Mobile_BLNet:基于 Big-Little Net 的轻量级卷积神经网络优化设计

袁海英,成君鹏,曾智勇,武延瑞
(北京工业大学信息学部,北京 100124)

摘 要: 针对深度卷积神经网络难以部署到资源受限的端侧设备这一问题,本文提出一种高效精简的轻量化卷积神经网络 Mobile_BLNet,在模型规模、计算量和性能之间取得了良好的平衡. 该网络引入深度可分离卷积和倒残差结构,通过合理分配不同分支的运算量缩减模型规模并节省大量计算资源;采用通道剪枝操作压缩网络模型,基于占总和比值方法裁剪对模型贡献度低的卷积通道,在相同压缩效果情况下提升了分类准确率;基于通道裁剪情况重构网络,进一步降低模型所需计算资源. 实验结果表明,Mobile_BLNet 结构精简、性能优异,在 CIFAR-10/CIFAR-100 数据集上以 0.1 M/0.3 M 参数量、9.6 M/12.7 M 浮点计算量获得 91.2%/71.5% 分类准确率;在 Food101/ImageNet 数据集上以 1.0 M/2.1 M 参数量、203.0 M/249.6 M 浮点计算量获得 82.8%/70.9% 分类准确率,满足轻量化卷积神经网络的端侧硬件高能效部署需求.

关键词: 轻量化设计;卷积神经网络;模型重构;剪枝操作;深度学习
基金项目: 国家自然科学基金(No.61001049);北京市自然科学基金(No.4172010)
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2023)01-0180-12
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211671

Mobile_BLNet: Optimization Design of Lightweight Convolutional Neural Network Based on Big-Little Net

YUAN Hai-ying, CHENG Jun-peng, ZENG Zhi-yong, WU Yan-rui
(*Faculty of Information Technology*, *Beijing University of Technology*, *Beijing* 100124, *China*)

Abstract: Since it is difficult for deep convolutional neural network to be deployed to terminal equipment with limited resources, this paper proposes an efficient, compact, and lightweight network Mobile_BLNet, which achieves a good balance between model size, computation, and performance. The network uses depthwise separable convolution and inverse residual structure, reduces the scale of the model and saves a lot of computing resources by reasonably allocating the amount of computation of different branches. The total ratio method is used to prune the convolution channel with low contribution, which has excellent performance under the same compression effect. Model reconstruction is based on the clipping, which further reduces the computational resources. The experimental results show that Mobile_BLNet has excellent performance. On CIFAR-10/CIFAR-100 dataset, 91.2%/71.5% accuracy is obtained with 0.1 M/0.3 M parameters and 9.6 M/12.7 M floating point operations. On Food101/ImageNet dataset, 82.8%/70.9% accuracy is obtained with 1.0 M/2.1 M parameters and 203.0 M/249.6 M floating point operations. The network meets the requirements of energy-efficient and lightweight hardware deployment.

Key words: lightweight design; convolutional neural network; model reconstruction; pruning operation; deep learning
Foundation Item(s): National Natural Science Foundation of China (No.61001049); Beijing Municipal Natural Science Foundation (No.4172010)

1 引言

近年来,计算机技术的进步和信息化需求的提升推动了深度学习技术迅猛发展,计算机视觉处理成为当下热门研究方向. 自 2012 年 AlexNet[1] 以巨大优势夺得 ImageNet 分类挑战赛冠军以来,卷积神经网络(Convolutional Neural Networks,CNN)被广泛应用于图像分

收稿日期:2021-12-18;修回日期:2022-03-07;责任编辑:朱梅玉

类、目标检测、图像语义分割和图像转文字等领域. 业界相继提出VGGNet[2],GoogLeNet[3]和ResNet[4]等出色CNN模型,它们在诸多计算机视觉数据集上表现优异.

然而,CNN模型精度提升往往意味着模型尺寸、计算成本成比例增加. 性能优异的CNN模型通常耗费高计算资源和大存储内存,这导致它们难以部署到需要实时推断和低内存占用的应用程序中,例如自动驾驶、智能机器人、医用检测仪以及移动设备上的人机交互系统等[5,6]. 受到人工智能应用驱动,业界陆续提出了许多轻量级网络模型及其压缩方法. 其中,SqueezeNet[7]采用了Fire module的模块化卷积,其参数是AlexNet的1/50,但模型性能与之接近. MobileNet V1[8]是Google提出的可部署于移动端的轻量级网络,采用深度可分离卷积代替传统卷积,在保证网络精度的前提下减少网络参数. 在MobileNet V1基础上,MobileNet V2[9]通过增加线性单元和倒残差结构进一步提高了模型性能. ShuffleNet V1[10]将各部分特征图的通道有序地排列成新的通道,解决了组卷积带来的信息流通不畅问题. ShuffleNet V2[11]从内存消耗成本、GPU并行性两个方面分析了模型可能带来浮点计算量(FLOating Point operations,FLOPs)之外的计算成本,提升了分类精度和计算效率. EfficientNet[12]改进了模型放缩方法,通过神经网络搜索技术平衡了网络的深度、宽度和分辨率,表现性能优异. 此外,低秩分解、网络剪枝、低位量化、知识蒸馏等模型压缩方法也极大地降低了网络规模和计算量.

针对CNN模型在资源受限设备上的部署难题,本文结合了网络轻量化设计与模型压缩技术,提出一种新型轻量级CNN架构Mobile_BLNet. 在轻量化设计方面,该架构以MobileNet V2的倒残差结构为基本单元,结合Big-Little Net[13]的多分支结构并加以改进;在模型压缩方面,采用通道剪枝技术对Mobile_BLNet模型进行裁剪,并以裁剪结果为标准重构模型,最终得到的网络模型能够满足资源受限场景下的部署需求. 网络具体设计流程如图1所示.

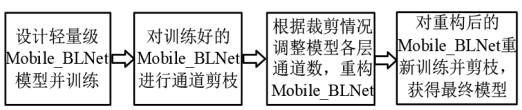


图1　本文轻量级卷积神经网络设计流程

参数量和FLOPs分别是衡量模型空间复杂度和时间复杂度的主要指标. 表1对比了CNN模型在ImageNet数据集中的复杂度,其中ShuffleNet V2模型的2×为性能最佳的版本. 相较于传统CNN模型与主流轻量级CNN模型,Mobile_BLNet大幅度降低了模型时空复杂度,提高了网络轻量化水平.

表1　CNN模型复杂度对比

模型类别	模型名称	参数量	浮点计算量
传统CNN模型	VGG16	138.4 M	15.5 G
	ResNet50	25.6 M	3.8 G
	DenseNet121[14]	8.0 M	2.9 G
轻量级CNN模型	MobileNet V2	3.4 M	300 M
	ShuffleNet V2(2×)	7.4 M	591 M
	EfficientNet b0	5.3 M	399.3 M
本文网络	Mobile_BLNet	2.1 M	249.6 M

2　相关工作

CNN模型的优异性能需要大量卷积计算进行支撑,降低网络所需的庞大计算量会影响模型性能. 因此,在保证性能前提下,如何降低CNN对硬件资源的消耗,使其能够高效部署于资源受限的移动终端应用场景,成为当前业界关注焦点.

