计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2020/14(9)-1441-15
doi: 10.3778/j.issn.1673-9418.2003056

深度神经网络模型压缩综述*

耿丽丽[1,2],牛保宁[1+]
1. 太原理工大学 信息与计算机学院,太原 030024
2. 山西财经大学 实验中心,太原 030006
+ 通信作者 E-mail: niubaoning@tyut.edu.cn

摘 要:近年来,随着深度学习的飞速发展,深度神经网络受到了越来越多的关注,在许多应用领域取得了显著效果。通常,在较高的计算量下,深度神经网络的学习能力随着网络层深度的增加而不断提高,因此深度神经网络在大型数据集上的表现非常卓越。然而,由于其计算量大、存储成本高、模型复杂等特性,使得深度学习无法有效地应用于轻量级移动便携设备。因此,压缩、优化深度学习模型成为目前研究的热点。当前主要的模型压缩方法有模型裁剪、轻量级网络设计、知识蒸馏、量化、体系结构搜索等。对以上方法的性能、优缺点和最新研究成果进行了分析总结,并对未来研究方向进行了展望。

关键词:深度学习;模型压缩;神经网络
文献标志码:A **中图分类号:**TP391.4

耿丽丽, 牛保宁. 深度神经网络模型压缩综述[J]. 计算机科学与探索, 2020, 14(9): 1441-1455.
GENG L L, NIU B N. Survey of deep neural networks model compression[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(9): 1441-1455.

Survey of Deep Neural Networks Model Compression*

GENG Lili[1,2], NIU Baoning[1+]
1. College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China
2. Experimental Center, Shanxi University of Finance and Economics, Taiyuan 030006, China

Abstract: In recent years, the deep neural networks have gained more and more attention with the rapid development of deep learning. It has achieved remarkable effect in many application fields. Usually, at a higher computation, the learning ability of deep neural networks is improved with the increase of depth, which makes the performance of deep learning on large datasets especially successful. However, the deep learning can't be effectively applied to the lightweight mobile portable device due to the characteristics of large amount of calculation, high storage cost and complicated model. Therefore, compressing and simplifying the deep learning model has become the research hot spot. Currently, the main model compression methods include pruning, lightweight network design, knowledge distillation, quantization, neural architecture search, etc. This paper analyses and summarizes the performance, advantages and limitations and the latest research results of the model compression methods, and prospects the

* The National Key Research and Development Program of China under Grant No. 2017YFB1401000 (**国家重点研发计划**); the Key Research and Development Program of Shanxi Province under Grant No. 201903D421007 (**山西省重点研发计划**).
Received 2020-03-24, Accepted 2020-06-02.
CNKI **网络出版**: 2020-06-10, https://kns.cnki.net/KCMS/detail/11.5602.TP.20200609.1752.008.html

future research direction.

Key words: deep learning; model compression; neural networks

1 引言

深度学习作为机器学习领域的分支,近年来在图像识别与检索、自然语言处理、语音识别等诸多领域中都展现出非常优越的性能。深度学习以人工神经网络为基本架构,通过对数据表征进行学习,将底层特征表示转化为高层特征表示,通过多层网络模型完成学习任务。自2016年AlphaGo击败人类顶尖选手,深度学习引起人们普遍关注,同时深度学习的概念也为大众所熟知。长期以来,深度学习研究人员致力于开发更深、更大的模型,达到更高的精度和准确度,同时也导致模型具有大量参数(例如VGG16有一亿三千多万个参数),存储空间占用率高,计算复杂的特性。矩阵运算损耗了庞大的计算资源,并且需要足够的功率;数十亿的网络参数需要大量的存储开销;为了达到优越的学习效果,必须使用GPU加速。对硬件的高要求使得深度网络模型在实际应用中受到限制,诸如手机等便携式以及嵌入式设备,无法满足深度学习的大规模计算要求。因此,在保证网络模型精度及准确度的条件下,压缩网络模型成为一个亟待解决的问题。

精度和准确度一般作为评价网络模型的标准,许多文章中将精度和准确度理解为一个概念,实际是有一定差别的。精度指的是数据测量的重复性如何,即多次测量数据值的离散度,多次测量值集中则精度高,测量值分散则精度低。准确度则是评价测量值和真实值之间偏差的一个指标,反映的是测量值与真实值之间的关系。压缩网络模型的最终目的是产生小规模、高精度及准确度的模型。

压缩的一般意义是通过减少数据大小以节省存储空间和提高传输速率。文中模型压缩是指对深度学习使用的深度网络进行重构、简化以及加速的技术。重构即指利用深度网络的典型模块重新设计一个简单的网络结构;简化指在现有深度网络结构上进行参数压缩、层次以及维度的缩减;加速即是提高深度网络训练、预测的速度。

深度神经网络(deep neural networks,DNN)模型以卷积神经网络为代表,得到广大研究人员的青睐。卷积神经网络一般是由输入层、卷积层、激活函数、池化层以及全连接层构成的前馈神经网络,其中卷积层和全连接层含有大量的参数,网络经过训练之后,参数存在大量冗余,这些冗余的参数是不重要的、可以删除的,去除这些参数并不影响网络的精度。由于参数减少,网络的计算得以简化并且速度大幅提高,从而能提升网络的整体性能。压缩网络这一思想早在文献[1]中就由LeCun等人提出,利用信息论的思想,通过删除网络中不重要的权重,使得神经网络能够更好地泛化,提高学习速率,达到压缩网络尺寸的目的。Han等[2]发表的一篇有关模型压缩方法的综述型文章,作为2016年ICLR的最佳论文,引起了模型压缩研究的热潮。文献[3-4]对近年来模型压缩方法进行了综述。

2 研究框架

深度神经网络模型中的参数决定了模型的复杂程度,因此研究人员将减少网络参数作为压缩优化模型的主要研究方向。另外,网络模型的深度也影响计算时间以及存储空间,因此缩减模型层次也是模型压缩的一个途径。

近年来,许多研究人员在网络模型压缩和加速领域进行了大量研究,提出了众多压缩方法,本文根据模型压缩后对网络结构的影响,将模型压缩方法分为两大类:浅层压缩和深层压缩。模型压缩方法框架如图1描述。

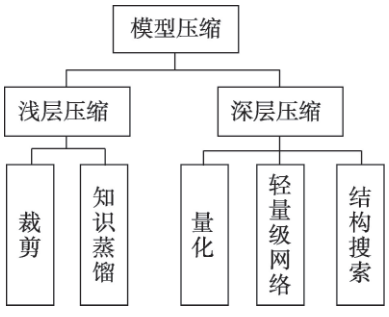

Fig.1 Model compression classification

图1 模型压缩分类

浅层压缩(见第2章):减少参数和模型层次都不会改变网络的结构。主要包括滤波器级别的剪枝、

Table 1 Comparison of network compression methods

表1 网络压缩方法对比

压缩方法	压缩描述	应用网络层级	优缺点	问题领域
裁剪	删除网络中不重要的参数	卷积层 全连接层	保证精度的同时极大地缩减参数规模，但迭代测试阈值耗时且计算量大	标准卷积网络
知识蒸馏	采用迁移学习，训练生成一个小型网络	卷积层 全连接层	可以很好地训练小规模网络，但人工设定学生网络结构会使训练效果产生很大差异	复杂混合网络
量化	将浮点运算转化为定点运算，参数低值化	卷积层	能够以很小的精度损失实现模型体积的大幅减小，但实现难度大，准确性不稳定，通用性差	复杂混合网络
轻量级网络	采用经典网络模块，设计适用于移动设备的网络模型	全部网络层	网络训练简单，计算简化，但设计困难，模型性能欠佳	复杂混合网络
结构搜索	采用神经结构搜索方法设计新的基线网络，通过寻找新的尺度均匀地缩放网络维度	全部网络层	针对具体任务的最优子结构计算简单，比人工设定的网络性能优越，但搜索算法复杂，搜索效率低	复杂混合网络

