控制与决策
Control and Decision

基于深度稀疏低秩分解的深度神经网络轻量化方法

程旗, 李捷, 高晓利, 唐培人, 盛良睿, 王维

引用本文:
程旗,李捷,高晓利,唐培人,盛良睿,王维. 基于深度稀疏低秩分解的深度神经网络轻量化方法[J]. 控制与决策, 2023, 38(3): 751−758.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.1213

基于深度稀疏低秩分解的深度神经网络轻量化方法

程 旗, 李 捷†, 高晓利, 唐培人, 盛良睿, 王 维

(四川九洲电器集团有限责任公司，四川 绵阳 621000)

摘 要: 基于嵌入式平台对深度神经网络轻量化的需求,结合模块化、逐层处理思想,以主流检测识别深度神经网络Faster RCNN轻量化为目标,设计基于深度稀疏低秩分解的轻量化方法. 针对Faster RCNN网络架构特点,首先采用深度可分离卷积和稀疏低秩理论对Faster RCNN网络的特征提取主干网络部分进行初始轻量化;其次采用稀疏低秩裁剪对主干网络进行“逐层通道裁剪,逐层重训练,逐层调优”轻量化,采用张量Tensor-Train分解理论,对区域建议网络进行轻量化处理,尽可能保证低性能损失;再次对识别与分类网络进行稀疏低秩分解和通道裁剪,增加模型压缩倍数,减少所需要和所消耗计算资源;最后,基于感兴趣区域定位感知的RPN网络输入特征知识蒸馏,提升检测识别性能. 数值实验表明,所提出方法可以实现模型压缩100倍,检测识别率仅下降5 %.

关键词: 轻量化；深度可分离卷积；目标识别；稀疏低秩裁剪；知识蒸馏；区域建议网络

中图分类号: TP391.4 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.1213

引用格式: 程旗,李捷,高晓利,等. 基于深度稀疏低秩分解的深度神经网络轻量化方法[J]. 控制与决策, 2023, 38(3): 751-758.

Lightweight method of deep neural network based on deep sparse low rank decomposition

CHENG Qi, LI Jie†, GAO Xiao-li, TANG Pei-ren, SHENG Liang-rui, WANG Wei

(Sichuan Jiuzhou Electrical Group Co. Ltd, Mianyang 621000，China)

Abstract: Based on the requirement of embedded devices for deep neural network lightweight, and combined with the idea of modularization and layer by layer processing, a lightweight method based on deep sparse low rank decomposition is designed to aim at the lightweight of the mainstream detection and recognition network Faster RCNN. In view of characteristics of the Faster RCNN network architecture, firstly, initially lightening the backbone part of the Faster RCNN feature extraction network is realized through the deep separable convolution and the sparse low-rank theory. Secondly, sparse low-rank pruning is used to further lighten the backbone network in the way of “layer by layer channel pruning, layer by layer retraining, and layer by layer tuning”. Then, the region proposal network is lightened based on the Tensor-train decomposition theory, and the performance loss is ensured as low as possible. Sparse low rank decomposition and channel pruning are applied to the recognition and classification network again, which results in more compression times, less memory and less computing resources required. Finally, the input feature knowledge distillation of the RPN network based on region of interest location perception improves the detection and recognition performance. Numerical experiments show that the proposed method can achieve model compression by 100 times, and the detection and recognition rate is only reduced by 5 %.

Keywords: lightweight; deep separable convolution; target recognition; sparse low-rank pruning; knowledge distillation; region proposal net

0 引 言

随着计算机视觉的发展,应用于视频/图像的深度学习算法在目标检测、识别、分类等任务中得到飞速发展,特别是基于深度卷积神经网络的目标检测与识别算法的成功应用,使得深度学习在计算机视觉领域得到了迅猛发展.

Ren等[1]提出了端到端近实时的目标检测模型Faster RCNN,引入区域建议网络(region proposal network, RPN)用于构建目标候选区域,极大提高了检测效果. 然而,Faster RCNN目标检测框架的模型

收稿日期: 2021-07-13；**录用日期:** 2021-12-09.

†通讯作者. E-mail: kewangdexin@126.com.

存储大小超过1 GB,对硬件存储和计算开销能力提出了更高的要求,需要在拥有高存储高计算性能的GPU (graphics processing unit)服务器上进行,实际工程实践中部署成本较为昂贵,无法直接部署到移动终端、嵌入式设备以及个人电脑等存储空间和计算能力都有限的设备中.

模型压缩是一种有效主流的深度卷积神经网络轻量化方法[2-6],模型剪枝是模型压缩算法中目标较为常见和使用的一种压缩方法[7-11],其通过去除模型中冗余和不重要的参数,保留重要的权值参数,实现整个深度卷积神经网络模型的轻量化设计. 然而,当待裁剪网络层结构较为复杂时,模型剪枝技术往往会出现较大问题. 近年来,基于张量分解的深度学习网络压缩受到学者们的广泛关注[12-16],其核心是利用张量分解的技术将网络的参数重新表达为小张量的组合,达到网络压缩的效果.