2.1　轻量级卷积神经网络设计

有别于传统CNN设计方法(先构建复杂网络再压缩模型),轻量级CNN致力于直接构建紧凑网络体系结构,便于控制参数量和计算量,网络易于实现且具有良好的可扩展性. 下面介绍MobileNet[8,9]与Big-Little Net[13]两种轻量级网络设计方法.

2.1.1　MobileNet模型

MobileNet V1[8]是Google提出的可部署于移动端的轻量级网络,采用深度可分离卷积代替传统卷积,在保证网络精度的情况下减少网络参数.

在传统卷积中,单个卷积核与整张特征图对应,卷积核的通道数与输入特征图通道数一致,卷积核的数量决定了输出特征图通道数. 不同于传统卷积,深度可分离卷积将卷积操作分解为深度卷积和点卷积两部分. 深度卷积中单个卷积核的通道数为1,仅与特征图的输入通道相对应. 这一操作大幅减小了卷积计算量,提升了移动终端正向计算的速率. 点卷积对深度卷积输出的特征图进行交叉融合,从而学习不同通道之间的相关性.

若输入和输出特征图的通道数分别为M和N,特征图与卷积核的大小分别为$D_F\times D_F$与$D_k\times D_k$,则传统卷积与深度可分离卷积对比如图2所示.

在MobileNet V1基础上,MobileNet V2[9]通过增加倒残差结构和线性单元进一步提高了模型性能. 倒残

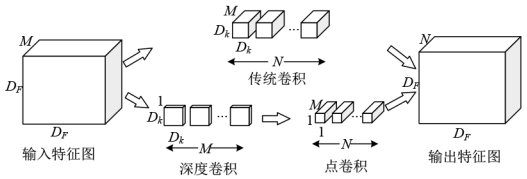


图2　传统卷积与深度可分离卷积对比

差结构如图3所示，首先进行点卷积，将输入通道数C_{in}扩张6倍(扩张程度由扩张系数决定)，再进行深度卷积实现数据处理，在更高的空间维度提取到精确的图像特征，最后采用点卷积将通道数降至输入维度C_{in}. 降维后的特征图与输入特征图进行相加，构成残差结构.

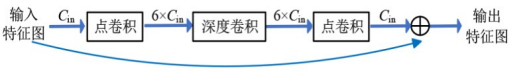


图3 倒残差结构

非线性激活函数在高维空间能够有效地增加模型的性能，但其在低维空间会损坏部分特征信息. 因此，MobileNet V2定义了线性单元，在特征降维后取消了ReLU6激活函数的使用.

2.1.2 Big-Little Net模型

由于CNN模型的计算成本和图像尺度通常成正比关系，若输入图像的长宽均减半，则模型可以节省75%计算量. 因此，Big-Little Net对不同分辨率的输入图像合理分配不同的计算资源，提出了一种简单有效的轻量化卷积神经网络设计理念.

Big-Little Net分为Big分支和Little分支两部分. Big分支复杂度高，但所处理的是下采样后的低分辨率特征；Little分支输入特征未经过下采样，但分支复杂度低. Big-Little Net精确控制了计算资源的使用，模型总体计算成本并不高. 此外，通过将不同比例的图像中所提取到的特征信息相互补充，Big-Little Net的分类效果比使用单一特征更强.

本文基于Big-Little Net的设计思想，以MobileNet V2的倒残差结构为基本单元，构建了网络轻量化水平更高的网络Mobile_BLNet.

2.2 卷积神经网络模型压缩

低秩分解采用奇异值分解等技术[15]，以低秩矩阵形式逼近神经网络中的权重矩阵，在全连接层上尤为有效. 然而，网络运行耗时主要来自卷积层，故低秩分解对运行效率提升有限. 此外，低秩分解应用于卷积层时会出现误差累积，需要对网络逐层微调，否则将导致较大的精度损失.

网络剪枝[16~18]是消除模型冗余并节省计算成本的主流技术，剪枝级别通常分为权重级、通道级和层级. 权重级剪枝灵活性高且泛化性强，能够获得最高的压缩比例，但它需要特殊硬件加速器才能在稀疏模型上进行快速推理[19]. 层级剪枝无需专用环境加速，但其灵活性差，仅在网络深度超过50层时才有明显压缩效果. 当网络深度不够时，层级剪枝会导致模型性能大幅下降[20]. 通道级剪枝在灵活性和易实现性之间提供了一个很好的性能平衡[21]，可以应用于任何典型CNN网络进行推理.

模型量化[22~24]分为2值量化、3值量化和多值量化，其基本思想是以更低的定点精度代替原浮点精度，减少网络中权值所需要的存储空间和计算资源. HashNet[25]提出对权重进行聚类和共享的量化方法，仅需存储共享的权值和索引. 尽管量化极大地提升模型压缩率，但难以保证模型精度.

此外，知识蒸馏[26,27]能有效压缩模型规模，它运用大模型所学到的知识训练小模型，从而让小模型具备大模型的泛化能力.

尽管目前模型压缩技术被大量使用，但其与轻量级网络之间的适应度并不高. 本文结合轻量化设计与模型压缩技术，在轻量级卷积神经网络的基础上，进一步缩减网络规模，提升模型轻量化水平.

3 Mobile_BLNet网络构建

轻量级CNN往往需要简洁有效的架构. 深度可分离卷积结构简单，在有效减少计算量的基础上维持网络性能，被广泛应用于主流轻量级网络设计. 因此，本文采用深度可分离卷积并结合Big-Little Net思想设计出Mobile_BLNet模型，如图4所示.

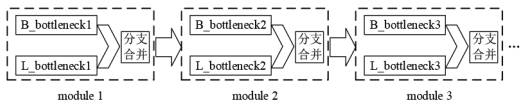


图4 Mobile_BLNet网络模型

Mobile_BLNet主体结构由3~4个module构成. 高复杂度分支BigNet用于处理低分辨率特征图，其在module中的基本单元为B_bottleneck；低复杂度分支LittleNet用于处理高分辨率特征图，其在module中的基本单元为L_bottleneck. 在每个module结构中，需要将这两部分网络的输出特征图进行分支合并，送入下一个module进行迭代处理.

3.1 BigNet设计

主体为倒残差结构的MobileNet V2在推理时不涉及大型张量的计算，由此减少了嵌入式设备对主存储器访问的需求，尤其适合在移动端部署. 目前，EfficientNet[12]等多数模型均采用倒残差结构来兼顾网络性能和轻量化结构设计. 因此，高复杂度分支BigNet以倒残差结构作为网络主要组成部分.

