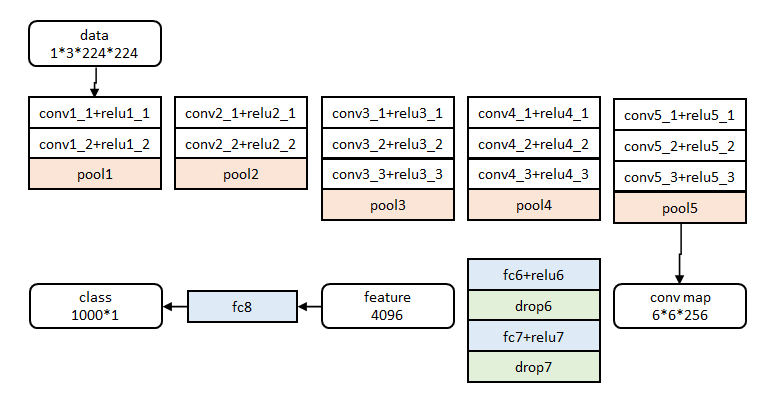


图4-1 有监督的预训练

## 4.2 特定样本下的微调

微调前，需要对有监督预训练后的模型进行2步转化：

1）RoI池化层取代有监督预训练后的VGG-16网络最后一层池化层；

2）两个并行层取代上述VGG-16网络的最后一层全连接层：并行层之一cls\_score层用于21个分类输出【20种类+背景】；并行层之二是bbox\_prdict层用于调整候选区域位置。

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。R个候选框分为正负样本，构成方式如表4-1所示。

表4-1 微调样本分布

|  |  |  |
| --- | --- | --- |
| 类别 | 比例 | 方式 |
| 正样本 | 25% | 与某类Ground Truth相交IoU∈[0.5,1]的候选框 |
| 负样本 | 75% | 与20类Ground Truth相交IoU中最大值∈[0.5,1]的候选框 |

## 4.3 单一尺度(brute-force)和多尺度(image pyramids)

对于如何处理尺度不变性问题，即如何使24×24和1080×720的车辆同时在一个训练好的网络中都能正确识别？ 文中提及两种方式处理：单一尺度(brute-force)和多尺度(image pyramids)。

单一尺度直接在训练和测试阶段将image定死为某种scale，直接输入网络训练就好，然后期望网络自己能够学习到scale-invariance的表达；多尺度在训练阶段随机从图像金字塔【缩放图片的scale得到，相当于扩充数据集】中采样训练，测试阶段将图像缩放为金字塔中最为相似的尺寸进行测试。

可以看出，多尺度应该比单一尺度效果好。Fast-RCNN作者对单一尺度和多尺度分别进行了实验，不管哪种方式下都定义图像短边像素为s，单一尺度下s=600【维持长宽比进行缩放】，长边限制为1000像素；多尺度s={480,576,688,864,1200}【维持长宽比进行缩放】，长边限制为2000像素，生成图像金字塔进行训练测试。实验结果表明AlexNet、VGG\_CNN\_M\_1024下单一尺度比多尺度mAP差1.2%~1.5%，但测试时间上却快不少，VGG-16下仅单一尺度就达到了66.9%的mAP，多尺度由于GPU显存限制无法实现，该实验证明了深度神经网络善于直接学习尺度不变形，对目标的scale不敏感。

# 5 全连接层SVD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，后一级数据为，全连接层参数为W，尺寸为。一次向前传播（forward）即为：

 5-1

计算复杂度为。

将W进行SVD分解，并用前t个特征值近似：

 5-2

原来的前向传播分解成两步：

 5-3

计算复杂度变为，若，则这种分解会大大减少计算量。

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，第一个全连接层不含偏置，第二个全连接层含偏置。实验表明，SVD分解全连接层能使mAP只下降0.3%的情况下提升30%的速度。

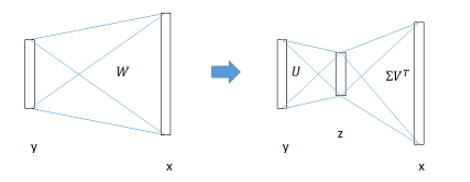


图5-1 全连接层SVD提速

# 6 算法存在的不足

Fast R-CNN中采用selective search算法提取候选区域，selective search算法候选区域提取需要2~3s，而提特征分类只需要0.32s，所以目标检测大多数时间都消耗在这里，这无法满足实时应用需求，而且因为候选区域是使用selective search算法先提取出来的，所以Fast R-CNN并没有实现真正意义上的端到端训练模式。

在Fast R-CNN之后，作者团队又推出了Faster RCNN，直接使用CNN产生候选区域并对其分类，进一步提高了检测速度和检测精度。

# 三 Faster RCNN算法

# 1 引言

R-CNN是 Faster R-CNN 的启发版本. R-CNN 是采用 [Selective Search](https://koen.me/research/selectivesearch/) 算法来提取(propose)可能的 RoIs(regions of interest) 区域，然后对每个提取区域采用标准 CNN 进行分类。出现于 2015 年早期的 [Fast R-CNN](https://arxiv.org/abs/1504.08083) 是 R-CNN 的改进，其采用兴趣区域池化(Region of Interest Pooling，RoI Pooling) 来共享计算量较大的部分，提高模型的效率.但还不是一个真正端到端的训练，在候选区域提取上耗费了大量时间。