本文基于模块化、逐层处理思想,提出一种轻量化Faster RCNN的目标检测识别方法. 首先基于深度可分离卷积实现Faster RCNN主干网络的初次轻量化,并基于稀疏低秩裁剪实现主干网络的二次轻量化;然后,通过张量Tensor-Train分解实现区域建议网络(RPN)轻量化,通过稀疏低秩裁剪实现识别与分类网络轻量化;最后,通过知识蒸馏提升Faster RCNN轻量化后检测识别性能. 数值实验表明,所提出方法可实现模型压缩100倍,而检测识别性能仅下降5%的指标要求.

1 预备知识

1.1 深度可分离卷积[2]

深度可分离卷积将标准卷积先实施深度卷积,再进行逐点卷积. 假设$H \times W$表示输入特征空间尺寸,H和W分别为特征图的高度和宽度,输入和输出特征空间尺寸不变,N为输入特征通道数,$K \times K$为卷积核尺寸,M为卷积核的数量.

1) 深度卷积. 深度卷积将$H \times W \times N$输入特征图按通道分成N组,每个组输入特征只有一个通道.

2) 逐点卷积. 逐点卷积是一种特殊的卷积运算,卷积核尺寸为1×1,具体是指对$H \times W \times N$的输入做M个标准的$1 \times 1 \times N$卷积.

深度可分离卷积的计算量为$HWK^2N + HWNM$,是标准卷积的$1/M + 1/K^2$,参数量为$K^2N + MN$. 因为网络结构中$M \gg K^2$,卷积核空间的尺寸K一般取3,即深度可分离卷积计算复杂度可显著降低标准卷积计算复杂度的1/8 ~ 1/9. 同理,深度可分离卷积模块的参数量可显著减少标准卷积层参数量的1/8 ~ 1/9.

1.2 稀疏表示

神经稀疏编码概念在1988年由Mitchison提出,后由Rolls等正式引用. 用较少基本信号的线性组合来表示大部分或全部原始信号的方式称为稀疏表示.

在机器学习中,稀疏表示的作用相当于“为普通稠密表达的样本找到合适的字典,将样本转化为合适的稀疏表达形式,得以简化学习任务且降低模型复杂度”,稀疏表示又称为“稀疏编码”或“字典学习”.

1.3 低秩分解

低秩分解是指利用多个小矩阵或张量代替原始矩阵或张量,包含以下3种分解算法:奇异值分解(singular value decomposition, SVD)、Tucker分解(高阶奇异值分解, high order sigular value decomposition, HOSVD)、块项(block term, BT)分解.

1.4 知识蒸馏

知识蒸馏是另一种常见的模型压缩方法,其将大型教师网络的知识转移到较小学生网络. 该方法将复杂、学习能力强的教师网络学到的特征表示蒸馏出来,传递给参数量小、学习能力弱的学生网络,以提高学生网络的精度.

1.5 Faster RCNN模型架构分析

Faster RCNN是一种端到端近实时的目标检测识别模型,丢弃Selective Search算法设计思路,引入RPN网络用于构建目标建议区域,包含4个核心部分:

1) 13层卷积层和4层池化层组成的主干网络,用于提取输入图像的深度语义特征,输出特征为Feature Map.

2) RPN网络,用于构建输入图像的目标建议区域,其输入为主干网络输出的Feature Map,输出为不同尺度的目标建议区域(也称为Anchor).

3) 采用RoI池化层代替最后一层最大池化层,可以将不同分辨率的输入对应的深度特征转换为固定长度的输出,以提高整个模型的灵活性.

4) 分类与回归网络,先通过2层全连接层,后分别通过分类和回归,输出目标建议区域所属分类类别及其在输入图像中的精确位置.

2 轻量化Faster RCNN网络架构设计

针对Faster RCNN网络轻量化需求,设计一种基于稀疏低秩裁剪与知识蒸馏的轻量化方法,可实现模型压缩与检测识别性能之间的平衡. 轻量化流程如图1所示.

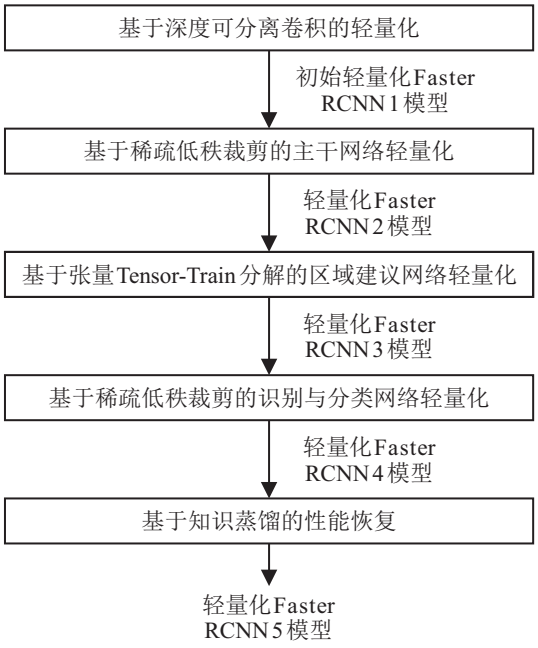


图 1　Faster RCNN的轻量化流程

2.1　基于深度可分离卷积的轻量化

将ImageNet 1 000数据集上训练好的MobileNet模型作为预训练模型,即用MobileNet模型替代初始Faster RCNN模型的主干网络,搭建基于深度可分离卷积的轻量化Faster RCNN1模型. 具体网络架构如图2所示.

