layout: post
title: "RCNN系列文章之Fast RCNN详解"
date: 2020-07-11
description: "目标检测"

tag: 目标检测

RCNN系列的文章主要是RCNN，Fast RCNN， Faster RCNN， Mask RCNN, Cascade RCNN,这一系列的文章是目标检测two-stage算法的代表，这系列的算法精度高，效果好，是一类重要的方法。

论文地址：Fast R-CNN

简要介绍

RCNN与SPPnet一些缺点与不足：

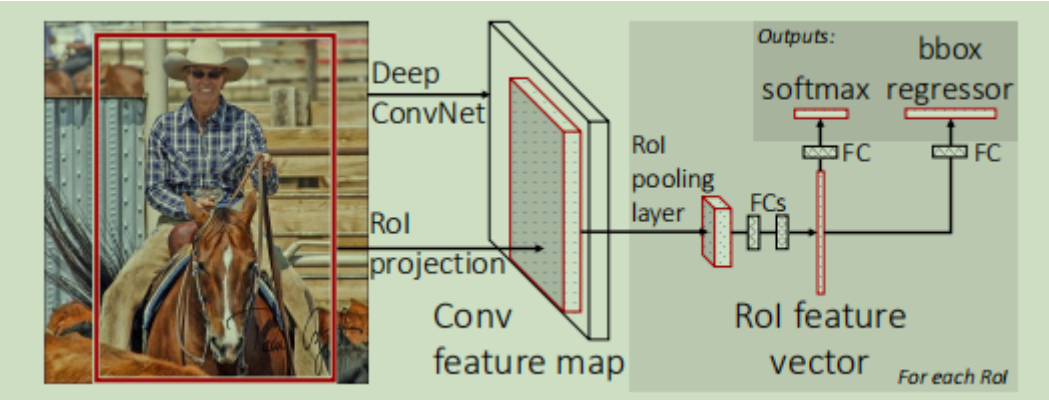
- **训练过程是一个multi-stage pipline**. RCNN首先在给定的region proposal上使用log损失进行微调。然后将卷积神经网络提取到的特征训练SVM分类器，利用SVM替代神经网络分类算法中常用的softmax。第三部分就是学习检测框的回归。
- **训练需要大量的空间与时间**， 由于训练过程中需要将卷积神经网络提取的特征写入磁盘，因此需要大量的物理存储空间，训练过程十分缓慢。
- **检测过程非常缓慢**，在测试时，从每个测试图像中的每个目标候选框提取特征。

RCNN主要对于每张图像的每个region proposal都输入CNN"网络进行计算，没有及逆行相应的共享计算，而SPPnet是利用共享卷积计算的方式来加速RCNN的检测过程，SPPnet将整张图片输入CNN网络得到特征图，然后利用空间金字塔池化网络对每个region proposal区域的特征图进行处理得到固定维度的特征向量，然后训练SVM分类器

为解决上述优点， Fast RCNN主要贡献在于：

- Fast RCNN具有更高的目标检测的精度
- 训练过程采用多任务的损失函数
- 训练可以更新所有网络层的参数
- 不需要额外的磁盘空间存储特征

Fast RCNN算法细节介绍



Fast R-CNN网络将整个图像和一组候选框作为输入。网络首先使用卷积层和最大池化层来处理整个图像，以产生卷积特征图。然后，对于每个候选框，RoI池化层从特征图中提取固定长度的特征向量。每个特征向量被送入一系列全连接（fc）层中，其最终分支成两个同级输出层：一个输出K个类别加上1个背景类别的Softmax概率估计，另一个为K个类别的每一个类别输出四个实数值。每组4个值表示K个类别的一个类别的检测框位置的修正。

ROI pooling layers（感兴趣区域池化）

RoI池化层使用最大池化将任何有效的RoI内的特征转换成具有$H \times W$（例如，7×7）的**固定尺度**的小特征图，其中H和W是层的超参数，独立于任何特定的RoI。在本文中，RoI是卷积特征图中的一个矩形窗口。 每个RoI由指定其左上角(r, c)及其高度和宽度(h, w)的四元组(r, c, h, w)定义。