BigNet在每个module中的基本单元为B_bottleneck，如图5所示. B_bottleneck首先将输入特征图进行下采样以减少网络运算量；随后，将特征图依次送入n个重复的block块中逐次进行特征提取. Block块为倒残差结构的基本单元，主要包括特征图升维、处理、降维这三个步骤，操作如下：(1)进行点卷积以扩张通道，输入通道数为C_{in}，扩张后通道数为$C_{in}\times t$，t为控

制扩张程度的扩张系数;(2)采用深度卷积处理数据,卷积核大小为3×3,卷积步长为1;(3)采用点卷积将通道数降至输入维度,输出通道数$C_{out}=C_{in}$,即每个B_bottleneck在升维前后的通道数均保持不变.此外,采用shortcut[4]将输入特征图与输出结果线性相加.

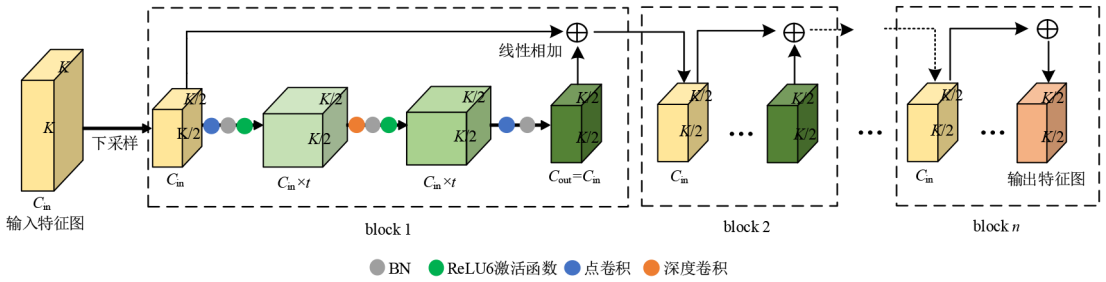


图5　B_bottleneck结构图

由于深度可分离卷积中点卷积最复杂[9],故在网络设计过程中改进了点卷积结构,将B_bottleneck设计为在升维前后通道数保持不变的结构.该结构相较于MobileNet V2中的原始倒残差结构计算复杂度更低.

在Big-Little Net中,为了与LittleNet输出特征图进行分支合并,需要将BigNet输出特征图进行双线性插值上采样到更高分辨率.双线性插值加重了网络的计算负担却无法带来更高的精度收益,这种低性价比方式不适用于计算资源受限的嵌入式设备.因此,在下采样后,BigNet未对图像进行双线性插值上采样操作,而是将LittleNet的输出进行下采样,以实现两个网络的输出特征图相互匹配.

3.2　LittleNet设计

BigNet对输入图像进行下采样的操作虽然极大地减小了计算量,但也丢失了部分上层特征信息.因此,本文对BigNet下采样前的特征图采用LittleNet进行特征提取,以弥补BigNet所丢失的特征信息.

LittleNet在每个module中的基本单元为L_bottleneck,如图6所示,其未采用倒残差结构,主要功能为补充上层网络信息.为避免特征信息损失,L_bottleneck遵从线性瓶颈原则[9],在深度卷积后未采用ReLU6激活函数.

L_bottleneck的输入为原始特征,执行如下操作.(1)通过点卷积来控制输入图像的通道数.有别于BigNet将特征图升到高维,此处点卷积旨在减少输入通道数.通道数量的缩减程度由系数α来控制,当输入特征图通道数为C_{in}时,点卷积后特征图的通道数为$C_{in}\times\alpha$.(2)通过深度卷积对特征进行提取,采用3×3大小的卷积核,卷积步长为2,以保证LittleNet输出特征图与BigNet输出特征图的尺寸一致.(3)使用点卷积对通道信息进行融合,输出特征图通道数$C_{out}=C_{in}\times\alpha$.

3.3　分支合并

分支合并可以融合BigNet和LittleNet两者输出的特征图.此外,它以最小计算成本拓宽了网络宽度,提升了模型性能[28].合并方案有两种:(1)线性组合,通过加法将两个网络的特征合并,每个分支均可以激活彼此的输出神经元,该方案要求所有分支的特征图尺寸和通道数均相同,计算成本较高;(2)拼接,将两个网络的输出直接堆叠,随后用点卷积控制输出通道数,该方案仅需对齐特征图大小,通道数无需一致,计算成本低.然而,拼接操作无法恢复每个分支在合并前损失的信息,如ReLU等非线性操作会彻底舍弃所有小于零的输出.

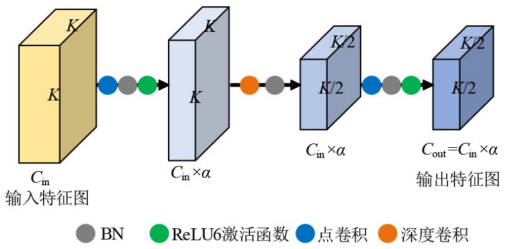


图6　L_bottleneck结构图

考虑到网络拼接方案适合低计算成本的应用场景,本文将BigNet和LittleNet的输出特征图拼接起来,采用点卷积进行特征融合,设置系数β控制输出特征图的通道数.BigNet和Littlenet的输出经过分支合并后送入下一个module[29].

3.4　Mobile_BLNet总体结构

根据原始输入特征图尺寸将Mobile_BLNet分为两类结构:当输入特征图分辨率小于224×224时,采用3个module;当输入特征图分辨率大于等于224×224时,由于图像尺寸较大,需增加一个module以加深网络,进而完成下采样.Mobile_BLNet总体结构如图7所示,单个module结构如图8所示.C_{in}为输入特征图的通道数,引入t,α和β三个参数来约束通道数量,从而控制网络规模.

输入特征图分辨率为32×32的Mobile_BLNet模型结构如表2所示,主体结构包括3个module、卷积层、池化层和全连接层.