特征提取主干网络由1层标准卷积和11层深度可分离卷积构成,用于提取输入图像的深度特征.

2.2　基于稀疏低秩裁剪的主干网络轻量化

2.2.1　裁剪网络层级设计

针对Faster RCNN 1模型特征提取主干网络,考虑到第1层普通卷积层和第1～第4层深度可分离卷积结构重在对图像边框、结构等重要特征的提取,为了保证特征的完整性,前5层初始参数固定不变,第11层深度可分离卷积结构作为区域建议网络的输入,用于构建目标的候选区域. 为了避免对目标检测结果影响过大,不进行裁剪和量化. 因此,对主干网络中第5～第10层深度可分离卷积结构进行逐层处理.

2.2.2　裁剪方法

在稀疏低秩分解、通道裁剪和训练的过程中,裁剪一层,重新训练并调优参数,再裁剪下一层,重新训练并调优参数,如此循环. 算法详细设计如图3所示.

假设第l层深度可分离结构中深度卷积核权重矩阵为$W_l^{dw} \in R^{k\times k\times S_{l-1}\times S_0}$,$1\times 1$点卷积层的权重矩阵为$W_l^{pw} \in R^{k\times k\times S_{l-1}\times S_l}$;第$l+1$层深度可分离卷积结构中深度卷积核权重矩阵为$W_{l+1}^{dw} \in R^{k\times k\times S_l\times S_0}$,点卷积层的权重矩阵为$W_{l+1}^{pw} \in R^{k\times k\times S_l\times S_{l+1}}$. 其中:$k$表示卷积核的尺寸,$S$表示特征通道数,$S_0=1$,$S_l$表示第$l$层深度可分离卷积结构中$1\times 1$点卷积层的特征通道数.