[Faster R-CNN](http://arxiv.org/abs/1506.01497) 随后被提出，其是第一个完全可微分的模型. [Faster R-CNN](http://arxiv.org/abs/1506.01497) 是 R-CNN 论文的第三个版本。Faster R-CNN与Fast R-CNN最大的区别就是提出了一个叫RPN(Region Proposal Networks)的网络，专门用来推荐候选区域的，RPN可以理解为一种全卷积网络，该网络可以进行end-to-end的训练，最终目的是为了推荐候选区域。

为了将RPNs与Fast R-CNN目标检测网络相结合，[Faster R-CNN](http://arxiv.org/abs/1506.01497" \t "_blank) 提出了一种简单的训练方案，在保持方案不变的情况下，在区域建议任务的微调和目标检测的微调之间交替进行。该方案收敛速度快，生成了一个具有conv特性的统一网络，并在两个任务之间共享。Faster R-CNN的整体结构如图1所示：

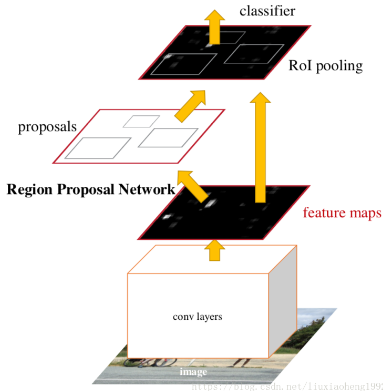


图1 Faster R-CNN整体结构

# 2 Faster R-CNN整体介绍

特征提取网络以VGG-16为例，图2展示了Faster R-CNN的全貌。

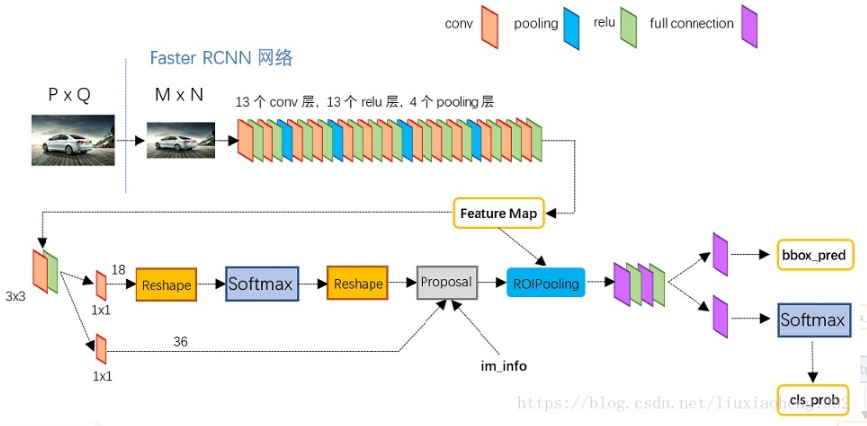


图2 Faster R-CNN全貌

通过上图可以看出Faster R-CNN由四个部分组成：

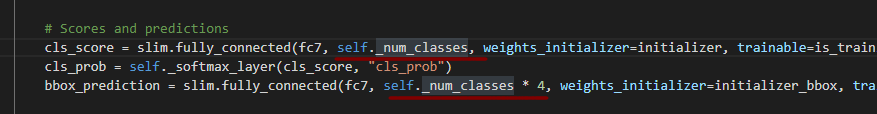
1）卷积层(conv layers)，用于提取图片的特征，输入为整张图片，输出为提取出的特征称为feature maps。

2）RPN网络(Region Proposal Network)，用于推荐候选区域，这个网络是用来代替之前的search selective的。输入为图片(因为这里RPN网络和Fast R-CNN共用同一个CNN，所以这里输入也可以认为是featrue maps)，输出为多个候选区域，这里的细节会在后面详细介绍。

3）RoI pooling，和Fast R-CNN一样，将不同大小的输入转换为固定长度的输出，输入输出和Faste R-CNN中RoI pooling一样。

4）分类和回归，这一层的输出是最终目的，输出候选区域所属的类，和候选区域在图像中的精确位置。

对于每一个候选框进行检测（类别得分，加回归预测）



# 3 Region Proposal Network

Faster R-CNN与Fast R-CNN最大的区别就是提出了一个叫RPN(Region Proposal Networks)的网络，专门用来推荐候选区域的，RPN可以理解为一种全卷积网络，该网络可以进行end-to-end的训练，最终目的是为了推荐候选区域。