知识蒸馏。这些方法不会改变网络的基本结构，因此将其归为浅层压缩。

深层压缩（见第3章）：会改变卷积核或者网络层级结构。包括量化、轻量级网络设计、结构搜索方法。

本文分别从浅层压缩和深层压缩两大类对深度神经网络模型压缩方法进行分析：首先分析了浅层压缩方法中模型裁剪以及知识蒸馏的一般压缩流程。在模型裁剪部分对权重裁剪、通道裁剪、核裁剪、神经元裁剪四种裁剪方法进行对比分析。其次对深层压缩方法进行分析，并列举了当前新的研究成果。表1为这几种压缩方法的对比描述，列出了各方法的优缺点、应用网络层以及一般适用的网络模型。

3 浅层压缩

3.1 模型裁剪

模型裁剪是模型压缩使用最广泛的方法。由于卷积神经网络中有大量的冗余参数，裁剪掉一定比例的参数之后，不影响网络的性能。传统意义上的裁剪流程一般包括：预训练原始模型；按照某种规则，对滤波器进行排序；保留排序靠前的一定比例滤波器，其余被裁剪；对裁剪后的模型进行微调再训练，最后达到相同甚至比原模型更高的精度。

3.1.1 权重裁剪

权重裁剪一般是通过在目标函数中引入一个正则化项，使权重趋向稀疏化。权重裁剪的基本流程：

第一步，在网络损失函数中加入一个正则化项，使得网络稀疏化；

第二步，设置一个裁剪阈值，删除权重低于阈值的所有节点；

第三步，对网络参数进行微调之后再训练网络；

第四步，迭代进行下一轮裁剪。

裁剪方法一般是迭代进行，逐层裁剪，每裁剪一部分权重对网络进行一次修复，这样可以在裁剪过程中保证网络的精度。

文献[5]提出了一种基于熵的方法来评价滤波器的重要性，裁掉不重要的滤波器权重，得到了一个更小的网络模型。在VGG-16上实现了16.64%压缩率，在RESNET-50上实现了1.47%压缩率，同时两者精度均下降1个百分点。文献[6]提出一种基于能量感知的卷积神经网络裁剪算法，直接利用卷积神经网络的能耗大小排序进行裁剪，这种方法对卷积网络分层裁剪，每裁剪完一层利用最小二乘法进行局部微调，恢复其精度。裁剪完所有层之后再进行网络全局调整。采用这种方法使得AlexNet和GoogleNet的能耗分别降低了73%和38%，精度损失小于1个百分点。

3.1.2 通道裁剪

通道裁剪是直接将网络中卷积层的整个通道删除。与权重删除相比，通道裁剪属于粗粒度裁剪，随着通道的删除，与之相联系的神经元也全部被删除掉，裁剪力度较大，精度损失也相对较大。较之权重裁剪，它的优点在于：首先，它不产生稀疏矩阵，因此不需要特殊的软件或硬件来针对所得到的模型进行计算；其次，在推理阶段不需要巨大的磁盘存储和内存运行时间。

文献[7]提出的一种基于遗传算法的通道裁剪方

法,用于对超深卷积神经网络进行压缩。根据每一层通道的灵敏度,逐层裁剪网络模型,之后再进行微调。文中将通道选择描述为一个搜索问题,用遗传算法进行求解。在VGG上参数规模压缩了80%,浮点数计算压缩了30%。文献[8]通过基于LASSO回归的通道选择方法和利用最小二乘重构进行有效的网络裁剪。通过此种方法可以使VGG-16在仅增加0.3%的误差下达到5倍的加速。

3.1.3 核裁剪

卷积神经网络的核心部分是卷积核,卷积核中包含有大量的参数,裁剪卷积核也是有效的压缩方法。通过对卷积核参数进行低秩分解,增加稀疏性,减小运算消耗。

文献[9]提出一种特征映射核裁剪方法。设计了选择最小对抗性裁剪掩码策略,根据这种策略可以随机生成裁剪模板,并使用验证集来选择最佳掩码对网络进行裁剪。这种方法与传统的迭代裁剪相比,消耗的时间更少。权重裁剪往往会造成网络模型的不规则稀疏性,不能充分降低卷积层中的计算成本,而对卷积核的低秩分解则可以保证网络模型稀疏性的同时简化矩阵计算,降低计算成本。文献[10]提出一种卷积神经网络加速方法,通过裁剪滤波器及其连接的特征映射,大大降低了计算成本。与权重裁剪相比,这种方法不会导致不规则稀疏连接。因此,不需要稀疏卷积库的支持。文献[11]提出了一个基于泰勒展开的裁剪准则,基于该准则,在细粒度分类任务中,裁剪后的卷积神经网络性能优越。

3.1.4 神经元裁剪

神经网络在接收到输入之后,并不是所有的神经元都被激活,相当大一部分神经元的输出值为零,这些为零的神经元被认为是多余的,删除它们并不会影响网络的精度。因此,出现了针对神经元裁剪的研究。

文献[12]中对激活值为零的神经元进行裁剪,之后利用裁剪前的权重再初始化网络,对网络进行再训练。裁剪和训练迭代交替进行,不断减少网络中的零激活神经元,可以极大地压缩网络规模,是一种迭代优化网络的方法。在LeNet和VGG-16上的实验表明,这种方法可以使压缩率达到2,同时保证模型精度,甚至在裁剪之后可以获得更高的精度。但是文章中并没有说明实验采用的数据类型以及数据规模,对于大型数据集以及特定类型的数据集是否适用有待进一步验证。

另外,在文献[13]中作者指出,裁剪之后仍能保持模型性能并不是归功于所选择的特定裁剪标准,而是由于深层神经网络的固有可塑性,这种可塑性使得网络在精调后能够恢复裁剪造成的精度损失,因此随机裁剪也可以达到在保证精度的同时极大地压缩网络的目标。文中对VGG-16和ResNet-50进行了评估,结果表明,采用一种简单的随机裁剪策略,可以显著提高目标检测速度,同时保持与原网络模型相同的精度。采用裁剪方法对模型进行压缩的相关文献还包括文献[14-22]。其中文献[22]提出一种混合裁剪方法,结合核裁剪以及权重裁剪进行模型压缩,在精度降低很小的情况下,获得了较好的压缩倍率。

以上裁剪方法都是针对卷积层进行,神经网络的全连接层同样包含大量参数,对全连接层进行裁剪能极大地缩小参数量。文献[23]提出一种基于权值相似性的剪枝方法,通过计算权值矩阵相似性删除隐含层单元,实现全连接层的压缩,模型参数减少了76.83%,精度降低只有0.1个百分点。但是该方法没有降低模型运算量,因为卷积网络的运算大部分集中于卷积层。表2为模型裁剪方法对比描述。

表3列举了几种典型裁剪压缩方法的模型性能参数对比,包括模型训练所使用的数据集、Top-*k*精度、参数减少量/压缩率以及浮点运算次数(Flops)减

Table 2 Comparison of cutting and compression methods

表2 裁剪压缩方法总体对比

裁剪压缩方法	压缩描述	优缺点
权重裁剪	在目标函数中引入一个正则化项,使权重趋向稀疏化	可以极大地缩减参数规模,但需要在多个待测试阈值上重复迭代,权重阈值在所有层共享,难以寻找合适的阈值
通道裁剪	删除网络中整个通道	通道裁剪是粗粒度裁剪,会删除大量通道,实现简单,但会严重降低网络性能
核裁剪	对卷积核的参数进行低秩分解,增加稀疏性	内核剪枝可以将密集的内核连接模式更改为稀疏的模式。但稀疏网络还没有成熟的框架或硬件来支持计算,速度提升有限
神经元裁剪	直接删除零激活神经元	实现简单,计算简化,但模型性能欠佳