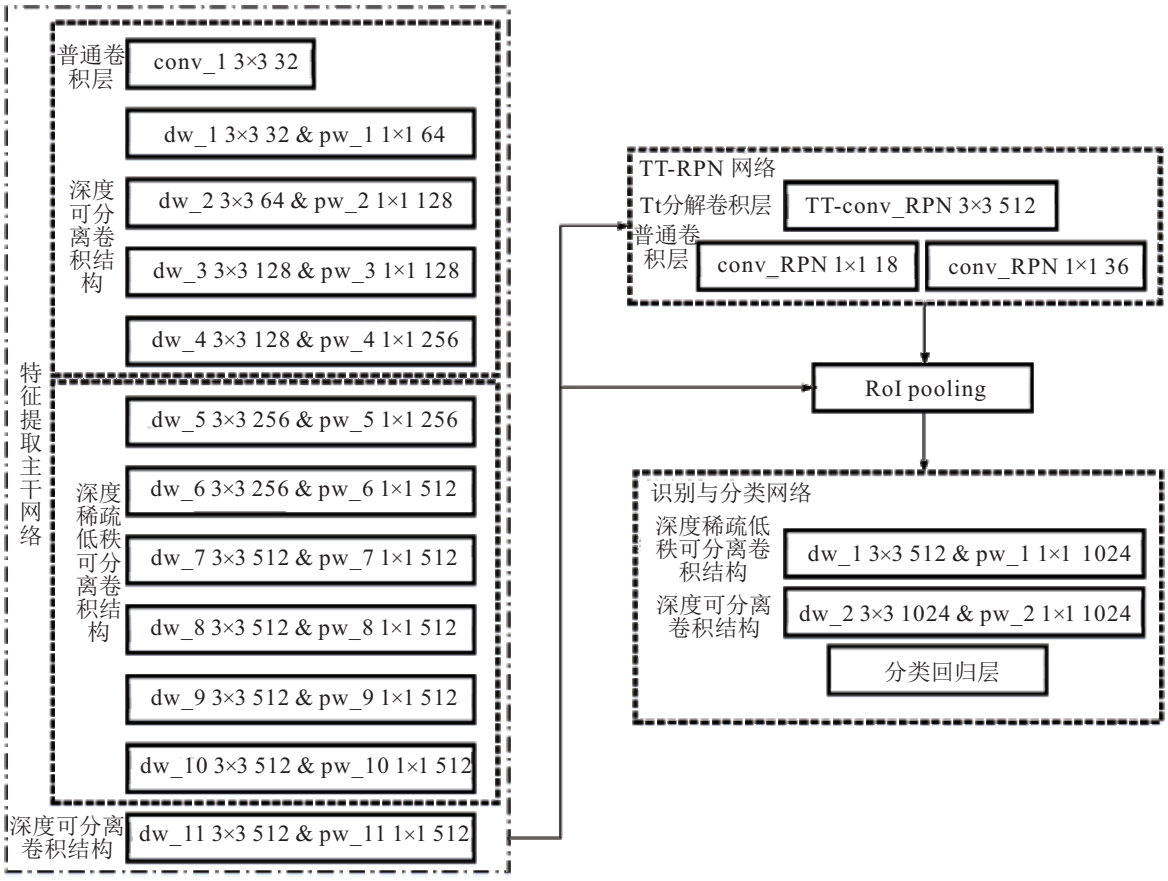


图 2　Faster RCNN轻量化流程

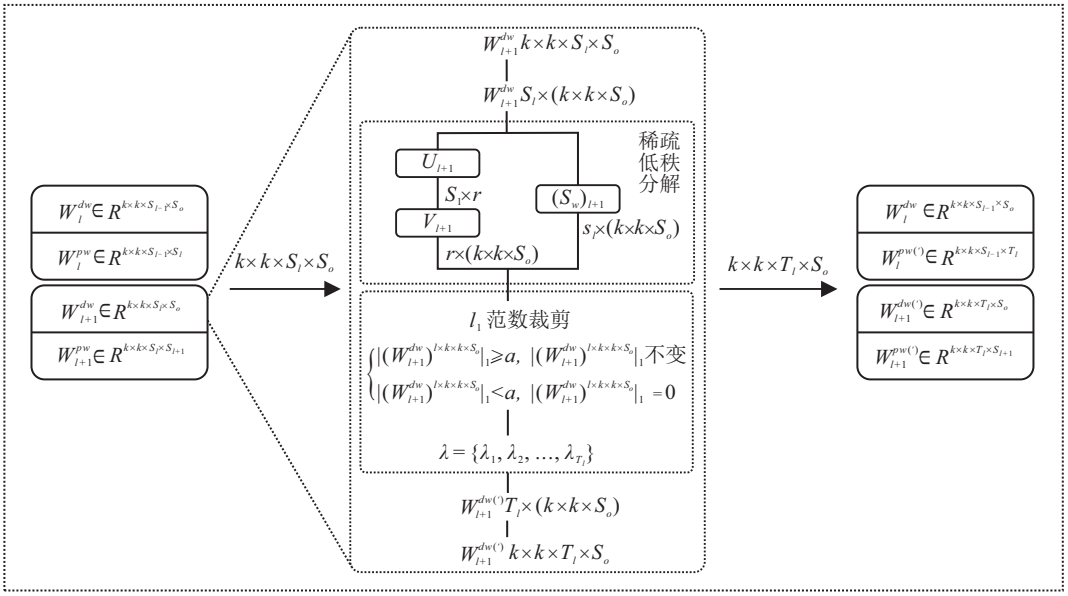


图 3 深度稀疏低秩裁剪可分离卷积结构

1) 稀疏低秩分解.

将W_{l+1}^{dw}按照式(1)分解为低秩矩阵$(L_w)_{l+1}$和稀疏矩阵$(S_w)_{l+1}$的和,并将低秩矩阵根据秩r的大小分解为两个小矩阵U_{l+1}和V_{l+1}的乘积,有

$$W_{l+1}^{dw} = (L_w)_{l+1} + (S_w)_{l+1},$$
$$(L_w)_{l+1} = U_{l+1}V_{l+1}, \tag{1}$$

其中:$W_{l+1}^{dw} \in R^{k\times k\times S_l\times S_0}$重构为$W_{l+1}^{dw} \in R^{S_l\times k\times k\times S_0}$,$U_{l+1} \in R^{S_l\times r}$,$V_{l+1} \in R^{r\times(k\times k\times S_0)}$.

在损失函数中添加正则化项对稀疏矩阵进行约束,得到深度卷积权重矩阵

$$\min_{(L_w)_{l+1}(S_w)_{l+1}} f(W_{l=1}^{dw}) + \gamma\sum_{j=1}^{l}||(S_w)_{l+1}||_1;$$
$$\text{s.t. } W_{l+1}^{dw} = U_{l+1}V_{l+1} + (S_w)_{l+1}. \tag{2}$$

其中:$f(W_{l+1}^{dw})$为损失函数;$||*||_1$为L_1范数;γ用于平衡性能与稀疏性的尺度因子,γ越大稀疏矩阵$(S_w)_{l+1}$越稀疏,压缩效率越高.

2) 通道维度计算及裁剪.

计算每个通道维度对应的L_1范数

$$|(W_{l+1}^{dw})^{i\times k\times k\times S_0}|_1 = \begin{cases} 1, & |(W_{l+1}^{dw})^{i\times k\times k\times S_0}|_1 \geqslant \alpha; \\ 0, & |(W_{l+1}^{dw})^{i\times k\times k\times S_0}|_1 < \alpha. \end{cases} \tag{3}$$

当范数大于等于阈值时,保留对应的通道,并记录通道位置;当范数小于阈值时,删除对应的通道.

3) 新权重矩阵计算.

