轻量级深度神经网络模型适配边缘智能研究综述

徐小华, 周长兵, 胡忠旭, 林仕勋, 喻振杰

引用本文

徐小华, 周长兵, 胡忠旭, 林仕勋, 喻振杰. 轻量级深度神经网络模型适配边缘智能研究综述[J]. 计算机科学, 2024, 51(7): 257-271.

XU Xiaohua, ZHOU Zhangbing, HU Zhongxu, LIN Shixun, YU Zhenjie. Lightweight Deep Neural Network Models for Edge Intelligence:A Survey [J]. Computer Science, 2024, 51(7): 257-271.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100045

轻量级深度神经网络模型适配边缘智能研究综述

徐小华[1,2]　周长兵[1]　胡忠旭[2]　林仕勋[2]　喻振杰[1,3]
1 中国地质大学(北京)信息工程学院　北京 100083
2 昭通学院信息技术教育中心　云南 昭通 657000
3 常州工学院计算机信息工程学院　江苏常州 213000
(10029@ztu.edu.cn)

摘　要　随着物联网和人工智能的迅猛发展,边缘计算和人工智能的结合催生了边缘智能这一新的研究领域。边缘智能具备一定的计算能力,能够提供实时、高效和智能的响应。它在智能城市、工业物联网、智能医疗、自动驾驶以及智能家居等领域都具有重要的应用。为了提升模型的准确度,深度神经网络往往采用更深、更大的架构,导致了模型参数的显著增加、存储需求的上升和计算量的增大。受限于物联网边缘设备在计算能力、存储空间和能源资源方面的局限,深度神经网络难以被直接部署到这些设备上。因此,低内存、低计算资源、高准确度且能实时推理的轻量级深度神经网络成为了研究热点。文中首先回顾边缘智能的发展历程,并分析轻量级深度神经网络适应边缘智能的现实需求,提出了两种构建轻量级深度神经网络模型的方法:深度模型压缩技术和轻量化架构设计。接着详细讨论了参数剪枝、参数量化、低秩分解、知识蒸馏以及混合压缩5种主要的深度模型压缩技术,归纳它们各自的性能优势与局限,并评估它们在常用数据集上的压缩效果。之后深入分析轻量化架构设计中的调整卷积核大小、降低输入通道数、分解卷积操作和调整卷积宽度的策略,并比较了几种常用的轻量化网络模型。最后,展望轻量级深度神经网络在边缘智能领域的未来研究方向。

关键词:边缘智能;深度神经网络;轻量级神经网络;模型压缩;轻量化架构设计
中图分类号　TP183

Lightweight Deep Neural Network Models for Edge Intelligence:A Survey

XU Xiaohua[1,2],ZHOU Zhangbing[1],HU Zhongxu[2],LIN Shixun[2] and YU Zhenjie[1,3]
1 School of Information Engineering,China University of Geosciences(Beijing),Beijing 100083,China
2 Information Technology Education Center,Zhaotong University,Zhaotong,Yunnan 657000,China
3 School of Computer Information Engineering,Changzhou Institute of Technology,Changzhou,Jiangsu 213000,China

Abstract　With the rapid development of the Internet of Things(IoT) and artificial intelligence(AI),the combination of edge computing and AI has given rise to a new research field called edge intelligence. Edge intelligence possess appropriate computing power and can provide real-time,efficient,and intelligent responses. It has significant applications in areas such as smart cities,industrial IoT,smart healthcare,autonomous driving,and smart homes. In order to improve the accuracy of models,traditional deep neural networks often adopt deeper and larger architectures,resulting in significant increases in model parameters,storage requirements,and computational complexity. However,due to the limitations of IoT terminal devices in terms of computing power,storage space,and energy resources,deep neural networks are difficult to be directly deployed on these devices. Therefore,lightweight deep neural networks with low memory,low computational resources,high accuracy,and real-time inference capability have become a research hotspot. This paper first reviews the development process of edge intelligence and analyzes the practical requirements for lightweight deep neural networks to adapt to intelligent terminals. Two methods for constructing lightweight deep neural network models:model compression techniques and lightweight architecture design are proposed. Next,it discusses in detail five main model compression techniques:parameter pruning,parameter quantization,low-rank decomposition,knowledge