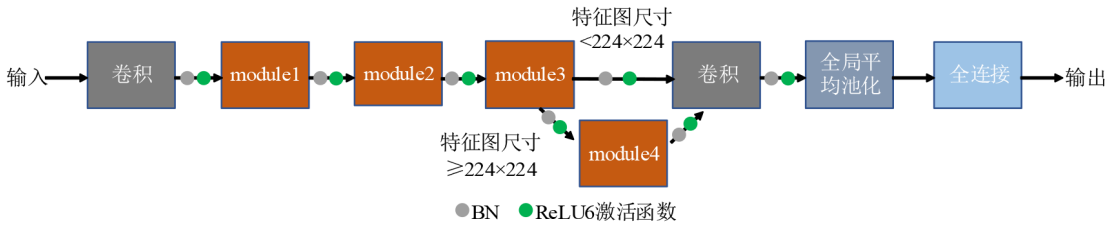


图7 Mobile_BLNet总体结构

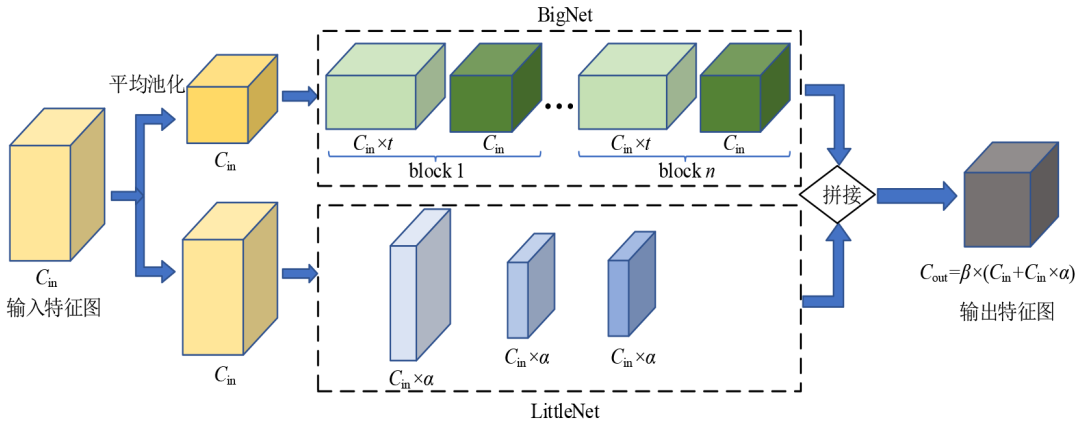


图8 Mobile_BLNet中单个module结构

表2 Mobile_BLNet结构

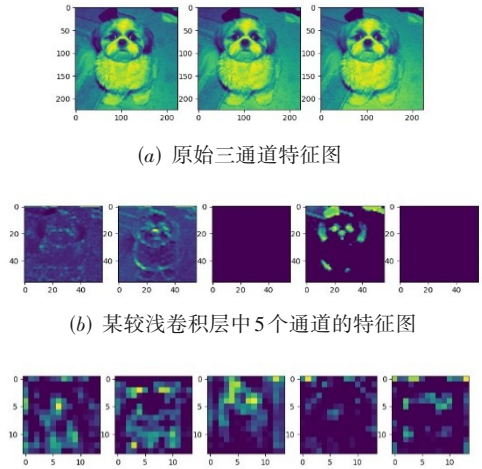
结构	操作	
卷积1	3×3卷积,步长=1,输出尺寸32×32×32	
Module1	B_bottleneck1 (t=6, n=3) 输出尺寸16×16×32	L_bottleneck1 输出尺寸16×16×(32×α)
	拼接并点卷积,输出通道数$k1=\beta\times(32+32\times\alpha)$	
Module2	B_bottleneck2 (t=4, n=4) 输出尺寸8×8×$k1$	L_bottleneck2 输出尺寸8×8×($k1\times\alpha$)
	拼接并点卷积,输出通道数$k2=\beta\times(k1+k1\times\alpha)$	
Module3	B_bottleneck3 (t=2, n=1) 输出尺寸4×4×$k2$	L_bottleneck3 输出尺寸4×4×($k2\times\alpha$)
	拼接并点卷积,输出通道数$k3=\beta\times(k2+k2\times\alpha)$	
卷积2	点卷积,输出通道数=512	
池化	4×4平均池化	
全连接	分类数目	

针对4个module情况:卷积1中步长为2;B_bottleneck的t分别为6、4、2、1,n分别为2、3、2、1;卷积2中输出通道数为1 024;采用7×7平均池化.

3.4.1 扩张系数t

CNN中较浅的层用来提取纹理、色彩等基本特征,网络深度越深,提取的特征越抽象,语义信息越强[4],如图9所示.

采用扩张系数旨在高维空间进行特征提取,由此获取充足的语义信息.然而,对于下采样到较低分辨率的图像,其特征往往拥有丰富的语义信息.如果它采用与较高分辨率图像相同的扩张系数,将会增加网络冗余性,进而浪费计算资源.因此,在下采样过程中采用递减的扩张系数,能够在保留图像特征信息的基础上大幅提升模型轻量化水平.目前,主流网络的倒残差结构扩张系数值均取为6,故本文中扩张系数t以6为初始值并进行递减.



(a) 原始三通道特征图

(b) 某较浅卷积层中5个通道的特征图

(c) 某较深卷积层中5个通道的特征图

图9 CNN中的特征提取示例

3.4.2 LittleNet通道控制系数α

在LittleNet的第一次点卷积时,采用通道控制系数$\alpha(0<\alpha\leqslant1$,默认为1)来缩减输入特征图的通道数,减少模型计算量.然而,大量缩减通道将影响模型性能,第5节将介绍如何合理选取α值.

3.4.3 分支合并通道控制系数β

分支合并将BigNet和LittleNet的输出拼接起来,随后,采用点卷积进行特征融合并控制输出特征图的通道数. 设BigNet和LittleNet的输出通道数分别为a和b,则分支合并后的通道数$k=\beta\times(a+b)$,其中β为通道控制系数.

假定点卷积输入和输出特征的通道数分别为c_1和c_2,特征图的空间大小为$h\times w$,则点卷积的浮点计算量FLOPs为

$$B=hwc_1c_2 \tag{1}$$

内存访问成本(Memory Access Cost,MAC)为

$$\mathrm{MAC}=hw\left(c_1+c_2\right)+c_1c_2 \tag{2}$$

令$c_2=B/hwc_1$,则$\mathrm{MAC}\geqslant 2\sqrt{hwB}+B/hw$. 当$c_1=c_2$时,MAC取最小值[11]. 因此,$\beta$取值在1附近,使得输入通道数尽量等于输出通道数. 在Mobile_BLNet中,β的默认初始值为1.

4 模型剪枝

在保证模型预测效果前提下,优秀的模型压缩技术能够缩减模型规模并减少运算量,利于将模型部署到移动端. 剪枝是一种优秀的模型压缩技术,Liu等人[21]提出一种有效的结构性剪枝方法,即规整的通道剪枝策略. 该剪枝策略在各个通道中引入批量归一化(Batch Normalization,BN)层的γ系数作为缩放尺度因子. 训练期间,在Loss函数中加入BN层γ因子并施加L1正则约束,使得模型朝着结构性稀疏的方向调整参数. 训练完成后,较小的γ因子所对应的卷积层通道将被裁剪.

Loss函数表示为

$$L=\sum_{(x,y)}l\left(f\left(x,\boldsymbol{W}\right),y\right)+\lambda\sum_{\gamma\in\Gamma}g\left(\gamma\right) \tag{3}$$

其中,第一项为CNN的损失函数,$\boldsymbol{W}$为网络权重;第二项是BN层γ因子的L1正则约束项,用来诱导BN层的稀疏化,γ为BN层的缩放尺度因子,λ为一个超参数,用于均衡这两项,约束函数$g\left(\gamma\right)=|\gamma|$.

BN层的γ因子控制信息流通道的开关闭合. γ越接近0,对应输出对结果的影响就越低. 因此,完成稀疏训练后对γ因子的绝对值进行排序,裁剪绝对值较小的γ因子所对应的通道,即可获得低存储占用的精简模型. 如图10所示,灰色部分为需要裁剪的卷积层通道.

上述通道剪枝方法通用性强,本文将其应用于Mobile_BLNet进行模型压缩. 由于BigNet中对通道数进行了扩张,故选择扩张后的BN层施加L1正则化操作. 为了令Mobile_BLNet更好地适应剪枝策略,需要调整λ和裁剪阈值.