将L_1范数不为0对应的通道记录在λ集合中,有

$$\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_T\},\ T \in R^{1\times T_l}. \tag{4}$$

根据记录的通道位置,对原权重矩阵进行裁剪后,得到第l层深度可分离卷积结构的点卷积层、第$l+1$层深度可分离卷积结构的深度卷积和点卷积层的新的权重矩阵分别为

$$W_l^{pw(')} = W_l^{pw}(:,\lambda) \in R^{k\times k\times S_{l-1}\times T_l}, \tag{5}$$
$$W_{l+1}^{dw(')} = W_{l+1}^{dw}(:,\lambda) \in R^{k\times k\times T_l\times S_0}, \tag{6}$$
$$W_{l+1}^{pw(')} = W_{l+1}^{pw}(:,\lambda) \in R^{k\times k\times T_l\times S_{l+1}}, \tag{7}$$

其中$T_l < S_l$.

4) 轻量化Faster RCNN2模型.

基于新的权重矩阵,训练后得到第l层轻量化后模型;设置$l = l+1$,当l小于等于预设的轻量化层数时,返回1),否则,完成稀疏低秩分解和通道裁剪,得到轻量化Faster RCNN2模型.

2.3 张量Tensor-Train分解的RPN网络轻量化

2.3.1 1TT-RPN网络结构

为了进一步有效压缩整个模型的存储大小,并保证区域建议网络不丢失过多的特征信息,采用张量Tensor-Train(简称TT)分解理论,只对区域建议网络的第1层$3\times3\times512\times512$标准卷积层进行轻量化设计,构建由TT-Conv_RPN 3×3 512、Conv_RPN 1×1 18和Conv_RPN 1×1 36构成的TT-RPN网络结构.

将标准卷积层的输入张量定义为$X \in R^{W\times H\times C}$,输出张量定义为$Y \in R^{W\times H\times S}$,张量卷积核定义为$K \in R^{k\times k\times C\times S}$. 根据张量TT分解形式,将给定的高阶张量卷积核分解为多个低阶张量卷积核的乘积,张量卷积核K的分解公式如下所示:

$$Y(k_w, k_h, s_1, s_2, \ldots, s_d) =$$
$$\sum_{j_w=1}^{k}\sum_{j_h=1}^{k}\sum_{c_1,c_2,\ldots,c_d} X(j_w + k_w + 1,$$

$$j_h + k_h - 1, c_1, c_2, \ldots, c_d) \cdot$$

$$G_0[j_w, j_h] G_0[c_1, s_1] G_1[c_2, s_2] \ldots G_d[c_d, s_d]. \quad (8)$$

其中$C = \prod_{m=1}^{d} c_m; S = \prod_{m=1}^{d} s_m; k_w, k_h = 1, 2, \ldots, k$为卷积核窗口宽度和高度方向上的任意一点,分别表示卷积核窗口宽度和高度上的迭代变量; $\{G_n\}_{n=0}^{d}$为张量TT核心, $n = 0, 1, \ldots, d$, 当$n = 0$时, $G_0 \in R^{k \times k \times \hat{r}_0 \times \hat{r}_1}$, 当$n = 1, 2, \ldots, d$时, $G_n \in R^{\hat{r}_n \times \hat{r}_{n+1} \times c_n \times s_n}$; $\{\hat{r}_n\}_{n=0}^{d=1}$为张量TT秩,这里$\hat{r}_0$和$\hat{r}_{d+1}$默认固定为1.

2.3.2 空间复杂度分析

利用式(8)将标准卷积分解为$d+1$个低阶卷积核,此时整个标准卷积的空间复杂度为

$$N_2 = (\mathrm{CS})^{1/d}(\hat{r}_1\hat{r}_2 + \ldots + \hat{r}_n\hat{r}_{n+1} + \ldots + \hat{r}_d\hat{r}_{d+1}) + k^2\hat{r}_0\hat{r}_1 = (\mathrm{CS})^{1/d}(\hat{r}_1\hat{r}_2 + \ldots + \hat{r}_n\hat{r}_{n+1} + \ldots + \hat{r}_d) + k^2\hat{r}_1. \quad (9)$$

文中$\hat{r}_n$均设置为20, d设置为3,即第1层4阶张量卷积核分解为4个低阶张量,其中首尾为2阶张量,中间为3阶张量.

原始RPN网络中3×3×512×512标准卷积的空间复杂度为

$$3 \times 3 \times 512 \times 512 = 2\,359\,296. \quad (10)$$

经过张量TT分解后,构建的TT-RPN网络空间复杂度为

$$(512 \times 512)^{1/3} \times (20^2 + 20^2 + 2) + 3 \times 3 \times 20 = 52\,660. \quad (11)$$

因此,张量TT分解在模型压缩性能上具有明显优势,从而得到轻量化Faster RCNN3模型.

2.4 基于稀疏低秩裁剪的识别与分类网络轻量化

由于识别与分类网络的第2层输出主要用于整个模型目标检测任务的分类与回归,对整个模型检测性能影响比较大,为了降低整个模型轻量化对检测性能的影响,只对RPN网络中第1层深度可分离卷积结构进行稀疏低秩分解和裁剪. 裁剪方法与第2.2.2节类似.

经过识别与分类网络的稀疏低秩裁剪得到轻量化Faster RCNN4模型.