到稿日期:2024-01-02　返修日期:2024-04-27
基金项目:地球科学文献知识服务与决策支撑二级项目(DD20230139);云南省地方本科高校基础研究联合专项-面上项目(202301BA070001-095)
This work was supported by the China Geological Survey(CGS) Work Project:Geoscience Literature Knowledge Services and Decision Supporting(DD20230139) and Special Basic Cooperative Research Programs of Yunnan Provincial Undergraduate Universities Association(202301BA070001-095).
通信作者:周长兵(zbzhou@cugb.edu.cn)

distillation, and mixed compression techniques. It summarizes their respective performance advantages and limitations, and evaluates their compression effects on commonly used datasets. Then, the paper analyzes in depth the strategies of adjusting the size of the convolution kernel, reducing input channel number, decomposing convolution operations, and adjusting convolution width in lightweight architecture design, and compares several commonly used lightweight network models. Finally, the future research direction of lightweight deep neural networks in the field of edge intelligence is prospected.

Keywords Edge intelligence, Deep neural networks, Lightweight neural network, Model compression, Lightweight architecture design

1 引言

随着物联网(Internet of Things, IOT)的快速发展和5G网络的普及,物联网的终端设备数量和产生的数据都呈爆炸式的增长。据IDC估计,到2025年,全球将有416亿台物联网设备,并且产生79.4 ZB的数据[1],其中近一半的物联网数据需要在边缘设备上进行处理[2]。云计算架构在处理物联网边缘设备产生的海量数据时显示出局限性,如无法承受的网络负载、数据隐私隐患、实时响应挑战,以及高能耗成本等问题逐渐暴露出来。为解决以上问题,在以云计算为核心的基础上,提出了边缘计算。边缘计算的模型有3种:1)内容分发网络(Content Delivery Network, CDN),它是一种分布式网络基础设施,通过在多个地理位置上放置缓存服务器来优化内容的交付速度和可靠性,加速用户访问网站和网上资源[3];2)雾计算(Fog Computing),它是一种分散式计算架构,将计算、存储和网络服务带到网络的边缘,更接近数据源[4];3)微云(Cloudlet),它是一种小型的、移动性支持的边缘服务器,为靠近最终用户的移动设备提供强大的计算资源[5]。从边缘计算模型可以看出,一些原本在云中进行的计算逐渐转移到了边缘。边缘计算的主要思想是"计算应该更靠近数据源、更贴近用户"。整个系统被划分为云、边缘和端3层,边缘计算位于云和端设备之间[6-7]。在边缘计算中,数据处理和计算任务被分配到边缘设备上,使得数据处理更加高效和实时。

近年来,深度神经网络(Deep Neural Networks, DNNs)等深度学习算法在计算机视觉、语音识别、自然语言处理和专家系统等人工智能相关应用领域取得了显著突破。与此同时,人工智能的应用正逐渐转向边缘计算,促使边缘计算与人工智能深度融合。这种融合产生了一个新的研究领域,即边缘智能(Edge Intelligence, EI),其正受到学术界和工业界的广泛关注[8]。边缘智能是一种技术范式,指人工智能算法直接在边缘设备上运行处理数据,而不需要将数据传输到云端或数据中心进行处理[9-10]。"边缘"可以是连接到网络的任何设备,如手机、传感器、智能家居设备、智能手表、智能摄像头或其他物联网设备等[11],这些设备通常位于数据源附近。在物联网中,边缘智能已成为一种趋势。据Gartner预测,超过80%的企业物联网项目将包含人工智能组件[12]。边缘智能具备一定的计算能力,并能够提供实时、高效和智能的响应,在智能城市、工业物联网、智能医疗、自动驾驶等领域都具有重要的应用。

为了提高模型精度,传统的深度神经网络倾向于使用更深层和更大的架构,直接导致了模型参数数量的增加,以及较高的存储需求和复杂的计算量。例如,仅5层卷积层和3个全连接层的AlexNet[13]大约需要6 100万个参数,占用233 MB存储空间,并需要进行7.1亿次浮点型运算(Floating-point Operations, FLOPs);13个卷积层和3个全连接的VGG-16[14]拥有接近1.38亿个参数,需要527 MB的存储空间,并要进行154.7亿次FLOPs;而16个卷积层和3个全连接的VGG-19[15]参数数量约为1.43亿,存储需求为548 MB,并需要进行196.3亿次FLOPs;相对偏浅的ResNet-8[16]包含约0.12亿个参数,占用44.59 MB的存储空间,需完成18.2亿次FLOPs;而较深的ResNet-152[17]则约有0.6亿个参数,占用229.62 MB存储空间,并需完成116亿次FLOPs。目前,配备GPU的高性能边缘服务器和内置移动处理器等部分智能设备已经能够有效地运行深度神经网络。然而,在物联网领域,大多数边缘设备的计算能力、存储能力以及能源等资源都十分有限,例如,一些搭载低功耗CPU的嵌入式设备和片上存储容量仅为几兆到几十兆字节的较低端设备(如ATmega328P),使得将深度神经网络部署到这些边缘设备变得十分困难,如图1所示。近年来,学术界和工业界格外关注具备低内存占用、低计算需求、高准确度和实时推理能力的轻量级深度神经网络,轻量级神经网络适配边缘智能成为了当前的一个研究热点。