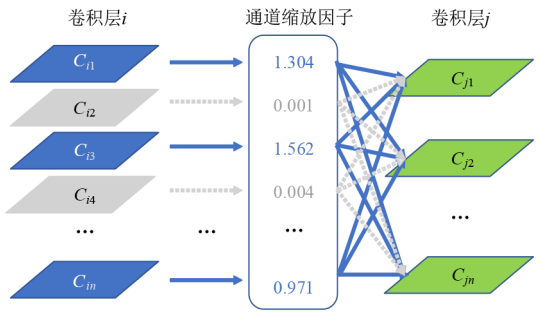


图10　剪枝示意图

4.1 λ取值

采用$\boldsymbol{W}$和γ进行联合训练时,λ值过大意味着大量γ因子被强制缩放到0附近,这导致模型损失一定精度. 在Mobile_BLNet中加入不同λ值进行训练,在CIFAR-10上的准确率如表3所示. 显然,随着λ值增加,模型性能逐渐下降.

表3　Mobile_BLNet在CIFAR-10上的准确率

λ值	$\lambda=10^{-3}$	$\lambda=10^{-4}$	$\lambda=10^{-5}$	$\lambda=0$
准确率	82.5%	91.7%	92.4%	92.7%

不同λ值下同一BN层γ因子的变化情况如图11所示. 当$\lambda=10^{-3}$时,γ因子大量集中于0附近,模型稀疏性明显优于其他三种情况. 可见,λ值越大,模型的稀疏化程度越高,剪枝后模型规模越小,反之亦然. 当$\lambda=10^{-4}$时,网络结构稀疏性与$\lambda=10^{-3}$时相近,且模型性能更优,故本文选择$\lambda=10^{-4}$.

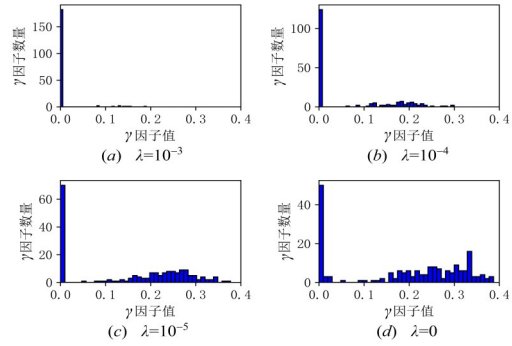


图11　不同λ值下γ因子的分布情况

4.2 阈值选取

文献[21]将γ因子由小到大排序,阈值选择固定比例位置处的取值. 裁剪率为80%表示取排序后在80%位置处的因子作为裁剪阈值. 然而,在稀疏化度较高情况下,大量γ因子分布在0附近,采用固定比例的裁剪方式并非最优解.

鉴于此,本文采用占总和比值法确定裁剪阈值. 将因子由小到大排序并依次相加,当累加结果达到因子总和的1/1 000时确定裁剪阈值. 具体地,针对网络中

任意层，设有f个由小到大排列的因子$\gamma_1,\gamma_2,\cdots,\gamma_k,\cdots,\gamma_f$，其总和$Z_{\text{sum}}=\sum_{i=1}^{f}y_i$. 依次累加求和表示为$Z_{\text{temp}}=\sum_{i=1}^{k}y_i$. 在累加过程中，当第一次满足$\frac{Z_{\text{temp}}}{Z_{\text{sum}}}>\frac{1}{1\,000}$时，记录$k$值，取阈值$\text{th}=\left(\gamma_{k-1}+\gamma_k\right)/2$.

5 模型重构

CNN模型的缩放程度往往取决于手工设计的经验. 神经网络结构搜索(Neural Architecture Search，NAS)是自动机器学习领域的研究热点，它促使神经网络结构设计从手工设计转为计算机自动设计. 然而，NAS搜索空间巨大，其性能评估涉及模型训练，需消耗高昂的运算成本.

本文提出一种简洁有效的模型缩放方法，即根据裁剪率p和q对网络结构进行重构. 设p和q分别反映BigNet和LittleNet中各卷积层的裁剪情况：

$$p=\frac{\sum_{j=1}^{n}\frac{X_j}{Y_J}}{n} \tag{4}$$

$$q=\frac{X}{Y} \tag{5}$$

其中，n为BigNet中bottleneck的重复次数，X为各层被裁剪的通道数，Y为各层的总通道数.

根据通道裁剪率来判断不同层对模型的重要性：层的裁剪率越高，保留的通道数越少，其对模型的贡献度越低，模型重构时将根据通道控制系数减少该层的通道数；层的裁剪率越低，保留的通道数越多，其对模型的贡献度越大，模型重构时将根据通道控制系数增加该层的通道数. 通过该操作，计算资源被尽可能地分配给模型中重要的层，在计算资源和模型性能之间达到良好的平衡.

本文根据裁剪率数值，重新平衡通道控制系数α和β来控制模型的缩放程度. α控制每个module中LittleNet的输出通道数，根据其裁剪率q进行调整；β控制分支融合后的输出通道数，根据下一个module中BigNet的裁剪率p进行调整，最后一个module中β值不调整.

α平衡原则为：减小取值以降低LittleNet网络的复杂度. β平衡原则为：以$1\times\beta$作为调整的基础值从而控制内存访问成本，增减β使计算资源合理分配给对应的卷积层.

采用式(6)和式(7)将α和β分别调整为α'和β'：

$$\alpha'=\begin{cases}\alpha, & q\leqslant 0.4\\ 0.8\alpha, & 0.4<q\leqslant 0.6\\ 0.6\alpha, & 0.6<q\leqslant 0.8\\ 0.4\alpha, & 0.8<q\leqslant 1\end{cases} \tag{6}$$

$$\beta'=\begin{cases}1.5\beta, & p\leqslant 0.4\\ 1.2\beta, & 0.4<p\leqslant 0.6\\ 0.9\beta, & 0.6<p\leqslant 0.8\\ 0.6\beta, & 0.8<p\leqslant 1\end{cases} \tag{7}$$

6 实验与评价

6.1 数据集介绍

6.1.1 CIFAR-10数据集

CIFAR-10数据集由60 000个32×32 RGB图像组成. 图像分为10类，每个类别包含6 000张图片(5 000张用于训练，1 000张用于测试).

6.1.2 CIFAR-100数据集

CIFAR-100数据集由60 000个32×32 RGB图像组成. 图像分为100类，每个类别包含600个图像(500张用于训练，100张用于测试).

6.1.3 Food101数据集

Food101数据集是一个食物数据集，共有101 000张图像(75 750张用于训练，25 250张用于测试). 该数据集有101种食物，每种食物包含1 000张图像(750张用于训练，250张用于测试).

6.1.4 ImageNet数据集

ImageNet数据集是覆盖分类、检测等任务的大规模图像数据集. 其中，分类数据集有1 000个类别，超过120万张图片作为训练集，5万张图片作为验证集. ImageNet由自然图片组成，光线及物体形态差异很大，符合真实世界图像的特点.