Table 3 Comparison of typical cutting and compression methods

表3 典型裁剪压缩方法性能对比

文献	模型	数据集	Top-1/%	Top-5/%	（参数减少量/10^6）/（压缩率/%）	(Flops减少量/10^9)/(加速率/%）	问题领域
文献[5]	VGG-16	ImageNet	−1.56	−1.16	—/16.64	—/3.30	分类
	ResNet		−2.04	−1.11	8.18/—	1.34/—	
文献[7]	VGG-16	Cifar100	—	—	8.30/—	0.23/—	分类
	ResNet		—	—	16.36/—	1.44/—	
文献[10]	VGG-16	Cifar10	—	—	0.96/—	0.11/—	分类
	ResNet-56		—	—	0.01/—	0.03/—	
文献[12]	LeNet	Mnist	−2.51	—	—/3.85	—	分类
	VGG-16	ImageNet	−2.08	−1.35	—/2.59	—	

少量/加速率这几项指标。深度神经网络模型训练采用Mnist手写数字集、ImageNet图像集、Cifar10/100图像集居多，此外也有采用目标检测、语音识别等数据集。Mnist数据集包含60 000张28×28像素的训练图像和10 000张测试图像，是手写数字的灰度图像。Cifar10/100数据集共有60 000个32×32像素的彩色图像，分为50 000个训练图像和10 000个测试图像，10/100个分类。ImageNet数据集是目前深度学习图像领域应用非常多的一个数据集，关于图像分类、定位、检测等研究工作大多基于此数据集展开。ImageNet数据集包含大约1 500万幅全尺寸图片，2.2万个分类。Top-k精度在分类检测中是指预测向量中排名前k个值的准确率，一般取k为1、5。参数减少量和压缩率是评价模型参数量的绝对值和相对值，压缩率为原参数量与压缩后参数量的比值。Flops是评价模型的运算复杂度以及加速率指标，以每秒浮点运算次数来评价模型效率是模型设计中最常用的方法。

裁剪对模型精度有一定影响，小规模网络模型，压缩后精度下降要大于大规模网络模型，同时也与选用的数据集有关，大数据集的网络对精度影响要小于小数据集网络，这也充分说明了大规模网络的参数冗余度更高，更适于压缩。

3.2 知识蒸馏

知识蒸馏是另一种常见的模型压缩方法，Hinton等[24]提出知识蒸馏的概念。这种模型压缩是一种将大型教师网络的知识转移到较小的学生网络的方法，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，一般可以提高学生网络的精度。教师网络和学生网络可以是同构也可以是异构的，教师网络传递的知识一般包括概率分布、输出的特征、中间层特征映射、注意力映射、中间过程，在神经元级别上监督学生网络训练，提高模型参数的利用率。

蒸馏模型采用迁移学习，通过将预先训练好的教师模型输出作为监督信号去训练另外一个轻量化网络。将教师模型的泛化能力传递给学生模型的一个有效方法是将教师模型产生的分类概率作为训练学生模型的“软目标”，以指导学生网络进行训练，实现知识迁移。Hinton等[24]认为，最好的训练目标函数L是软目标和硬目标两者的结合。如下公式：

$$L=\alpha L(soft)+(1-\alpha)L(hard)$$

$$0<\alpha<1$$

知识蒸馏不仅仅是缩减网络的规模，其重点在于减化网络结构的同时如何保留网络中的知识。文献[25]通过定义卷积神经网络的注意力机制，强迫学生卷积神经网络模拟强大的教师网络注意力映射，从而显著提高其性能。文中提出基于响应图和基于梯度两种利用热力图传输注意力方法。注意力传输的作用是将教师网络某层的空间注意力映射传递给学生网络，让学生网络相应层的空间注意力映射可以模仿教师网络，从而达到知识蒸馏的目的。文献[26]提出一种结构稀疏学习（learning structured sparsity，SSL）方法来规范滤波器、通道、滤波器形状和网络层深度。SSL可以从较大的卷积网络中学习一个紧凑的结构，以降低计算成本，提高分类精度。

文献[27]采用教师-学生算法，将深度网络与随机森林相结合，生成一个精度高、紧凑性好的学生网络模型。在训练过程中使用一个额外的数据集来防止教师模型（T-Model）过拟合，这个新的软目标数据

Table 4 Performance analysis of typical knowledge distillation methods

表4 知识蒸馏典型方法性能分析

文献	模型	数据集	平均精确率(AP)/%	平均召回率(AR)/%	平均假正类率(AFPR)/%	问题领域
文献[27]	ResNet(基准)	TUD行人数据库	73.5	74.6	3.9	目标检测
	T-Model		85.6	84.6	2.0	
	S-Model		78.3	79.3	2.9	

集能够捕获比原始硬目标数据更多的信息。提出一种新的估计行人姿态方向的方法,实验结果表明,该算法在姿态定位方面的分类性能优于基于卷积网络的其他最先进方法。另外,由于模型结构简单,参数少,所提出的学生模型(S-Model)的计算速度快于其他深度卷积神经网络。如表4。

在文献[28]中,作者用一种新的知识蒸馏算法对18层卷积神经网络进行压缩,得到一个两层的用于实时SAR(synthetic aperture radar)识别系统的网络模型。它是一个三元网络,所有权重都是−1, 0, 1。系统遵循师生范式,Dcnn是教师网络,Mcnn是学生网络。在Mstar数据集上的实验表明,所提出的Mcnn算法能够获得与Dcnn基本相同的高识别率。然而,与Dcnn相比,所提出的Mcnn的内存占用被压缩了99.45%,计算量减少了92.20%。

徐喆等[29]将衡量样本相近关系的比例因子引入知识蒸馏算法,通过调节神经元参数,增强网络的泛化能力,在准确率和分类时间上都有所提升。

4 深层压缩

4.1 量化

量化就是将神经网络的浮点运算转换为定点运算。这不仅可以在移动设备上实现网络的实时运行,同时对部署云计算也有帮助。

神经网络中的运算为浮点运算。一般而言,神经网络模型的参数都是FP32(32位浮点数)表示,可以通过量化,牺牲精度来降低每一个权值所占空间,采用低精度FP16(半精度浮点数)和INT8(8位定点整数)表示,甚至将其量化为INT4(4位定点整数)或INT1(1位定点整数),模型尺寸也随之缩小。浮点数量化可以分为两个步骤(以INT8为例):

第一步,在权重矩阵中找到参数最小值 min 和最大值 max ,确定映射区间 x_{scale} 和零点量化值 x_{zero_point} 。

第二步,将权重矩阵中的每个FP32值转换为INT8类型值。如下公式:

$$x_{float} \in \left[x_{float}^{\min_{float}^{max}} \right]$$

$$x_{scale} = \frac{x_{float}^{max} - x_{float}^{min}}{x_{quantized}^{max} - x_{quantized}^{min}}$$

$$x_{zero\_point} = \frac{x_{quantized}^{max} \times x_{scale} - x_{float}^{max}}{x_{scale}}$$

$$x_{quantized} = \frac{x_{float} + x_{zero\_point} \times x_{scale}}{x_{scale}}$$

量化的常用策略是将32位的权值量化为1位或2位,将极大地减少模型大小并节省内存。但实验结果表明,权值量化后的网络性能明显下降,这对于性能要求高的任务来说是一大损失。为了平衡网络规模和性能之间的矛盾,出现了3位、INT8等新的量化方法,在深度神经网络性能损失最小的同时尽可能节省占用空间。

目前的量化方法主要包括:

二值化神经网络:具有二进制权重、激活单元以及参数梯度[30-36]。

三值化神经网络:权重为(+1,0,−1)的神经网络[37]。

INT8量化:将模型从FP32转换为INT8,以及使用INT8进行推理[38-40]。

其他量化[41-48]。

二值化、三值化网络基于简单的矩阵逼近,忽略对精度的影响,因此当处理大型的卷积神经网络时,这种压缩后的网络精度会明显降低。

文献[2]表明量化卷积层通常需要8 bit,全连接层需要4 bit以避免显著的精度损失。

表5列举了典型量化压缩方法性能参数。可以看出在ImageNet大型数据集上进行模型量化压缩后,精度下降明显,错误率提高,在小型数据集上精度没有大幅降低,因此量化效果受限于数据集的选择。对全连接层参数量化,不能显著降低运算复杂度,因为运算量集中在卷积层。

4.2 轻量级网络

网络模型压缩的另一种思路是设计一个轻量级

Table 5 Weight and network accuracy in quantification

表5 量化中的权重位和网络准确率

文献	模型	数据集	量化层	权重位/bit	Top-1错误率/%	Top-5错误率/%	（参数量/KB）/（压缩率/%）	问题领域
文献[2]	LeNet-300-100	Mnist	卷积层	6	1.58	—	—/40	分类
	LeNet-5			4	0.74	—	—/33	
	AlexNet	ImageNet	全连接层	4	42.78	19.70	—/35	
	VGG-16			4	31.17	10.91	—/49	
文献[34]	MLP	Mnist	全连接层	2	1.18	—	—	分类
		Cifar10		2	8.27	—	—	
文献[43]	自定义CNN模型	Mnist	全连接层	3	2.94	—	84.60/—	分类
		Cifar10		3	8.72	—	192.40/—	
		ImageNet		4	49.35	—	1 792/—	

网络，直接适用于移动和嵌入式设备。一般使用深度可分的卷积结构来构建轻量级深层神经网络，通过改变或重组网络结构以及使用高效的计算方法来简化网络，使网络结构更加紧凑。

4.2.1 MobileNet

Google提出的MobileNet[49]网络模型是适用于移动和嵌入式设备的有效模型。MobileNet模型是基于深度可分离卷积的，其核心思想就是卷积核分解，将标准卷积分解为深度方向卷积和1×1卷积，这样可以有效减少网络参数。卷积核分解，实际是将$K\times K\times M$的卷积分解成$K\times K\times 1$的卷积和$1\times 1\times M$的卷积，其中K是卷积核大小，M是卷积核的通道数。其中$K\times K\times 1$的卷积称为深度可分离卷积，它对前一层输出的特征映射的通道进行$K\times K$的卷积来提取空间特征，然后使用1×1的卷积将通道的信息线性组合起来，1×1卷积也称为点积。如图2。

图2中K为原始卷积核的大小，F为输入特征映射尺寸，深度卷积和标准卷积的计算代价可通过以下公式比较，可以看出分解后深度卷积的运算量降至$1/K^2$左右。

$$\frac{K\times K\times M\times F\times F+N\times F\times F}{K\times K\times M\times F\times F}=\frac{1}{N}+\frac{1}{K^2}$$

在MobileNet网络中使用3×3深度卷积，其计算量比标准卷积下降89%左右，同时精度下降很少。此外，算法中还引入了一个宽度因子α来缩减网络宽度而不是缩减层数。与标准卷积相比，深度分离卷积使得模型在ImageNet数据集上的精度只降低1个百分点，同时参数规模减少了86%。MobileNet V2[50]模型是对MobileNet的改进，该模型中引入了残差结构，使用线性激活函数代替ReLU激活函数来减少特征损失，从而提升了MobileNet的性能。

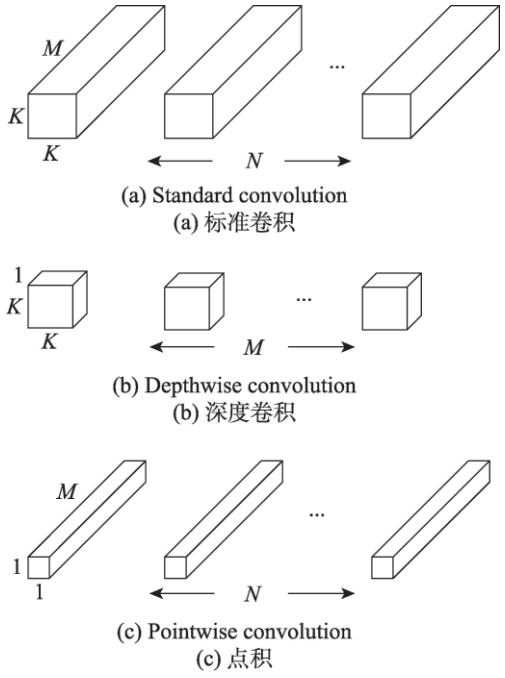

Fig.2 Decomposition convolution

图2 分解卷积

4.2.2 ShuffleNet

虽然分离卷积之后，实现了网络压缩的目的，但是1×1卷积的计算消耗还是比较大，MobileNet在1×1的卷积中花费了94.86%的计算时间，并且占有整个网络模型中75%的参数量。为进一步减少计算量，Zhang等人[51]提出在1×1的卷积上采用分组（group）操作，由于使用1×1卷积的作用是为了整合所有通道的信息，如果使用分组操作就无法使所有通道共享信息，因此又提出了一种channel shuffle的方法。如图3，虽然对1×1的卷积使用了分组操作，但会产生信息不流通的问题，于是在上一层生成的特征映射中增加一个channel shuffle操作。首先将每

个分组中的通道细分为几个组，然后在下一层中输入不同的细分组，使得每个分组都可以接受到上一层不同组的特征信息，解决信息互通的问题，同时还降低了模型计算量。如图3左边显示分组卷积之间没有信息互通。图3右边Channle Shuffle操作将特征映射进行细分之后分配给下一层中的不同组卷积。

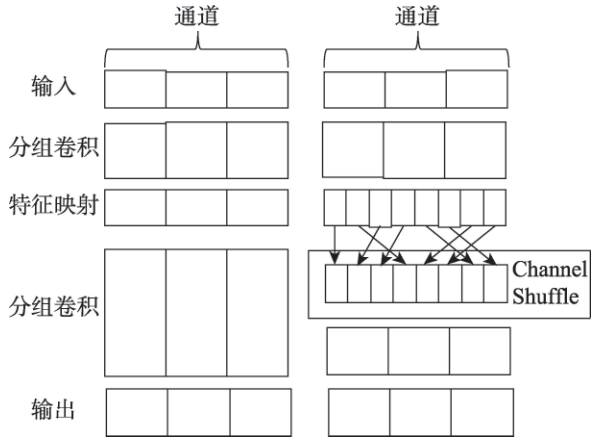

Fig.3 Channel Shuffle structure

图3 Channel Shuffle结构

SuffleNet V2[52]是SuffleNet的改进版，其引入通道分割（channel split）操作，该操作将每个单元的特征通道输入分为两个分支，一个分支保持不变，另一个分支由三个卷积操作组成，两个1×1卷积和一个3×3深度卷积。与ShuffleNet不同的是，这里的1×1卷积不再是组卷积。卷积操作之后，把两个分支通过Concat操作拼接起来，使通道数量保持不变。然后通过ShuffleNet结构中的Channel Shuffle操作进行分支间的信息互通。Channel Shuffle之后，开始进行下一个单元的运算。

文中还指出有效的网络架构设计应该考虑两个原则。首先，应使用直接度量（如速度），而不是间接度量（如Flops）。仅使用Flops作为计算复杂度的唯一指标是不够的，在某些如组卷积操作中，内存的访问成本占很大一部分运行时间，可能成为计算能力强的设备（如GPU）的瓶颈。其次，根据平台的不同，使用相同的Flops操作其运行时间也不同，应在目标平台上评估度量网络能效。