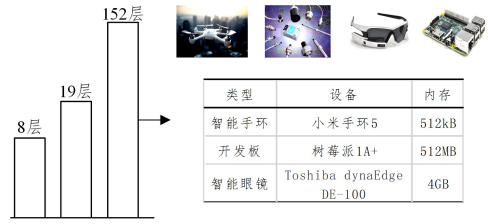

图1 挑战:在边缘环境实现深度神经网络的有效部署

Fig. 1 Challenges: effective deployment of deep neural networks in edge environments

本文系统地介绍了轻量化神经网络适配边缘智能的进展。第1章回顾了边缘智能的发展历程,并分析了轻量级深度神经网络适应智能的现实需求,提出了两种构建轻量级深度神经网络模型的方法:深度模型压缩技术和轻量化架构设计;第2章详细讨论了参数剪枝、参数量化、低秩分解、知识蒸馏以及混合压缩5种主要的深度模型压缩技术,归纳了它们各自的性能优势与局限,并评估了它们在常用数据集上的压缩效果;第3章主要分析了轻量化架构设计中的调整卷积核大小、降低输入通道数、分解卷积操作和调整卷积宽度的策略,并比较了几种常用的轻量化网络模型;第4章探讨了轻量级深度神经网络在边缘智能领域的未来研究方向。

2 深度模型压缩

在深度神经网络模型中，通常包含卷积层（Convolutional Layer，Conv. Layer）、池化层（Pooling Layer）和全连接层（Fully Connected Layer，FC Layer）。这些层中存在大量参数，其中的一些参数可能是冗余的。采用不同的压缩技术对冗余参数进行调整，能有效减少参数量、降低计算复杂度和加速推理，使模型更适配在资源受限的边缘设备。

2.1 参数剪枝

参数剪枝是深度模型压缩技术中最常采用的策略之一。它通过移除对模型性能影响较小的冗余参数和连接来减少模型的参数量，并利用稀疏矩阵进行存储和计算。根据剪枝操作的具体粒度，参数剪枝被分为非结构化剪枝和结构化剪枝两种类型[18]。非结构化剪枝粒度非常细，专注于单个权重，剔除那些数值接近于零的权重。非结构化剪枝后的网络结构很可能变得不规则，要处理这类稀疏数据通常需要额外的硬件支持。结构化剪枝的粒度相对较大，它在网络的结构层面进行剪枝。采用结构化剪枝后，网络的拓扑结构会保持整洁，使得剪枝后的模型更容易在硬件上实现。然而，在实践中，两种技术也可以结合使用，即可以先应用结构化剪枝进行大规模的降维，然后用非结构化剪枝进行微调，以最大程度地提升模型性能和效率。非结构化剪枝和结构化剪枝是两种常见的剪枝技术，它们各自有不同的优势和劣势，如表 1 所列。

表 1 结构化剪枝和非结构化剪枝的对比

Table 1 Comparison between structured pruning and unstructured pruning

Methods	Advantages	Disadvantages
Unstructured Pruning	细粒度剪枝、灵活性强、压缩高且性能损失小	加速依赖特殊硬件和软件，不一定能提高推理速度
Structured Pruning	易于硬件加速、减少内存占用与加速推理	剪枝的粒度较粗，可能降低模型性能，需要精细调优