RoI最大池化通过将大小为$h \times w$的RoI窗口分割成$H \times W$个网格，子窗口大小约为$h/H \times w/W$，然后对每个子窗口执行最大池化，并将输出合并到相应的输出网格单元中。同标准的最大池化一样，池化操作独立应用于每个特征图通道。

微调

Fast RCNN能够使用反向传播来更新训练所有的网络权重。SPPnet不能更新所有的权重，不能更新spp之前层的参数。（**注意**：这里不是说不能更新，而是由于在finetune的过程中反向传播非常低效。）

根本原因是当每个训练样本（即RoI）来自不同的图像时，通过SPP层的反向传播是非常低效的，这正是训练R-CNN和SPPnet网络的方法。低效的部分是因为每个RoI可能具有非常大的感受野，通常跨越整个输入图像。由于正向传播必须处理整个感受野，训练输入很大（通常是整个图像）。

（来自知乎某个大佬的解释：RoI-centric sampling和image-centric sampling的区别：SPP-net是先把所有图像用SS计算的RoIs存起来，再从中每次随机选128个RoIs作为一个batch进行训练，这128个RoIs最坏的情况来自128张不同的图像，那么，要对128张图像都送入网络计算其特征，同时内存就要把128张图像的各层feature maps都记录下来（反向求导时要用），所以时间和空间上开销都比较大；而Fast R-CNN虽然也是SS计算RoIs，但每次只选2张图像的RoIs(一张图像上约2000个RoIs)，再从中选128个作为一个batch，那么训练时只要计算和存储2张图像的Feature maps，所以时间和内存开销更小）

论文提出了一种更有效的训练方法，利用训练期间的特征共享。在Fast RCNN网络训练中，随机梯度下降（SGD）的小批量是被分层采样的，首先采样N个图像，然后从每个图像采样R/N个 RoI。关键的是，来自同一图像的RoI在向前和向后传播中共享计算和内存。减小N，就减少了小批量的计算。例如，当$N = 2$和$R = 128$时，得到的训练方案比从128幅不同的图采样一个RoI（即R-CNN和SPPnet的策略）快64倍。

这个策略的一个令人担心的问题是它可能导致训练收敛变慢，因为来自相同图像的RoI是相关的。这个问题似乎在实际情况下并不存在，当$N = 2$和$R = 128$时，我们使用比R-CNN更少的SGD迭代就获得了良好的结果。

除了分层采样，Fast R-CNN使用了一个精细的训练过程，在微调阶段联合优化Softmax分类器和检测框回归，而不是分别在三个独立的阶段训练softmax分类器，SVM和回归器。 下面将详细描述该过程（损失，小批量采样策略，通过RoI池化层的反向传播）。

损失函数

一个Fast RCNN网络有两个输出层，第一个输出为K+1个类别的离散概率分布，而第二个输出为bbox回归的偏置，每一个正在训练的ROI均利用一个ground truth类别u与ground truth框v，采用多任务损失进行分类与边框回归：

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

第一部分是类别的log损失

第二部分是为止损失的回归损失，V为ground truth， t为预测值，方括号是一个指示函数，满足条件为1，否则为0，按照惯例，u=0为背景类，此时忽略回归损失，对于检测框的回归采用了smooth-L1损失，没有使用L2损失。

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

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

这个损失函数相比于L2损失对于对于异常值更加鲁棒

小批量采样策略

当及逆行fine-tune时，每个SGD的小批量由$N = 2$个图像构成，均匀地随机选择（如通常的做法，我们实际上迭代数据集的排列）。 我们使用大小为$R = 128$的小批量，从每个图像采样64个RoI。 从候选框中获取25%的RoI，这些候选框与检测框真值的IoU至少为0.5。 这些RoI只包括用前景对象类标记的样本，即$u \geq 1$。 剩余的RoI从候选框中采样，该候选框与检测框真值的最大IoU在区间$[0.1, 0.5)$上。 这些是背景样本，并用$u = 0$标记。0.1的阈值下限似乎充当难负样本重训练的启发式算法。 正负样本比例为1：3，防止易分负样本过多。