6.2 Mobile_BLNet裁剪前的实验结果

6.2.1 训练设置Training setup

选用PyTorch作为实验平台，显卡为两张11 GB显存的GTX 2080 Ti. 实验中采用的超参数以谷歌MobileNet V2中的参数为基准，根据数据集规模有所调整. 训练过程采用了调整大小、标准化、翻转和旋转等数据增强手段和随机梯度下降算法，其中动量设置为0.9. 由于随机梯度下降算法要求将网络的权重初始化为小的随机值，本文对所有模型均采用随机初始化方式设置权重，具体策略为kaiming均匀分布. 在Food101和ImageNet实验中，标准权重衰减设置为4×10^{-5}. 在CIFAR-10和CIFAR-100实验中，为了提升模型拟合度，增大了标准权重衰减，设置为1×10^{-4}.

为了分析所提出模型在不同分辨率图像下的性能表现，实验使用了低分辨率图像数据集CIFAR-10/100，以及高分辨率图像数据集Food101和ImageNet.

在CIFAR-10和CIFAR-100的实验中，采用32×32×3的原始输入特征图尺寸. 学习率余弦衰减，初始设置为0.1；Batch size=64，epoch=160.

在Food101和ImageNet的实验中,设置输入特征图尺寸为224×224×3. 对于Food101数据集,对比网络主要加载于torchvision.model,具体设置为:学习率余弦衰减,初始设置为0.05;Batch size=96,epoch=160. 对于ImageNet数据集,仅测试了Mobile_BLNet模型,其余模型的测试结果来自于相关论文. 具体设置为:学习率余弦衰减,初始设置为0.045;Batch size=320,epoch=300.

损失函数采用交叉熵函数,如式(8)所示:

$$L=-\sum_{i=1}^{k}y_i\log\left(p_i\right) \tag{8}$$

其中,k为样本类别的数量,y_i为样本真实概率分布,p_i为样本预测概率分布.

6.2.2　在CIFAR-10和CIFAR-100上的实验结果

表4中对比了Mobile_BLNet与目前主流轻量级网络在CIFAR-10和CIFAR-100上的测试结果,其中ShuffleNet V1选用打乱通道g为3的版本;ShuffleNet V2选用标准1×版本和性能最优的2×版本;GhostNet选用标准1×版本和轻量化0.5×版本. 由于主流轻量级网络所适应的图片输入分辨率并非32×32,故需对模型进行微调,模型主要来自https://github.com/weiaicunzai/pytorch-cifar100.

表4　不同网络在CIFAR-10和CIFAR-100上的实验结果

网络	参数量/M		浮点计算量/M		准确率/%	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
SqueezeNet[7]	0.7	0.8	54.1	54.9	90.5	69.4
Xception[30]	20.8	21.0	1 135.7	1 135.9	94.8	75.0
MobileNet V1[8]	3.2	3.3	47.2	47.3	89.6	66.0
MobileNet V2[9]	2.3	2.4	94.6	94.7	91.8	68.1
MobileNet V3-Small[31]	1.4	1.5	18.9	19.1	91.6	68.8
MobileNet V3-Large[31]	4.3	4.4	69.5	69.6	93.4	74.3
ShuffleNet V1(g=3)[10]	0.9	0.9	42.2	42.3	89.9	70.1
ShuffleNet V2(1×)[11]	1.3	1.4	47.4	47.5	90.9	69.5
ShuffleNet V2(2×)[11]	5.3	5.5	186.1	186.5	92.0	71.8
GhostNet (0.5×)[32]	1.3	1.4	13.2	13.4	91.4	69.9
GhostNet (1×)[32]	3.9	4.0	44.9	45.0	92.6	72.8
Mobile_BLNet	0.5	0.6	32.9	33.0	92.7	72.9

从表4可见,Mobile_BLNet的参数量和浮点计算量均少于表中所对比的其他轻量级网络,且在CIFAR-10和CIFAR-100上的分类精度优于除Xception和MobileNet V3-Large外的其他网络. 在CIFAR-10上,Mobile_BLNet的分类准确率比Xception低2.1%,比MobileNet V3-Large低0.7%,但其模型复杂度更小. Mobile_BLNet的参数量仅为Xception的2.4%、MobileNet V3-Large的11.6%;浮点计算量仅为Xception的2.9%、MobileNet V3-Large的47.3%. 综合模型规模与性能考虑,Mobile_BLNet的表现更为优异.

综上所述,在CIFAR-10和CIFAR-100这类小型数据集中,Mobile_BLNet拟合能力强,模型时空复杂度极低,拥有出色的模型分类性能.

值得注意的是,EfficientNet b0在上采样处理后的CIFAR-10和CIFAR-100数据集中的表现优良,但其参数量为4 M,浮点计算量为0.39 B,需要在ImageNet上进行迁移训练. 若使用未预训练的EfficientNet b0网络直接在CIFAR-10上训练与测试,则效果不佳,分类准确率低于90%.

6.2.3　在Food101和ImageNet上的实验结果

将Mobile_BLNet在Food101和ImageNet中分别进行网络测试,表5对比了Mobile_BLNet与主流轻量级网络的测试结果,CondenseNet的G和C分别为分组数和压缩率. 各模型在参数量和计算量上的差异主要来自全连接层.

根据表5数据对Mobile_BLNet的模型复杂度和性能进行对比分析.

就空间复杂度而言,Mobile_BLNet的参数量低于表中绝大部分模型,优势突出. 就时间复杂度而言,Mobile_BLNet的浮点计算量处于较低区间且模型性能与同区间模型相近. ShuffleNet V2(1×),GhostNet(0.5×)等网络的参数量和浮点计算量很低,但其模型性能远差于Mobile_BLNet.

就模型性能而言,Mobile_BLNet在Food101数据集上的分类准确率高于大部分网络,仅低于Xception和EfficientNet b0,但Mobile_BLNet轻量化程度更高,其参数量仅为EfficientNet b0的36.6%、Xception的7.1%,浮点计算量为EfficientNet b0的93.7%、Xception的4.4%. 权衡网络复杂度与模型性能,Mobile_BLNet消耗计算

表5 不同网络在Food101和ImageNet上的实验结果

网络类型	参数量/M		浮点计算量/M		准确率/%	
	Food101	ImageNet	Food101	ImageNet	Food101	ImageNet
SqueezeNet[7]	0.8	1.2	751.8	830.1	77.9	57.5
Xception[30]	21.0	22.9	8 418.7	8 420.5	85.7	79.0
MobileNet V1[8]	3.3	4.2	574	575	82.5	70.6
MobileNet V2[9]	2.4	3.4	299	300	83.1	72.0
MobileNet V3-Small[31]	1.5	2.5	55	56	78.2	67.4
MobileNet V3-Large[31]	4.4	5.4	218	219	83.3	75.2
ShuffleNet V1(g=3)[10]	1.0	1.8	139	140	77.3	67.4
ShuffleNet V2(1×)[11]	1.4	2.3	145	146	79.7	69.4
ShuffleNet V2(2×)[11]	5.6	7.4	590	591	83.2	74.9
CondenseNet(G=C=8)[33]	2.0	2.9	273	274	82.7	71.0
GhostNet (0.5×)[32]	1.4	2.6	40	42	75.0	66.2
GhostNet (1×)[32]	4.0	5.2	140	141	82.4	73.9
EfficientNet b0[12]	4.1	5.3	398.2	399.3	84.0	77.1
Mobile_BLNet	1.5	2.4	373.0	373.9	83.8	71.6

资源产生的收益更高. 在ImageNet数据集上,由于仅使用了两张11 GB显存的2080 Ti显卡,训练时Batch size较小,Mobile_BLNet在ImageNet数据集上训练效果较差,但仍与MobileNet V2相当.