Han 等[19]通过非结构化剪枝、权重聚类和霍夫曼编码的 3 个过程，在几乎没有损失精度的条件下，将 AlexNet 和 VGG-16 分别压缩到原来的 1/49 和 1/35，通过在 CPU、GPU 和移动 GPU 上进行基准测试，速度提高了 3～4 倍，能效提高了 3～7 倍，取得了深度学习模型极致化的压缩效果。Molchanov 等[20]通过 Taylor 展开式，用于评估卷积神经网络中各个权重的重要性，从而实现神经网络的剪枝。Dong 等[21]提出了基于神经架构搜索通道和层参数的剪枝方法。在早期的结构化神经网络剪枝研究中，剪枝率往往是手动指定的，这会导致在维持模型精度的同时，难以确定最佳的剪枝率。为了解决这一问题，Sakai 等[22]提出了一种利用梯度和损失函数自适应确定每层剪枝率的方法，从而改善了剪枝决策过程。在多任务学习环境中，Choi 等[23]利用结构化剪枝技术在不同任务间共享特征，通过联合稀疏约束来促进，有效地压缩了深度学习模型的规模。目前，剪枝研究的重点已经转向自动化剪枝。例如，Wang 等[24]提出一种通过 actor-critic 结构自动压缩深度神经网络模型的框架，该框架与神经网络中的每一层进行相互作用。在这个框架中，actor 网络负责决定压缩策略，而 critic 网络则通过预测值来确保 actor 网络的决策准确性，从而提高网络的压缩质量。Zhao 等[25]提出了针对组卷积的高效结构化剪枝和架构搜索技术。Xu 等[26]将剪枝问题视为组合优化问题，并设计出层自适应的剪枝方案。此外，Anwar 等[27]引入了结构稀疏性和粒子滤波技术相结合的网络剪枝技术，取得了在实现显著节约计算资源的同时保持模型精度的成果。Louati 等[28]利用双层进化策略优化网络结构和剪枝策略，实现了既轻量化又高精度的神经网络架构。Xia 等[29]通过联合剪枝粗粒度和细粒度的模块，控制每个参数的剪枝决策，从而生成高度可并行化的子网络，在保持准确性的情况下显著减小模型大小，实现超过 10 倍的加速。Eccles 等[30]设计了一种端到端的深度神经网络训练、空间剪枝和模型切换 DNN Shifter 系统，通过结构化剪枝生成更小、更快速的模型变体，适用于边缘计算设备并保持接近原始模型的准确性。Sun 等[31]设计了 SPSRC 框架，通过重新组织卷积操作并利用转换后矩阵的奇异值和矩阵范数作为显著性评分，实现了对卷积核权重的更好利用，从而改进了通道结构化剪枝方法并显著提升了性能。Qian 等[32]提出了基于傅里叶空间的鲁棒剪枝算法（FSRP），提升模型在边缘设备上的实时性和鲁棒性。Basha 等[33]提出了基于网络训练历史的滤波剪枝（HBFP）方法。

2.2 参数量化

参数量化是通过减少参数位数来控制计算和存储成本，用低精度数值格式来代替高精度格式。由表 2[34]可得，低精度定点操作数的硬件面积大小和能耗比高精度浮点数少。将深度神经网络的参数从 FP32 转为低精度格式（如 FP16 和 Int8），不仅减少了模型对存储空间的需求，加速了推理过程，还有助于降低边缘智能设备的能耗。

表 2 不同精度定点操作数所需能耗和硬件面积

Table 2 Energy consumption and hardware area required for different precision fixed-point operands

Operation	Energy/p	Area/μm^2
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3,1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read(8kB)	5.0	—
32b DRAM Read	640	—

根据不同的校准方法，量化可分为均匀量化、非均匀量化以及对称量化、非对称量化。均匀量化函数如下：

$$Q(r)=\mathrm{Int}\left(\frac{r}{S}\right)-Z \tag{1}$$

其中，Q 为量化操作，r 是输入浮点数的激活值或权重值，S 是

实值比例因子，Z 是零值偏移量，Int 函数通过四舍五入将一个实值映射为一个整数值。$Q(r)$的实质是一个实值 r 到一些整数的映射。

非均匀量化指量化值不一定均匀间隔，其量化函数定义：

$$Q(r)=X_i, r\in[\Delta_i,\Delta_{i+1}) \tag{2}$$

即当一个实数 r 的值处于量化步骤Δ_i和Δ_{i+1}之间时，量化器 Q 将其映射到相应的量化级别X_i，Δ_i和Δ_{i+1}的间隔都不是均匀的。在式(1)中，有：

$$S=\frac{\beta-\alpha}{2^b-1} \tag{3}$$

S 将一个给定的实值范围 r 划分为若干区域。其$[\alpha,\beta]$表示裁剪范围，b 是量化位宽。当 α 与 β 取真实值 r 的最小值 $r_{\min}$和$r_{\max}$时，若$-\alpha$ 与 β 不相等，此时量化为非对称量化；若$-\alpha$ 与 β 相等，此时量化为对称量化。