2.5 基于感兴趣区域定位感知的RPN网络输入特征知识蒸馏

为了恢复由于网络模型参数量的减少导致的性能损失,结合文献[17]感兴趣区域定位感知技术(fine-grained feature imitation),设计基于感兴趣区域定位感知的RPN网络输入特征蒸馏方法,使学生网络更好地学习到教师网络的RPN输入特征,获得更优的目标检测性能. 首先,利用图像目标的真实检测框,计算其与RPN网络生成的IoU,构成维度为$W \times H \times K$的特征图m. W和H分别为特征图的宽度和高度, K为RPN网络生成感兴趣区域的个数. 随后从特征图m中找到最大的IoU值如下所示:

$$M = \max(m). \quad (12)$$

设置阈值因子(根据实验和经验设置为0.7),由下式计算得到滤波阈值:

$$F = \alpha \cdot m. \quad (13)$$

利用F对特征图进行滤波操作,保留其中大于F的IoU值,并采用加权求和生成细粒度度量的感兴趣区域掩膜I.

根据文献[18],教师网络与学生网络的RPN输入特征欧氏距离与相应损失函数修改如下:

$$\mathrm{Loss}^{(\mathrm{mask})} = \frac{||(f_t - f_s)||_2^2}{||f_t||_2^2}. \quad (14)$$

轻量化Faster RCNN4模型的损失函数定义为

$$\mathrm{Loss}_{\mathrm{FasterRCNN4}} = \mathrm{Loss}^{(\mathrm{hard})} + \lambda \mathrm{Loss}^{(\mathrm{mask})}, \quad (15)$$

$$\mathrm{Loss}^{(\mathrm{hard})} = \mathrm{Loss}_{\mathrm{RPN\_cls}} + \mathrm{Loss}_{\mathrm{RPN\_}}\mathrm{reg} + \mathrm{Loss}_{\mathrm{cls\_cls}} + \mathrm{Loss}_{\mathrm{cls\_reg}}. \quad (16)$$

其中: f_t、f_s分别为Faster RCNN框架、轻量化Faster RCNN4框架中RPN网络的输入特征, $\mathrm{Loss}_{\mathrm{RPN\_cls}}$、$\mathrm{Loss}_{\mathrm{RPN_reg}}$分别为RPN网络的分类损失和边界框回归损失函数, $\mathrm{Loss}_{\mathrm{cls\_ls}}$、$\mathrm{Loss}_{\mathrm{cls_reg}}$分别为分类和回归网络的损失函数, λ为平衡参数,用于平衡两个损失的权重.

3 数值实验

本文依托Ubuntu16.04软件配置环境、 Nvidia GeForce GTX 1080Ti硬件环境以及PASCAL VOC2007数据集,从算法有效性、与传统轻量化方法以及泛化能力3个方面进行仿真验证. 对轻量化Faster RCNN1 ~ Faster RCNN5开展仿真测试.

3.1 算法有效性仿真

轻量化Faster RCNN1 ~ Faster RCNN5仿真测试结果如表1和表2所示. 从表1和表2中可以看出,本文方法可以将网络的模型大小从1 121.88 Mb压缩到11.21 Mb,深度模型压缩率达到100.08倍,检测识别率仅有约5 %的下降,实现了模型压缩与检测率之间的平衡. 同时,检测效率从0.076 s/幅提升至0.034 s/幅,达到29.4幅/s的帧率,基本达到实际应用对于深度学习算法实时性的要求.

表1 5种轻量化方法的检测性能对比分析

模型	mAP (%)	轻量化设计位置	学习率 (MB)	模型大小 (MB)	压缩性能	检测效率 (s/幅)
Faster RCNN	70.81	–	0.001	1 121.88	1	0.076
Faster RCNN1	66.07	第2～第13层卷积层、全连接层 (深度可分离卷积、全局平均池化层)	0.001	45.36	24.73	0.039
Faster RCNN2	65.48	第5～第10层卷积层 (深度可分离卷积、稀疏通道裁剪)	0.001	43.10	26.03	0.039
Faster RCNN3	64.00	第5～第10层卷积层、RPN网络 (深度可分离卷积、稀疏通道裁剪、TT分解PRN)	0.001	36.26	30.94	0.035
Faster RCNN4	63.47	第5～第10层、第12层卷积层、RPN网络 (深度稀疏低秩裁剪可分离卷积结构、张量TT分解)	0.001	11.21	100.08	0.034
Faster RCNN5	65.87	第5～第10层、第12层卷积层、RPN网络 (深度稀疏低秩裁剪可分离卷积结构、张量TT分解PRN、知识蒸馏)	0.001	11.21	100.08	0.034