在训练期间，图像以概率0.5水平翻转。不使用其他数据增强

通过ROIpooling层的反向传播

为了清楚起见，假设每个小批量($N = 1$)只有一个图像，扩展到$N > 1$是显而易见的，因为前向传播独立地处理所有图像。

令$x_i \in \mathbb{R}$是到RoI池化层的第i个激活输入，并且令y_{rj}是来自第r个RoI层的第j个输出。RoI池化层计算$y_{rj} = x_{i*(r,j)}$，其中$x_{i*(r,j)} = argmax_{i' \in \mathcal{R}(r,j)} x_{i'}$。$\mathcal{R}(r,j)$是输出单元$y_{rj}$最大池化的子窗口中的输入的索引集合。单个$x_i$可以被分配给几个不同的输出$y_{rj}$。

RoI池化层反向传播函数通过遵循argmax switches来计算关于每个输入变量x_i的损失函数的偏导数：

换句话说，对于每个小批量RoI r和对于每个池化输出单元y_{rj}，如果i是y_{rj}通过最大池化选择的argmax，则将这个偏导数$\frac{\partial L}{\partial y_{rj}}$积累下来。在反向传播中，偏导数$\frac{\partial L}{\partial y_{rj}}$已经由RoI池化层顶部的层的反向传播函数计算。

尺度不变性（scale invarient）

对于尺度不变性实现，一般采用两种方式：

- "brute force" learning（暴力学习）
- 图像金字塔

在暴力学习的方法中，每张图片处理为相同大小的网络输入，然后网络从特定尺寸的训练数据中学习尺度不变性的目标检测。（**RCNN的方法**）

图像金字塔方法：多尺度方法通过图像金字塔向网络提供近似尺度不变性。 在测试时，图像金字塔用于大致缩放-规一化每个候选框。 在多尺度训练期间，每次图像采样时随机采样金字塔尺度。

Fast RCNN检测

当完成Fast R-CNN的fine-tune之后，检测相当于运行前向传播（假设候选框是预先计算的）。网络将图像（或图像金字塔，编码为图像列表）和待计算概率的R个候选框的列表作为输入。在测试的时候，R通常在2000左右，虽然我们将考虑将它变大（约45k）的情况。当使用图像金字塔时，每个RoI被缩放，使其最接近中的224^2个像素。

对于每个测试的RoI r，正向传播输出类别后验概率分布p和相对于r的预测的检测框偏移集合（K个类别中的每一个获得其自己的精细检测框预测）。我们使用估计的概率$Pr(class = k|r) \triangleq p_k$为每个对象类别$k$分配$r$的检测置信度。然后，对每个类别独立执行非最大抑制。

Truncated SVD for faster detection（截断奇异值分解）

对于图像分类的处理过程， 时间消耗主要在卷积层，全连接层时间花费会很少，但是对于目标检测算法，由于需要处理大量的ROI，因此需要花费一般以上的时间在全连接层上，利用截断的奇异值分解很容易实现对与大的全链接层的加速。

利用这项技术，如果一层的权重矩阵W为$u \times v$维，那么对W执行奇异值分解：

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

在这种分解中，U是一个$u \times t$的矩阵，包括W的前t个左奇异向量，Σ_t是$t \times t$对角矩阵，其包含W的前t个奇异值，并且V是$v \times t$矩阵，包括W的前t个右奇异向量。截断SVD将参数计数从uv减少到$t(u+v)$个，如果t远小于$min(u,v)$，则SVD可能是重要的。 为了压缩网络，对应于W的单个全连接层由两个全连接层替代，在它们之间没有非线性。这些层中的第一层使用权重矩阵$\Sigma_t V^T$（没有偏置），

并且第二层使用U（其中原始偏差与W相关联）。当RoI的数量大时，这种简单的压缩方法给出良好的加速.

总结

文章的主要贡献工作为：

- 分析SPPNet低效的原因
- 使用共享卷积计算加速算法
- ROI Layer
- 使用多任务的损失函数
- 端到端的训练
- 截断的SVD加速