原文中RPN网络为CNN后面接一个3\*3的卷积层，再接两个并行的1\*1的卷积层，其中一个是用来给softmax层进行分类，另一个用于给候选区域精确定位。

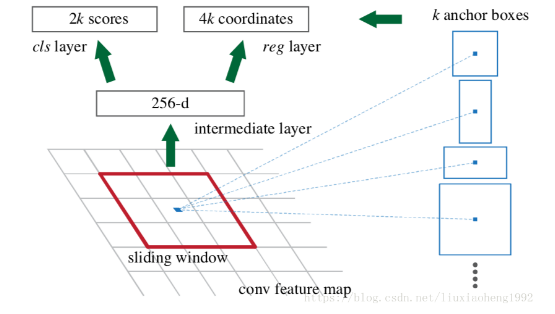


图3 RPN 结构

## 3.1 Anchors

anchors可以理解为一些预设大小的框，是通过利用不同的尺寸和比例在图片上放置得到的 boxes，并作为第一次预测 object 位置的参考 boxes。anchors的种类用k表示，表示在conv feature map上的每个点产生9个anchors，在原文中k=9，由3种面积(1282,2562,5122)和3种长宽比(1:1,1:2,2:1)组成，这里anchors的大小选取是根据检测时的图像定义，在检测时会将最小边缩放到600。假设CNN得到的feature map大小为，那总的anchors个数为*，*

虽然 anchors 是基于卷积特征图定义的，但最终的 anchos 是相对于原始图片的。由于只有卷积层和 pooling 层，特征图的维度是与原始图片的尺寸成比例关系的。假设原图大小为，由SPP-net文章详细解读知，，S为之前所有层的stride size相乘，所以feature map上的点乘以S即为anchors的原点位置，得到所有框的原点位置以及大小就可以得到原图上的anchors区域了。

RPN 利用所有的参考anchors，输出一系列 objecs 的良好的 proposals. 针对每个anchor，都有两个不同的输出：

1）anchor 内是某个 object 的概率

对于分类层，每个 anchor 输出两个预测值：anchor 是背景(background，非object)的 score 和 anchor 是前景(foreground，object) 的 score。  
 注：RPN 不关注于 anchor 是某个 object class，只是确定其可能是一个 object (而不是背景background). 即： RPN 不关心 object 类别，只确定是 object 还是 background。

2）anchor 边界框回归输出

对于回归层，也可以叫边界框调整层，而每个anchor都有[x, y, w, h]对应4个偏移量，所以每个anchor 输出 4 个预测值：用于调整 anchors 来更好的拟合预测的 object。

## 3.2 RPN的损失函数

RPN训练中对于正样本文章中给出两种定义。第一，与ground truth box有最大的IoU的anchors作为正样本；第二，与ground truth box的IoU大于0.7的作为正样本。文中采取的是第一种方式。文中定义的负样本为与ground truth box的IoU小于0.3的样本。

然后，随机采样 anchors 来生成batchsize=256 的 mini-batch，尽可能的保持 foreground 和 background anchors 的比例平衡。

训练RPN的loss函数定义如下：

 3-1

其中，i表示mini-batch中第i个anchor，pi表示第i个anchor是前景的概率，当第i个anchor是前景时为1反之为0，表示预测的bounding box的坐标，为ground truth的坐标。

这部分的loss函数和Fast R-CNN一样，除了正负样本的定义不一样，其他表示是一样的。

## 3.3[非极大值抑制(Non-maximum suppression)](http://blog.csdn.net/zziahgf/article/details/78525279)处理

由于 Anchors一般是有重叠的overlap，因此，相同 object 的 proposals 也存在重叠。为了减少冗余，采用 NMS 算法处理，丢弃与一个score 更高的 proposal 间 IoU 大于预设阈值的 proposals.。论文中将NMS的IoU阈值设置为0.7。NMS 处理后，根据 score 对 topN 个 proposals 排序， 使用排名前N的建议区域进行检测，论文中 N=2000。

虽然 NMS 看起来比较简单，但 IoU 阈值的预设需要谨慎处理。如果 IoU 值太小，可能丢失 objetcs 的一些 proposals；如果 IoU 值过大，可能会导致 objects 出现很多 proposals。

# 4 ****RPN网络与Fast R-CNN网络的权值共享****

RPN最终目的是得到候选区域，但在目标检测的最终目的是为了得到最终的物体的位置和相应的概率，这部分功能由Fast R-CNN做的。因为RPN和Fast R-CNN都会要求利用CNN网络提取特征，所以文章的做法是使RPN和Fast R-CNN共享同一个CNN部分。训练过程如下：

Step1：先独立训练RPN；

Step2：然后利用Step1得到的候选区域，训练Fast R-CNN；

（此时两个网络不共享conv层）

Step3：再使用得到的Fast R-CNN中的CNN去初始化RPN的CNN再次训练RPN，这里不更新RPN的CNN，仅更新RPN特有的层；

Step4：最后再次训练Fast R-CNN，这里不更新CNN，仅更新Fast R-CNN特有的层。

因此，两个网络共享相同的conv层，形成统一的网络。