文章还提出了有效的网络架构设计需要遵循的4个准则：（1）使用等通道宽度；（2）避免过度使用组卷积，增加内存存取成本；（3）降低网络碎片化程度，避免因此降低网络并行度；（4）减少元素级操作。

4.2.3 SqueezeNet

介绍SqueezeNet[53]这篇是ICLR 2017的文章，使用的是分类网络中的代表AlexNet，SqueezeNet将AlexNet模型参数减少了99.79%，只有0.5 MB大小。设计者在网络结构中引入了称为Fire Module的模块，该模块由Squeeze卷积层和Expand层组成。Squeeze卷积层只使用1×1卷积，Expand层使用1×1卷积和3×3卷积的组合。如图4，Fire Module模块中有3个可调参数 $S_{1\times1}$、$E_{1\times1}$、$E_{3\times3}$。$S_{1\times1}$ 是Squeeze卷积层中1×1卷积的个数，$E_{1\times1}$ 是Expand层1×1卷积的个数，$E_{3\times3}$ 是Expand层3×3卷积的个数。SqueezeNet使用1×1卷积代替3×3卷积，参数减少了1/9，同时令参数 $S_{1\times1}<E_{1\times1}+E_{3\times3}$，可以限制输入通道的数量，减少总参数量，在SqueezeNet结构中池化层也有所减少。与MobileNet相比，Top-1降低了1个百分点，GPU速度提高了10%。

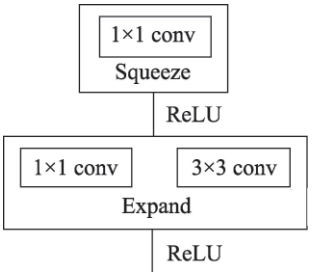

Fig.4 Fire module

图4 Fire模块

Xception[54]也是一种经典的轻量级网络，其认为通道和空间的相关性在卷积操作中是可以分解的，通过分解卷积中的这两部分，可以减小通道和空间上的运算，从而提高模型性能。

Xception卷积分解与以上几种网络卷积分解的不同之处在于：深度分离卷积是先进行深卷积操作，然后进行点积操作；Xception则与之相反，先进行点积操作，然后进行深卷积操作。

除了以上几种经典的轻量级神经网络，还有许多其他效果不错的网络设计。

文献[55]介绍了一种用于可视化和时序数据建模的通用的卷积神经网络EspNetv2。文献[56]中设计了一种基于注意力机制的轻量级卷积神经网络TANet。网络由reduction module、self-attention operation、group convolution三部分组成。其中reduction

module可以减少池化操作带来的信息损失。self-attention operation使模型能够集中学习图像中的重要部分，group convolution实现了模型压缩和多分支融合。该网络可以在移动设备上进行有效的浮游生物分类。

文献[57]提出一种使用迭代硬阈值(iterative hard thresholding，IHT)的方法来训练瘦深度神经网络(skinny deep neural networks，SDNN)。SDNN拥有比卷积神经网络更少的参数，但可以获得更好的性能。使用IHT方法通过两个阶段来训练一个SDNN，首先执行硬阈值，用小激活降低连接，并微调其他重要的过滤器，之后重新激活冻结的连接，训练整个网络以提高其总体识别能力。

文献[58]提出一种简单、高度模块化的图像分类网络体系结构，只需要设置超参数即可重复构造具有相同拓扑变换的块。揭示了一个称之为“基数”的新的维度，通过增大基数可以提高分类精度，比构造更深、更宽的网络更有效。

文献[59]提出一种基于轻量级卷积神经网络资源约束下估计人群计数和生成密度图的新方法。该网络由3个组件组成：基本特征提取器(basic feature extractor，BFE)、堆叠的卷积模块(stacked àtrous convolution module，SACM)和上下文融合模块(context fusion module，CFM)。BFE用降低的空间分辨率对基本特征信息编码，通过SACM中的短流水线来生成各种上下文信息。CFM从上述组件中提取特征图生成上下文融合密度图，整个网络以首尾相连的方式进行训练，并使用压缩因子来限制其大小。

文献[60]提出一个小特征提取网络(feature distilled network，FDN)，通过模仿一个更深的网络中间表示来进行跟踪。引入一种移位拼接方法来减少运算，针对提取出来的特征，提出一种尺度自适应鉴别相关滤波器，用于视觉跟踪，以处理目标的尺度变化。与目前最先进的深度跟踪器相比，速度提高了5倍。

表6对比了几种典型的轻量级网络模型性能。在ImageNet数据集上Xception的精度较优，但参数规模也较大，说明模型精度与规模之间的正比关系。

4.3 体系(网络)结构搜索

随着模型压缩研究的进展，人们逐渐开始关注神经网络自身的结构问题，深度神经网络结构在深度学习中是起决定作用的关键要素，每一类任务都有其最适合的神经网络结构，简化深度神经网络结构可以极大地压缩网络模型，因此关于网络结构搜索方法的研究引起了研究人员越来越多的关注。

网络结构搜索一般采用神经结构搜索方法设计新的基线网络，通过寻找新的尺度均匀地缩放网络维度。文献[61]是关于神经结构搜索(neural architecture search，NAS)算法的一篇综述文章，体系结构搜索正处于快速发展阶段，不断涌现出许多新的研究方法。

NAS是一种自动设计神经网络的技术，可以通过某种算法自动设计出高性能的网络结构，有效地降低神经网络的使用和实现成本，这些神经网络结构已经在性能上逐渐超过人工设计的网络结构。

NAS方法的原理是给定一个搜索空间，用某种搜索算法从中搜索出一个最优神经网络结构，每次迭代产生的神经网络称为子网络，训练这些子网络，评估其性能，直到找到最优子网络。其中搜索空间定义了被算法搜索到的神经网络的结构、大小，也决定了NAS算法的搜索时间。

文献[62]用强化学习解决NAS问题，将强化学习算法应用于模型预测，使用循环神经网络控制器预测参数，自动产生一个神经网络，但面临计算量大的问题，实验使用了800个GPU。文献[63]提出了一种称为NASNet体系结构的方法，利用搜索方法在感兴趣的数据集中找到良好的卷积体系结构，通过搜索

Table 6 Performance comparison of lightweight network

表6 轻量级网络性能对比

文献	模型	数据集	Top-1/%	Top-5/%	参数/10^6	Flops/10^6	问题领域
文献[49]	MobileNet	ImageNet	60.2	—	1.32	76	分类、检测
文献[51]	ShuffleNet	ImageNet	43.2	—	—	38	分类、检测
文献[52]	SqueezeNet	ImageNet	57.5	—	1.25	1 700	分类
文献[54]	Xception	ImageNet	79.0	94.5	2.29	—	分类
文献[56]	TANet	Plankton	77.6	96.3	1.15	—	分类

注：Plankton为浮游生物数据集。

CIFAR-10数据集上最好的卷积层,预测出基本块(building block),然后将该块应用到ImageNet数据集中,方法是将块的更多副本堆叠在一起,根据每个块各自的参数设计一个卷积结构,这样既降低了搜索空间的大小,还增强了网络结构的泛化性。

基于离散空间的搜索算法存在计算量大的问题,文献[64]提出了一种称为可微结构搜索(differentiable architecture search,DARTS)的算法,将搜索空间转化为连续的领域,通过采用梯度下降的方式来进行优化,这种方法可以得到更高的精度,节省计算资源,也可以同时进行卷积和循环结构的搜索。

文献[65]提出了彩票假设:认为网络中包含中奖网络即子网(中奖彩票),并设计了一种识别中奖彩票的算法。在训练时,该网络在相同的迭代次数中达到与原始网络相当的测试精度。它们之间的连接权重具有初始权重,这使得训练特别有效,虽然网络裁剪技术可以使训练后的网络参数减少90%以上,但是裁剪必须是在网络训练之后进行,如果直接对网络进行裁剪,将会极大地影响网络精度,而中奖彩票网络则比原来的网络学习更快,可以达到更高的测试精度。