此外,MobileNet V3,EfficientNet等主流轻量化网络添加了注意力机制和复杂的激活函数,这些结构能够提升模型性能但硬件适用性较差. 相对而言,Mobile_BLNet结构简单,非常贴合硬件的部署需求. 综上所述,基于Big-Little Net思想与MobileNet倒残差结构所设计的Mobile_BLNet网络在分类精度、参数量和计算复杂度之间取得了良好的平衡,模型复杂度较低但性能出色,是一款贴合端侧部署需求的优秀轻量级模型.

6.3 Mobile_BLNet裁剪后的实验结果

6.3.1 裁剪阈值

下面验证4.2节中占总和比值法确定裁剪阈值的效果. 将Mobile_BLNet分别在CIFAR-10和CIFAR-100中进行了测试(从头开始训练). 将训练好的模型按照不同裁剪率进行裁剪,将其与占总和比值法进行实验对比,结果如表6所示.

表6显示,当固定裁剪率为0.9时,裁剪效果最优,但模型性能下降严重,在CIFAR-10与CIFAR-100上分别为16.61%和1.79%;当固定裁剪率为0.1时,裁剪效果欠佳,但模型性能最优. 采用占总和比值法自动确定裁剪阈值,其裁剪效果接近固定裁剪率为0.7时分类效果,但所对应分类精度在CIFAR-10和CIFAR-100上分别提高11.8%和56.4%.

综上所述,占总和比重法能够根据模型训练情况自动确定最佳裁剪阈值,无需反复调参,在相同分类精度情况下,裁剪效果明显优于固定比值法,对模型剪枝技术的发展有积极作用.

表6 不同裁剪阈值下的实验结果

裁剪率	裁剪后参数量/M	裁剪后浮点计算量/M	裁剪后准确率/%
	CIFAR-10/100	CIFAR-10/100	CIFAR-10/100
0.1	0.49/0.53	30.63/30.69	91.6/69.7
0.2	0.44/0.48	27.73/27.80	91.6/67.0
0.3	0.40/0.43	24.85/24.92	91.0/64.5
0.4	0.35/0.38	21.96/22.03	90.8/61.5
0.5	0.31/0.33	18.93/19.00	89.5/54.1
0.6	0.27/0.28	16.08/16.16	87.2/36.5
0.7	0.22/0.23	13.19/13.27	79.7/15.0
0.8	0.18/0.19	10.31/10.40	41.5/4.7
0.9	0.13/0.14	7.41/7.51	16.6/1.8
占总和比值	0.23/0.37	11.92/15.75	91.5/71.4

6.3.2 Mobile_BLNet裁剪后在各数据集中实验结果

在损失函数中加入BN层γ因子并从头开始训练,训练完成后对模型进行裁剪,实验结果如表7显示. 模型裁剪造成了少量的精度损失,故采用微调技术(学习率=0.01)再次训练裁剪后的网络.

由表7可知,剪枝方案在小型数据集中效果显著,模型在参数量和运算量上的裁剪率均保持较高水平,模型性能与裁剪前大致相当. 在大型数据集中,由于需要更多的模型参数进行拟合,模型裁剪结果略差,但对轻量化网络Mobile_BLNet仍有积极作用.

多个数据集的实验结果表明:本文采用的剪枝方案是一种适用于轻量级网络的优秀模型压缩手段,在几乎无精度损失情况下,极大地降低了Mobile_BLNet

表7　Mobile_BLNet裁剪前后变化

数据集	参数量变化/M	参数量裁剪率/%	FLOPs变化/M	FLOPs裁剪率/%	准确率变化/%	微调准确率/%
CIFAR-10	0.5->0.2	60.00%	32.9->11.9	63.80%	91.7->91.5	91.7
CIFAR-100	0.6->0.4	33.30%	33.0->15.1	54.20%	71.6->71.4	71.5
Food101	1.5->1.2	20.00%	373.0->219.8	41.10%	82.8->81.9	82.8
ImageNet	2.4->2.1	12.50%	373.9->257.5	31.10%	70.8->69.7	70.7

模型的计算成本.

6.4　模型重构结果

根据模型在不同数据集上的裁剪率表现,采用调整后的α'和β'对Mobile_BLNet模型进行重构,重构后模型结构如表8所示.

表8　Mobile_BLNet通道结构变化

Layers	输入通道变化			
	CIFAR-10	CIFAR-100	Food101	ImageNet
B_bottleneck1	32->32	32->32	32->32	32->32
L_bottleneck1	32->32	32->32	32->25	32->32
B_bottleneck2	64->57	64->57	64->68	64->57
L_bottleneck2	64->22	64->34	64->68	64->57
B_bottleneck3	128->71	128->109	128->122	128->136
L_bottleneck3	128->28	128->65	128->73	128->108
B_bottleneck4	—	—	256->292	256->292
L_bottleneck4	—	—	256->116	256->233

在重构模型的损失函数中加入BN层γ因子,从头开始训练模型,再对训练后模型进行结构裁剪. 裁剪后网络性能表现如表9所示.

表9　Mobile_BLNet重构前后在各数据集上的表现

数据集		裁剪前后参数量/M	裁剪前后FLOPs/M	裁剪前后准确率/%	微调准确率/%
CIFAR-10	重构前	0.5->0.2	32.9->11.9	91.7->91.5	91.7
	重构后	0.3->0.1	25.1->9.6	91.2->91.0	91.2
CIFAR-100	重构前	0.6->0.4	33.0->15.1	71.6->71.4	71.5
	重构后	0.4->0.3	27.4->12.7	71.5->71.3	71.5
Food101	重构前	1.5->1.2	373.0->219.8	82.8->81.9	82.8
	重构后	1.2->1.0	350.3->203.0	82.8->81.8	82.8
ImageNet	重构前	2.4->2.1	373.9->257.5	70.8->69.7	70.8
	重构后	2.4->2.1	355.8->249.6	71.0->69.9	70.9

表9显示,Mobile_BLNet模型在重构前后模型性能基本保持一致,但重构后模型的结构更紧凑,时空复杂度大幅降低. 以CIFAR-10数据集为例,Mobile_BLNet仅有0.1 M参数量和9.6 M浮点计算量,但模型准确率高达91.2%.

这种结合剪枝策略的模型重构方式简单快捷,对模型无特殊要求,通用性强,为模型优化设计提供了新思路. 通过重构,本文在模型中合理地分配计算资源,进一步提升了Mobile_BLNet模型的轻量化水平,在保持分类准确率前提下大幅度降低了模型的参数量和计算量,最终设计出一种高能效的轻量化卷积神经网络模型.