在边缘智能设备中，特别是在微控制器方面，由于硬件能力的限制，很多边缘处理器(如 ARM Cortex-M0/M0+和 Atmel ATmega 系列)不支持硬件级的浮点运算。因此，常常需要将浮点数量化为 16 位、8 位，有时甚至用更低的纯整型位数表示。在模拟量化中，量化后的模型参数以低进度存储，但卷积中的运算以浮点数运算进行，因此需要进行反量化。而在纯整型位数量化中，所有的操作都是使用低精度的整数运算，不需要对参数或激活值进行浮点反量化，其流程如图 2 所示。

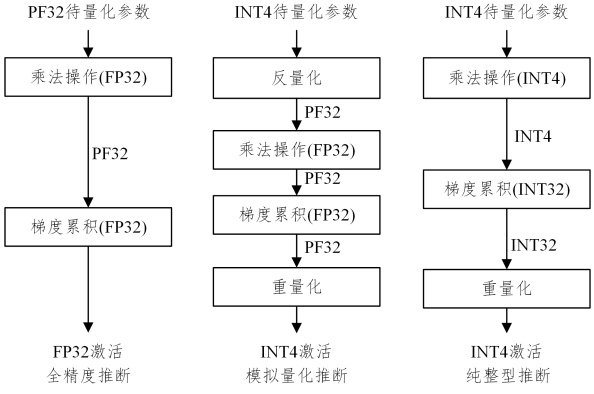

图 2 不同推断模型示意图

Fig. 2 Schematic diagram of different inference models

量化技术中一个极端的例子是二值量化，它将深度神经网络的参数和激活值从浮点数转换为只有两个可能的值，通常表示为+1 或−1。Rastegari 等[35]使用二值量化近似深度神经网络的滤波器，可以实现高达 32 倍的内存节省。此外，Vorontsov 等[36]结合最新的持续学习技术和二值神经网络，能够高效地在低功耗嵌入式 CPU 上执行深度学习模型。三值量化，也称为三级量化，将参数量化为 3 个可能的值：+1，0 和−1。Liu 等[37]提出了三元权重约束的方法，成功实现了三级参数表示。这种方法不仅有助于缩减深度神经网络的规模，以便适配到边缘设备，还能提高模型在特定任务上的准确性。Razani 等[38]提出了一种自适应的二值和三值量化组合方法，即智能量化(Smart Quantization，SQ)，通过正则化函数直接调整量化深度，从而使模型仅需进行一次训练。Chen 等[39]将量化方法作为一种正则化策略，可以减少深度神经网络的规模以适应边缘设备和提升模型的准确性。Tmamna 等[40]提出了基于粒子群优的量化方法来压缩卷积神经网络。

混合精度量化旨在解决使用较低精度量化时可能引起的精度显著下降问题，其核心任务是在考虑量化和网络层之间关系的基础上，为各个网络层分配合适的量化位宽，如图 3 所示。Schaefe 等[41]提出了针对边缘智能的混合精度卷积神经网络量化方法，通过建立新的帕累托前沿在模型准确性和内存占用之间平衡，展示了预训练的量化模型。Zhang 等[42]设计了基于 Zynq SoC 的混合精度神经网络加速器，推断速度提升了 4.96 倍。Li 等[43]采用对称逐渐降低的混合精度量化方法，为不同层级的网络块分配不同位宽，以在低位量化的情况下获得高性能分割准确性，显著降低模型规模、内存占用和计算成本，适用于边缘设备部署。Wang 等[44]提出了智能物联网终端资源自适应混合精度模型量化方法，在相同的压缩比下，与统一精度量化方法相比，该方法具有更高的准确率。Huang 等[45]提出了 MXQN 方法，通过固定点和对数对深度卷积神经网络进行量化，无需重新训练和微调即可实现高准确性。Kundu 等[46]使用比特梯度来分析层敏感性并产生混合精度量化模型的训练方法 BMPQ 模型，只需进行单次训练迭代。Louizos 等[47]提出了可微分量化。