文献[66]提出了一种自动移动神经结构搜索(automated mobile neural architecture search,MNAs)方法,将模型延迟明确地纳入到主要目标中,从而使搜索能够识别出一个在准确性和延迟之间取得良好平衡的模型。通过在移动电话上执行模型来直接测量真实世界的推理延迟,而不是通过代理任务来进行学习培训。文献[67]也考虑到传统的神经结构搜索算法计算量过大,需要利用代理,在代理任务上优化的体系结构不能保证在目标任务上是最优的。提出一种可直接学习大型目标任务体系结构和目标硬件平台的无代理神经结构搜索,解决了高内存消耗问题,并将计算成本降低到相同水平的常规训练。

Bergomi等人在2019年*Nature Machine Intelligence*发文[68]提出根据拓扑数据分析理论(TDA)可以使神经网络从局部特征理解全局特征,通过TDA选择过滤器来观察数据,找到网络的拓扑特征。研究解决了以下简单问题:当训练一个深层次的神经网络来区分路径时,如何告诉网络,它只需要关心简单的几何形状,比如圆和三角形,它的工作就会容易得多,以便使它探索一个更有限的可能特征空间。

基于网络结构搜索的方法实现深度神经网络压缩正逐渐成为模型压缩领域的主流方法。相关文献还包括文献[69-83]。

通过表7中参数的比较,可以看出结构搜索方法所设计的模型精度、参数量指标都较优,但是运算量很大,计算复杂,搜索时间长是结构搜索方法的瓶颈。

5 模型压缩方法存在问题及未来研究方向

通过以上模型压缩方法的介绍,目前深度神经网络模型压缩主要是针对深度卷积网络进行简化。不同的压缩方法作用网络层不同,也有各自的优缺点。以下对各种压缩方法的特点进行总结并指出未来的研究方向。

(1)模型裁剪是深度神经网络模型压缩研究方法中使用最多的一种有效方法,但是不同的裁剪方法都针对特定任务的分类,无法适用于多目标任务,由于神经网络自身的可塑性,裁剪之后几乎都可以在保证一定精度的同时达到网络压缩的效果。传统裁剪方法需要在多个待测阈值上进行反复迭代,手动设置灵敏度,对参数进行微调,不仅耗时而且计算量大,同时由于权重阈值在网络所有层共享,因此难以寻找到一个合适的阈值。

目前大多数已有研究工作都专注于设计用于滤波器排序的准则,排序准则和真实模型分类、检测的准则是不一样的,采用分段式独立的裁剪方式无法得到神经网络的最优性能。在模型压缩过程中,预训练、排序和微调是相互独立的,很难保证最后模型的性能。此外,传统的裁剪方法都采取统一的裁剪比例,但网络不同层的参数其冗余程度不一样,相同

Table 7 Performance comparison of typical NAS networks

表7 典型NAS网络性能对比

文献	数据集	Top-1/%	Top-5/%	参数/10^6	Flops/10^6	问题领域
文献[64]	ImageNet	74.1	91.0	4.9	595	图像分类、语言建模
文献[66]	ImageNet	76.7	93.3	5.2	403	图像分类、语言建模
文献[67]	ImageNet	75.1	92.5	5.7	—	图像分类

的裁剪比例势必会造成裁剪的过度和欠缺。总之，裁剪可以有效地压缩模型，其关键是如何衡量权重对于模型的重要性，如何选择需要裁剪掉的权重值有众多策略，但对于深度学习来说，没有理论解释哪一种策略是最优策略。

如何对裁剪操作进行形式化描述与推理，以得到一个更加理论化的选择标准，是下一步亟待解决的问题。

(2)知识蒸馏方法中教师模型信息的丰富程度在模型训练过程中起着至关重要的作用。信息越丰富，训练效果越好，知识蒸馏可以很好地进行小规模网络的训练。但与主流的裁剪、量化等技术相比，还存在一定的差距，学生网络结构的构造一般是由人工指定的，最后的训练效果也会因此有很大差异。

因此，如何根据教师网络结构来设计一个合理、能够获取较高模型性能的学生网络结构，是未来的一个研究重点。

(3)参数量化算法，无论是二值量化、三值量化或者多值量化，其本质都是将多个权重映射到一个数值，实现权重共享，从而降低存储以及运算开销。参数量化作为一种主流的模型压缩技术，能够以很小的精度损失实现模型体积的大幅减小。其不足之处一方面在于量化实现难度大、准确性不稳定，网络经过量化之后，很难再进行其他改变；另一方面，通用性较差，往往需要特定的硬件支持，一种量化方法需要开发一套专门的运行库，增加了实现难度和维护成本。

(4)轻量级网络因为其设计方法是借鉴已有的深度神经网络结构，采用经典的网络模块重新架构神经网络，不是对已有网络模型的简化，由于其模型简单、存储空间占用低、计算简化等性能将其归于模型压缩研究范畴。轻量级网络的设计也是针对某种特定任务，特别设计适用于移动设备的网络模型，虽然这种网络的设计使得深度学习落地，能够广泛应用于智能设备，但其具有任务单一、泛化性差的缺点，也使得深度学习这一技术有大材小用之嫌。

(5)深度网络模型的训练十分耗时，虽然直接训练一个小型轻量级网络模型省时省力，但是设计结构却是一项困难的任务，这对于设计者来说，需要足够丰富的经验与技术。另外，轻量级网络由于其参数量小，模型的性能尤其是泛化性不能与大模型相媲美。虽然目前体系结构搜索方法取得了显著的进展，但其主要应用于有关图像分类任务的网络模型压缩，针对其他领域的算法研究比较少。另外，搜索空间也存在局限性，空间中的部分参数需要人工指定，搜索策略不能实现自动搜索。

因此，设计能够适应多领域空间搜索的搜索算法是体系结构搜索方法未来的研究重点。此外，体系结构搜索算法的计算量仍然很大，如何降低计算复杂度、提高搜索效率也是未来研究方向之一。体系结构搜索方法都是采用代理机制，先在小的代理任务上进行搜索性能的测量，然后移植到实际模型，这样难以保证搜索算法的性能。能否不采用代理机制，直接对大规模的应用设计体系结构搜索策略也是一个亟待解决的问题。

一般地，裁剪和知识蒸馏方法用于具有卷积层和全连接层的深度神经网络中，可以获得相似的压缩性能。对于每种压缩方法，没有一个标准评价哪一种压缩效果最好，网络压缩效用的评价标准并不统一。目前，几乎所有的评价标准都是侧重于压缩后和压缩前模型几个性能的比较，比如TOP-1、TOP-5、加速比、存储空间节省率、压缩率指标来表示模型压缩效果。如何选择合适的压缩方法，取决于具体的应用需求以及网络类型。文献[84]给出了一些针对某项任务如何选择压缩方法的建议。

深度神经网络模型压缩方法未来的研究工作会专注于通用性、标准化、压缩率高、精度损失小等模型综合性能提升上，最大化压缩网络规模的同时还能提升网络性能。探索新的模型压缩方法，将多种压缩技术进行结合，通过软硬件协同进行压缩，提高模型运行速度。

6 结束语

本文对近年来深度神经网络模型压缩的主流技术方法进行了概括和总结。模型压缩的主要目标是要在保证准确率的前提下，提高模型压缩率和模型速度。本文总结了网络裁剪、知识蒸馏、轻量级网络设计、量化、体系结构搜索这五个方面的压缩方法。其中，裁剪方法重点在于对网络的冗余权值进行修剪，去掉模型中影响因子较小的结构；知识蒸馏则是通过教师网络的指导去训练一个精简的性能相似的学生网络；轻量级网络设计侧重于设计一个全新的

小型网络,满足移动设备对于深度学习使用的需求;量化方法是目前最直接有效的压缩网络模型的方法,通过对网络中参数进行低秩量化,减小网络存储空间,加快运算速度;体系结构搜索关注深度网络自身的结构性问题,致力于通过搜索算法的设计找到最优子网络,极大地简化网络模型,压缩网络结构。通过本文介绍,读者可以对深度网络模型压缩有一个较为全面的了解,并在今后的研究工作中加以利用,找到新的研究方法。

References:

[1] LeCun Y, Denker J S, Solla S A. Optimalbrain damage[J]. Advances in Neural Information Processing Systems, 1990, 2: 598-605.