7　结论

受限于嵌入式系统的硬件资源和应用场景,移动终端设备在处理实时图像时往往面临两个棘手问题:规模较大的卷积神经网络模型难以进行部署;规模较小的卷积神经网络模型难以保证性能. 本文在分类精度、计算复杂度和模型规模之间进行权衡,提出一种基于深度可分离卷积结构的轻量级CNN模型Mobile_BLNet. 在主体网络中引入倒残差结构,采用低、高复杂度的分支网络分别处理高、低分辨率的特征图,大量节省了网络计算资源;通过结构剪枝操作进一步缩减网络规模,大幅度减少参数量和计算资源;根据网络剪枝情况进行结构调整,提升模型轻量化水平,得到精简的Mobile_BLNet模型. 将该模型在多个数据集上进行测试,实验结果表明:与主流轻量级网络模型相比,Mobile_BLNet作为一种高效紧凑的轻量级卷积神经网络结构,在分类精度、参数量和计算复杂度之间取得了良好的平衡,在保证网络性能前提下满足资源受限的移动终端设备和嵌入式系统的部署需求.

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe Nevada: ACM, 2012: 1097-1105.

[2] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2014-09-04)[2021-12]. https://arxiv.org/abs/1409.1556.

[3] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015: 1-9.

[4] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016: 770-778.

[5] HUANG H H, ZHOU P, LI Y, et al. A lightweight atten-

tion-based CNN model for efficient gait recognition with wearable IMU sensors[J]. Sensors(Basel), 2021, 21(8): 2866.

[6] SHUVO S B, ALI S N, SWAPNIL S I, et al. A lightweight CNN model for detecting respiratory diseases from lung auscultation sounds using EMD-CWT-based hybrid scalogram[J]. IEEE Journal of Biomedical and Health Informatics, 2021, 25(7): 2595-2603.

[7] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[EB/OL]. (2016-02-24)[2021-12]. https://arxiv.org/abs/1602.07360.

[8] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17)[2021-12]. https://arxiv.org/abs/1704.04861.

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

[10] ZHANG X Y, ZHOU X Y, LIN M X, et al. ShuffleNet: An extremely efficient convolutional neural network for mobile devices[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 6848-6856.

[11] MA N N, ZHANG X Y, ZHENG H T, et al. ShuffleNet V2: Practical guidelines for efficient CNN architecture design[C]//European Conference on Computer Vision. Munich: Springer, 2018: 122-138.

[12] TAN M X, LE Q V. EfficientNet: Rethinking model scaling for convolutional neural networks[C]//Proceedings of the 36th International Conference on Machine Learning. Long Beach: MLResearch Press, 2019, 97: 6105-6114.

[13] CHEN C F, FAN Q F, MALLINAR N, et al. Big-Little Net: An efficient multi-scale feature representation for visual and speech recognition[EB/OL]. (2018-07-10)[2021-12]. https://arxiv.org/abs/1807.03848.

[14] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE,2017: 2261-2269.

[15] DENTON E, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. New York: ACM, 2014: 1269-1277.

[16] DONG X Y, HUANG J S, YANG Y, et al. More is less: A more complicated network with less inference complexity[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Honolulu: IEEE, 2017: 1895-1903.

[17] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB/OL]. (2016-08-31) [2021-12]. https://arxiv.org/abs/1608.08710.

[18] TANG Y H, WANG Y H, XU Y X, et al. Manifold regularized dynamic network pruning[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE, 2021: 5016-5026.

[19] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[EB/OL]. (2015-06-08)[2021-12]. https://arxiv.org/abs/1506.02626.

[20] WEN W, WU C P, WANG Y D, et al. Learning structured sparsity in deep neural networks[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. New York: ACM, 2016: 2082-2090.

[21] LIU Z, LI J G, SHEN Z Q, et al. Learning efficient convolutional networks through network slimming[C]//2017 IEEE International Conference on Computer Vision(ICCV).Venice: IEEE, 2017: 2755-2763.

[22] PARK E, YOO S, VAJDA P. Value-aware quantization for training and inference of neural networks[C]//European Conference on Computer Vision. Munich: Springer, 2018: 608-624.

[23] YAMAMOTO K. Learnable companding quantization for accurate low-bit neural networks[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville: IEEE, 2021: 5027-5036.

[24] ZHANG D Q, YANG J L, YE D, et al. LQ-Nets: Learned quantization for highly accurate and compact deep neural networks[C]//European Conference on Computer Vision. Munich: Springer, 2018: 373-390.

[25] CHEN W L, WILSON J T, TYREE S, et al. Compressing neural networks with the hashing trick[C]//Proceedings of the 32nd International Conference on International Conference on Machine Learning. New York: ACM, 2015: 2285-2294.

[26] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[EB/OL]. (2015-03-09) [2021-12]. https://arxiv.org/abs/1503.02531.

[27] ZHU J G, TANG S X, CHEN D P, et al. Complementary relation contrastive distillation[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition

(CVPR). Nashville: IEEE, 2021: 9256-9265.
[28] GOLUBEVA A, NEYSHABUR B, GUR-ARI G. Are wider nets better given the same number of parameters? [EB/OL]. (2020-10-27) [2021-12]. https://arxiv. org/abs/2010.14495.
[29] 袁海英, 成君鹏. 面向移动端图像分类的轻量级卷积神经网络的设计方法: 202110462584.4[P]. 2021-07-02.
[30] CHOLLET F. Xception: Deep learning with depthwise separable convolutions[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017: 1800-1807.
[31] HOWARD A, SANDLER M, CHEN B, et al. Searching for MobileNetV3[C]//2019 IEEE/CVF International Conference on Computer Vision(ICCV). Seoul: IEEE, 2019: 1314-1324.
[32] HAN K, WANG Y H, TIAN Q, et al. GhostNet: More features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020: 1577-1586.
[33] HUANG G, LIU S C, MAATEN L V D, et al. CondenseNet: An efficient DenseNet using learned group convolutions[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 2752-2761.

作者简介

袁海英　女, 1976年出生于四川阆中. 北京工业大学信息学部副教授. 主要研究方向为面向人工智能应用的高能效计算芯片系统、面向信号检测与信息处理的嵌入式系统、电子系统容错与通信总线技术.
E-mail: yhycn@126.com

成君鹏　男, 1995年出生于江苏盐城. 北京工业大学信息学部硕士研究生. 主要研究方向为轻量级卷积神经网络建模技术.
E-mail: chengjp@emails.bjut.edu.cn

曾智勇　男, 1997年出生于北京. 北京工业大学信息学部硕士研究生. 主要研究方向为基于FPGA的卷积神经网络加速器架构设计.
E-mail: m_x_zy@126.com

武延瑞　男, 1996年出生于河北衡水. 北京工业大学信息学部硕士研究生. 主要研究方向为医学影像智能处理技术.
E-mail: 15226525376@163.com