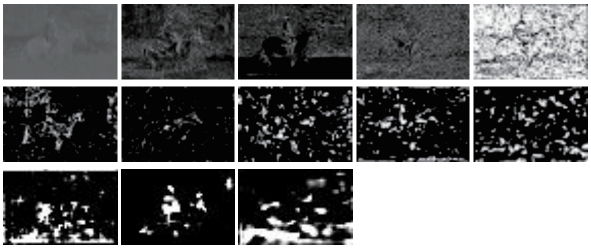
表2 5种轻量化方法在各分类类别的检测性能对比分析

类别	模型					
	Faster RCNN	Faster RCNN1	Faster RCNN2	Faster RCNN3	Faster RCNN4	Faster RCNN5
2(bicycle)	77.95	75.81	76.52	72.17	72.95	73.93
3(bird)	72.33	65.28	63.25	60.58	55.81	60.23
4(boat)	56.15	50.64	49.69	45.13	51.12	51.77
5(bottle)	55.32	45.90	45.16	41.32	39.17	42.17
6(bus)	78.89	66.90	71.65	68.83	70.62	71.03
7(car)	82.36	78.17	79.08	77.65	76.51	79.78
8(cat)	79.89	79.64	78.88	76.50	73.62	79.30
9(chair)	53.00	45.54	43.38	45.19	42.82	46.07
10(cow)	73.77	71.12	71.71	69.04	66.91	67.35
11(diningtable)	67.42	60.06	62.43	56.78	64.25	63.52
12(dog)	80.44	74.76	71.53	68.77	69.97	73.22
13(horse)	84.01	78.70	78.83	79.35	78.40	79.04
14(motorbike)	76.50	74.58	73.88	72.44	70.81	74.16
15(person)	77.79	74.93	73.31	74.18	74.11	75.70
16(pottedplant)	42.87	40.98	38.89	41.35	38.17	41.22
17(sheep)	70.92	67.04	65.16	63.04	63.95	66.03
18(sofa)	67.78	63.95	63.16	63.75	62.25	65.90
19(train)	76.50	72.13	72.28	72.80	70.42	74.34
20(tvmonitor)	72.35	67.17	64.00	65.64	59.06	61.82
mAP/%	**70.81**	**66.07**	**65.48**	**64.00**	**63.47**	**65.87**

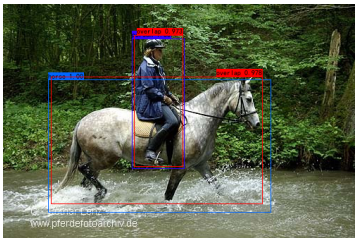
3.2 深层语义特征图提取

不失一般性,以原始图4为例,经过轻量化Faster RCNN5网络13层卷积层提取的深层次语义特征如图5(a)所示,检测结果如图5(b)所示. 其中图5(a)语义特征提取网络参数如表3所示.

图 4　原始图示例



(a)　深层语义特征图序列



(b)　检测识别结果

图 5　深层语义特征图序列和检测结果

表 3　语义特征图提取网络

名称	卷积核	输出通道数	特征图尺寸
原始图	–	3	500×332
语义特征图1	3×3	64	903×600
语义特征图2	3×3	64	903×600
语义特征图3	3×3	128	451×300
语义特征图4	3×3	128	451×300
语义特征图5	3×3	256	225×150
语义特征图6	3×3	256	225×150
语义特征图7	3×3	256	225×150
语义特征图8	3×3	512	112×75
语义特征图9	3×3	512	112×75
语义特征图10	3×3	512	112×75
语义特征图11	3×3	512	56×37
语义特征图12	3×3	512	56×37
语义特征图13	3×3	512	56×37
检测结果图像	–	3	500×332

从图5(a)可以看出,经过13层特征提取网络后,目标语义特征细节日趋明显,为候选框选择及目标检测识别提供了丰富的特征. 从图5(b)可以看出,针对该图例检测识别框与目标真实框重叠区域达97%以上,表明轻量化Faster RCNN5网络具有较好的检测性能.

3.3　算法对比性仿真

将本文方法与传统深度可分离方法、二值化[19]等轻量化方法进行对比仿真,结果如表4所示.

表 4　与传统轻量化方法的对比性分析

模型	mAP (%)	模型大小 (MB)	压缩性能	检测效率 (s/幅)
本文方法	65.87	11.21	100.08	0.034
深度可分离	68.89	130.25	8.614	0.076
二值化	64.36	35.00	32.05	0.005

从表4可以看出: 1) 深度可分离方法有较高的mAP,但压缩性能较差; 2) 二值化方法检测效率较好,但mAP较低; 3) 本文方法在模型压缩的同时,mAP降低程度最小,但检测效率远不如二值化,其原因在于稀疏低秩裁剪、知识蒸馏等过程较为耗时,而二值化方法是将32bit float型数据量化为1bit整型,模型推理相对简单,故耗时较短.

3.4　算法泛化性仿真

为验证算法泛化性,将本文的稀疏低秩裁剪、知识蒸馏等核心技术推广应用于YOLO V3网络,以验证算法泛化性,仿真结果如表5所示.

表 5　YOLO V3轻量化对比性分析

模型	mAP (%)	模型大小 (MB)	压缩性能	检测效率 (s/幅)
YOLO V3	77.3	248.0	1	0.030 7
通道裁剪阈值0.5	73.3	13.1	18.9	0.022 9
通道裁剪阈值0.8	71.7	11.5	21.6	0.022 5
知识蒸馏	74.2	11.5	21.6	0.022 6

针对YOLO V3网络,轻量化mAP仅下降3.1%,压缩性能为21.6倍,检测效率缩短了约26%,验证了本文方法具有较好的泛化能力和推广性.