[2] Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding[J]. Fiber, 2015, 56(4): 3-7.

[3] Zhang C, Tian J, Wang Y S, et al. Survey of model compression method for neural networks[J]. Computer Science, 2018, 45 (10): 1-5.

[4] Cao W L, Rui J W, Li M. Survey of neural network model compression methods[J]. Application Research of Computers, 2018, 36(3): 649-656.

[5] Luo J, Wu J. An entropy-based pruning method for CNN compression[J]. arXiv:1706.05791, 2017.

[6] Yang T, Chen Y, Sze V. Designing energy-efficient convolutional neural networks using energy-aware pruning[C]// Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Jul 21-26, 2017. Washington: IEEE Computer Society, 2017: 6071-6079.

[7] Hu Y, Sun S, Li J, et al. A novel channel pruning method for deep neural network compression[J]. arXiv:1805.1139419, 2018.

[8] He Y H, Zhang X Y, Sun J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Oct 22-29, 2017. Washington: IEEE Computer Society, 2017: 1389-1397.

[9] Anwar S, Sung W Y. Coarse pruning of convolutional neural networks with random masks[C]//Proceedings of the 2017 International Conference on Learning Representations, Toulon, Apr 24-26, 2017: 134-145.

[10] Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient ConvNets[J]. arXiv:1608.08710, 2016.

[11] Pavlo M, Stephen T, Tero K, et al. Pruning convolutional neural networks for resource efficient inference[J]. arXiv: 1611.06440, 2016.

[12] Hu H, Peng R, Tai Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[J]. arXiv:1611.05128, 2016.

[13] Deepak M, Shweta B, Mitesh M, et al. Recovering from random pruning: on the plasticity of deep convolutional neural networks[J]. arXiv:1801.10447, 2018.

[14] He Y, Liu P, Wang Z, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration [C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Washington: IEEE Computer Society, 2019: 4340-4349.

[15] Yu R, Li A, Chen C F, et al. NISP: pruning networks using neuron importance score propagation[C]//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18-23, 2018. Washington: IEEE Computer Society, 2018: 9194-9203.

[16] Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient DNNs[J]. arXiv:1608.04493, 2016.

[17] Tian Q, Tal A, James J, et al. Deep LDA-pruned nets for efficient facial gender classification[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Jul 21-26, 2017. Washington: IEEE Computer Society, 2017: 512-521.

[18] Cheng Y, Yu F X, Feris R S, et al. An exploration of parameter redundancy in deep networks with circulant projections[C]//Proceedings of the 2015 IEEE International Conference on Computer Vision. Washington: IEEE Computer Society, 2015: 2857-2865.

[19] Hsiao T Y, Chang Y C, Chou H, et al. Filter-based deep-compression with global average pooling for convolutional networks[J]. Journal of Systems Architecture, 2019, 95: 9-18.

[20] Lin S, Ji R, Yan C, et al. Towards optimal structured CNN pruning via generative adversarial learning[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Washington: IEEE Computer Society, 2019: 2790-2799.

[21] Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks[J]. ACM Journal on Emerging Technologies in Computing Systems, 2017, 13(3): 32.

[22] Jin L L, Yang W Z, Wang S L, et al. Mixed pruning method for convolutional neural network compression[J]. Journal of Chinese Computer Systems, 2018, 39(12): 2596-2601.

[23] Huang C, Chang T, Tan H, et al. Neural network pruning

based on weight similarity[J]. Journal of Frontiers of Computer Science and Technology, 2018, 12(8): 1278-1285.

[24] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. Computer Science, 2015, 14(7): 38-39.

[25] Sergey Z, Nikos K. Paying more attention to attention: improving the performance of convolution neural networks via attention transfer[J]. arXiv:1612.03928, 2016.

[26] Wen W, Wu C, Wang Y, et al. Learning structured sparsity in deep neural networks[J]. arXiv:1608.03665, 2016.

[27] Du Y H, Jae Y N, Byoung C K. Estimation of pedestrian pose orientation using soft target training based on teacher-student framework[J]. Sensors, 2019, 19(5): 1147.

[28] Min R, Hai L, Zong J, et al. A gradually distilled CNN for SAR target, recognition[J]. IEEE Access, 2019, 7: 42190-42200.

[29] Xu Z, Song Z Q. Convolution neural network compression method with scale factor[J]. Computer Engineering and Applications, 2018, 54(12): 105-109.

[30] Rastegari M, Ordonez V, Redmon J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks[J]. arXiv:1603.05279, 2016.

[31] Lin X, Zhao C, Pan W. Towards accurate binary convolutional neural network[J]. arXiv:1711.11294, 2017.

[32] Li Z, Ni B, Zhang W, et al. Performance guaranteed network acceleration via high-order residual quantization[J]. arXiv: 1708.08687, 2017.

[33] Liu Z, Wu B, Luo W, et al. Bi-Real Net: enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm[J]. arXiv:1808. 00278, 2018.

[34] Courbariaux M, Bengio Y, David J P. BinaryConnect: training deep neural networks with binary weights during propagations[J]. arXiv:1511.00363, 2015.

[35] Zhu C, Han S, Mao H, et al. Trained ternary quantization [J]. arXiv:1612.01064, 2016.

[36] Xu Y, Dong X, Li Y, et al. A main/subsidiary network framework for simplifying binary neural networks[J]. arXiv: 1812.04210, 2018.

[37] Li F, Zhang B, Liu B. Ternary weight networks[J]. arXiv: 1605.04711, 2016.

[38] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for eifficient integer-arithmetic-only inference[J]. arXiv:1712.05877, 2017.

[39] Sangil J, Son C Y, Seohyung L, et al. Learning to quantize deep networks by optimizing quantization intervals with task loss[J]. arXiv:1808.05779, 2018.

[40] Dong Y, Ni R, Li J, et al. Learning accurate low-bit deep neural networks with stochastic quantization[J]. arXiv:1708. 01001, 2017.

[41] Zhou A J, Yao A B, Guo Y W, et al. Incremental network quantization: towards lossless CNNS with low-precision weights[J]. arXiv:1702.03044, 2017.

[42] Wang Y, Xu C, You S, et al. CNNpack: packing convolutional neural networks in the frequency domain[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41: 2495-2510.

[43] Qin Z D, Zhu D, Zhu X W, et al. Accelerating deep neural networks by combining block-circulant matrices and low-precision weights[J]. Electronics, 2019, 8(1): 78.

[44] Seo S, Kim J. Efficient weights quantization of convolutional neural networks using kernel density estimation based non-uniform quantizer[J]. Applied Sciences, 2019, 9(12): 2559.

[45] Tan W R, Chan C S, Aguirre H E, et al. Fuzzy qualitative deep compression network[J]. Neurocomputing, 2017, 251: 1-15.

[46] Lin J, Gan C, Han S. Defensive quantization: when efficiency meets robustness[J]. arXiv:1904.08444, 2019.

[47] Li Y, Lin S, Zhang B, et al. Exploiting kernel sparsity and entropy for interpretable CNN compression[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Washington: IEEE Computer Society, 2019: 2800-2809.

[48] Wang K, Liu Z, Lin Y, et al. HAQ: hardware-aware automated quantization[J]. arXiv:1811.08886, 2018.

[49] Howard A G, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[J]. arXiv:1704.04861, 2017.

[50] Sandler M, Haward A, Zhu M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18-23, 2018. Washington: IEEE Computer Society, 2018: 4510-4520.

[51] Zhang X Y, Zhou X Y, Lin M X, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18-23, 2018. Washington: IEEE Computer Society, 2018: 6848-6856.

[52] Ma N N, Zhang X Y, Zheng H T, et al. ShuffleNet-V2: practical guidelines for efficient CNN architecture design[C]//

Proceedings of the 15th European Conference on Computer Vision, Munich, Sep 8-14, 2018. Cham: Springer, 2018: 122-138.

[53] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and<0.5 MB model size[J]. arXiv:1602.07360, 2016.

[54] Chollet F. Xception: deep learning with depthwise separable convolutions[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Jul 21-26, 2017. Washington: IEEE Computer Society, 2017: 1251-1258.

[55] Mehta S, Rastegari M, Shapiro L, et al. ESPNetv2: a light-weight, power efficient, and general purpose convolutional neural network[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Washington: IEEE Computer Society, 2019: 9190-9200.

[56] Li X, Long R, Yan J, et al. TANet: a tiny plankton classification network for mobile devices[J]. Mobile Information Systems, 2019(4): 1-8.

[57] Jin X, Yuan X, Feng J, et al. Training skinny deep neural networks with iterative hard thresholding methods[J]. arXiv: 1607.05423, 2016.

[58] Xie S, Girshick R, Dollár P, et al. Aggregated residual transformations for deep neural networks[J]. arXiv:1611.05431, 2016.

[59] Yu Y, Huang J, Du W, et al. Design and analysis of a lightweight context fusion CNN scheme for crowd counting[J]. Sensors, 2019, 19(9): 2013.

[60] Zhu G, Wang J, Wang P, et al. Feature distilled tracking[J]. IEEE Transactions on Cybernetics, 2019, 49(2): 440-452.

[61] Elsken T, Metzen J H, Hutter F. Neural architecture search: a survey[J]. arXiv:1808.05377, 2018.

[62] Barret Z, Quoc V L. Neural architecture search with reinforcement learning[J]. arXiv:1611.01578, 2016.

[63] Fan S, Yu H, Lu D, et al. CSCC: convolution split compression calculation algorithm for deep neural network[J]. IEEE Access, 2019, 7: 71607-71615.

[64] Liu H, Simonyan K, Yang Y. Darts: differentiable architecture search[J]. arXiv:1806.09055, 2018.

[65] Frankle J, Carbin M. The lottery ticket hypothesis: finding sparse, trainable neural networks[J]. arXiv:1803.03635, 2018.

[66] Tan M, Chen B, Pang R, et al. MnasNet: platform-aware neural architecture search for mobile[C]//Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, Jun 16-20, 2019. Washington: IEEE Computer Society, 2019: 2820-2828.

[67] Cai H, Zhu L G, Han S. Proxyless NAS: direct neural architecture search on target task and hardware[J]. arXiv: 1812.00332, 2018.

[68] Bergomi M G, Frosini P, Giorgi D, et al. Towards a topological-geometrical theory of group equivariant non-expansive operators for data analysis and machine learning[J]. Nature Machine Intelligence, 2019, 1: 423-433.

[69] Prost-Boucle A, Bourge A, Petrot F. High-efficiency convolutional ternary neural networks with custom adder trees and weight compression[J]. ACM Transactions on Reconfigurable Technology & Systems, 2018, 11(3): 1-24.

[70] Tran D T, Alexandros I, Moncef G. Improving efficiency in convolutional neural networks with multilinear filters[J]. Neural Networks, 2018, 105: 328-339.

[71] Tan M, Le Q V. EfficientNet: rethinking model scaling for convolutional neural networks[J]. arXiv:1905.11946, 2019.

[72] Zhu S L, Dong X, Su H. Binary ensemble neural network: more bits per network or more networks perbit?[J]. arXiv: 1806.07550, 2018.

[73] Wang X, Kan M, Shan S, et al. Fully learnable group convolution for acceleration of deep neural networks[J]. arXiv: 1904.00346, 2019.

[74] Pham H, Guan M Y, Zoph B, et al. Efficient neural architecture search via parameter sharing[J]. arXiv:1802.03268, 2018.

[75] Baker B, Gupta O, Naik N, et al. Designing neural network architectures using reinforcement learning[J]. arXiv:1611.02167, 2016.

[76] Zhong Z, Yan J, Wu W, et al. Practical block-wise neural network architecture generation[C]//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18-23, 2018. Washington: IEEE Computer Society, 2018: 2423-2432.

[77] Zhong Z, Yan J, Liu C L. Practical network blocks design with Q-learning[J]. arXiv:1708.05552, 2017.

[78] Luo R, Tian F, Qin T, et al. Neural architecture optimization [J]. arXiv:1808.07233, 2018.

[79] Liu C, Zoph B, Neumann M, et al. Progressive neural architecture search[J]. arXiv:1712.00559, 2017.

[80] Kirthevasan K, Willie N, Jeff S, et al. Neural architecture search with Bayesian optimisation and optimal transport[J]. arXiv:1802.07191, 2018.

[81] Zela A, Klein A, Falkner S, et al. Towards automated deep

learning: efficient joint neural architecture and hyperparameter search[J]. arXiv:1807.06906, 2018.

[82] Hsu C H, Chang S H, Liang J H, et al. MONAS: multi-objective neural architecture search using reinforcement learning[J]. arXiv:1806.10332, 2018.

[83] Dong J D, Cheng A C, Juan D C, et al. DPP-Net: device-aware progressive search for pareto-optimal neural architectures[J]. arXiv:1806.08198, 2018.

[84] Cheng Y, Wang D, Zhou P, et al. Model compression and acceleration for deep neural networks: the principles, progress, and challenges[J]. IEEE Signal Processing Magazine, 2018, 35(1): 126-136.

附中文参考文献：

[3] 张弛, 田锦, 王永森, 等. 神经网络模型压缩方法综述[J]. 计算机科学, 2018, 45(10): 1-5.

[4] 曹文龙, 芮建武, 李敏. 神经网络模型压缩方法综述[J]. 计算机应用研究, 2018, 36(3): 649-656.

[22] 靳丽蕾, 杨文柱, 王思乐, 等. 一种用于卷积神经网络压缩的混合剪枝方法[J]. 小型微型计算机系统, 2018, 39(12): 2596-2601.

[23] 黄聪, 常滔, 谭虎, 等. 基于权值相似性的神经网络剪枝[J]. 计算机科学与探索, 2018, 12(8): 1278-1285.

[29] 徐喆, 宋泽奇. 带比例因子的卷积神经网络压缩方法[J]. 计算机工程与应用, 2018, 54(12): 105-109.

GENG Lili was born in 1979. She is a Ph.D. candidate at Taiyuan University of Technology, and working at Shanxi University of Finance and Economics. Her research interests include deep learning and model compression.

耿丽丽（1979—），女，山西阳泉人，太原理工大学博士研究生，任职于山西财经大学，主要研究领域为深度学习，模型压缩。

NIU Baoning was born in 1964. He is a professor and Ph.D. supervisor at Taiyuan University of Technology, and the senior member of CCF, the member of IEEE and ACM. His research interests include big data management and analysis, cloud computing, etc.

牛保宁（1964—），男，山西太原人，太原理工大学信息与计算机学院教授、博士生导师，CCF高级会员，CCF数据库专业委员会委员，IEEE和ACM会员，主要研究领域为大数据管理与分析，云计算等。