4　结　论

本文提出的基于深度稀疏低秩分解的深度神经网络轻量化方法,针对Faster RCNN网络,在实现模型压缩100倍的同时性能损失仅下降5%,有效保证了模型压缩与性能损失之间的平衡. 相对于传统的深度可分离卷积、二值化等轻量化方法,所提出方法具有较好的性能,同时也具有较好的泛化能力,可为基于深度神经网络生成的检测识别“大网络”向存储、计算资源有限的嵌入式平台移植提供了必备可靠途径,有助于深度神经网络的工程化实现.

参考文献(References)

[1]　Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[2] Howard A, Zhu M, Chen B, et al. MobileNets: Efcient convolutional neural networks for mobile vision applications[J/OL]. 2017, arXiv: 1704.04861.

[3] Sandler M, Howard A, Zhu M L, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 4510-4520.

[4] Richard S, Charles P, Dawn S. Differentiable neural network architecture search[C]. International Conference on Learning Representations Workshop. Washington, 2018: 1-4.

[5] Yang T J, Howard A, Chen B, et al. NetAdapt: Platform-aware neural network adaptation for mobile applications[C]. European Conference Computer Vision. Munich, 2018: 289-304.

[6] Howard A, Sandler M, Chen B, et al. Searching for MobileNetV3[C]. IEEE/CVF International Conference on Computer Vision. Seoul, 2019: 1314-1324.

[7] Li L, Li Z, Li Y, et al. Incremental deep neural network pruning based on hessian approximation[C]. Data Compression Conference. Snowbird, 2019: 590.

[8] Zhang C L, Hu T, Guan Y D, et al. Accelerating convolutional neural networks with dynamic channel pruning[C]. Data Compression Conference. Snowbird, 2019: 563.

[9] 柳长源, 王琪, 毕晓君. 多目标小尺度车辆目标检测方法[J]. 控制与决策, 2021, 36(11): 2707-2712.
(Liu C Y, Wang Q, Bi X J. Multi-target and small-scale vehicle target detection method[J]. Control and Decision, 2021, 36(11): 2707-2712.)

[10] Yu X Y, Liu T L, Wang X C, et al. On compressing deep models by low rank and sparse decomposition[C]. IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 67-76.

[11] Guo K L, Xie X N, Xu X M, et al. Compressing by learning in a low-rank and sparse decomposition form[J]. IEEE Access, 2019, 7: 150823-150832.

[12] 薛俊韬, 马若寒, 胡超芳. 基于MobileNet的多目标跟踪深度学习算法[J]. 控制与决策, 2021, 36(8): 1991-1996.
(Xue J T, Ma R H, Hu C F. Deep learning algorithm based on MobileNet for multi-target tracking[J]. Control and Decision, 2021, 36(8): 1991-1996.)

[13] Kim Y D, Park E, Yoo S, et al. Compression of deep convolutional neural networks for fast and low power mobile applications[J/OL]. 2015, arXiv: 1511.06530.

[14] Novikov A, Podoprikhin D, Osokin A, et al. Tensorizing neural networks[J/OL]. 2015, arXiv: 1509.06569.

[15] Garipov T, Podoprikhin D, Novikov A, et al. Ultimate tensorization: Compressing convolutional and FC layers alike[J/OL]. 2016, arXiv: 1611.03214.

[16] Su J, Li J, Bhattacharjee B, et al. Tensorial neural networks: Generalization of neural networks and application to model compression[J/OL]. 2018, arXiv: 1805.10352.

[17] Lin M, Chen Q, Yan S. Network in network[J/OL]. 2013, arXiv: 1312.4400.

[18] Wang T, Yuan L, Zhang X P, et al. Distilling object detectors with fine-grained feature imitation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2019: 4928-4937.

[19] Zhao R, Song W N, Zhang W T, et al. Accelerating binarized convolutional neural networks with software-programmable FPGAs[C]. Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York, 2017: 15-24.

作者简介

程旗 (1966−), 女, 正高级工程师, 从事军事电子、空管等研究, E-mail: 260507191@qq.com;

李捷 (1969−), 女, 正高级工程师, 博士, 从事军事电子系统、目标识别和人工智能等研究, E-mail: kewangdexin@126.com;

高晓利 (1983−), 女, 高级工程师, 硕士, 从事数据融合、人工智能等研究, E-mail: xiaoligao22@163.com;

唐培人 (1992−), 男, 工程师, 博士, 从事人工智能、图像处理、目标检测等研究, E-mail: tpr@mail.ustc.edu.cn;

盛良睿 (1995−), 男, 助理工程师, 硕士, 从事深度学习、目标检测等研究, E-mail: 15681989801@163.com;

王维 (1983−), 男, 高级工程师, 从事人工智能、图像处理、数据融合等研究, E-mail: wangwei110v@163.com.

(责任编辑：郑晓蕾)