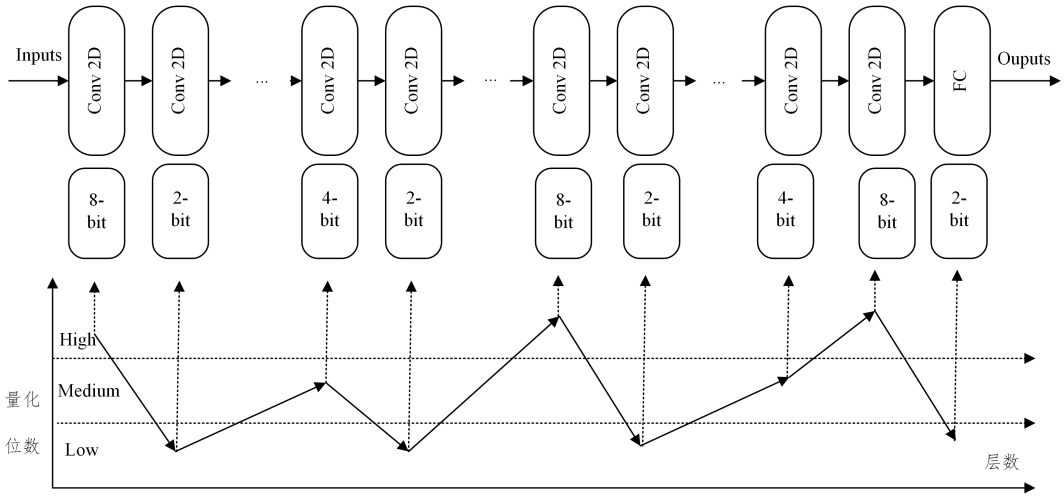

图 3 混合精度量化

Fig. 3 Mixed precision quantization

参数量化在深度神经网络中通过降低模型权重的数值精度来减小模型尺寸和加速推理，有助于节约存储空间、减少计算资源需求和降低功耗，使其更适合边缘智能设备，但这一过程可能会损失模型精度，并且可能受限于硬件平台和软件框架对低精度运算的支持。

2.3 低秩分解

矩阵的秩用来度量矩阵行列间的相关性，当矩阵的秩远小于行和列的值时，则矩阵为低秩矩阵，低秩矩阵具有大量的冗余信息。深度神经网络模型中的参数矩阵同时具备低秩和稀疏的性质。低秩分解通过将神经网络模型的参数矩阵分解成两个或更多较小的矩阵乘积的方式，减少了矩阵乘法操作的计算量、模型的参数量和存储空间，进而提高了模型的推理速度和训练速度。其代表方法有 SVD[48] 分解、Tucker[49] 分解、CP 张量分解[50] 等。SVD 的权值压缩技术[48] 是在两个相邻层 L 和 $L+1$ 层之间引入一个中间层 L'，L' 中含有 K 个隐单元，从而压缩权值 $\boldsymbol{W}$。SVD 将相邻两层的权值矩阵 $\boldsymbol{W}^{L+1}_{A\times B}$ 分解成两个分量，通过 K 来进一步地与矩阵的特征数目保持一致，以此保证 $\boldsymbol{W}^{L+1}_{A\times B}$ 可以用 $\boldsymbol{W}^{L+1}_{A\times K}\boldsymbol{W}^{L'}_{K\times B}$ 进行计算。其 L，L' 与 $L+1$ 这 3 层之间的映射关系如下：

$$\begin{aligned}\boldsymbol{y}&=\boldsymbol{W}^{L+1}_{A\times B}X+\boldsymbol{b}\\&=(\boldsymbol{W}^{L+1}_{A\times K}\boldsymbol{W}^{L'}_{K\times B})X+\boldsymbol{b}\\&=\boldsymbol{W}^{L+1}_{A\times K}(\boldsymbol{W}^{L'}_{K\times B}X)+\boldsymbol{b}\end{aligned}\tag{4}$$

其中，$\boldsymbol{y}'=\boldsymbol{W}^{L'}_{K\times B}X$ 是 L 与 L' 进行无偏移连接，L' 与 $L+1$ 之间的映射关系是 $\boldsymbol{y}=\boldsymbol{W}^{L+1}_{A\times K}\boldsymbol{y}'+\boldsymbol{b}$。压缩后，权值 W 和计算量都将降低。

针对权值矩阵稀疏性的思想和神经元对实现目标存在不平等的贡献，Swaminathan 等[51] 提出了一种稀疏低秩方法，以实现更好的压缩率。Bao 等[52] 提出了先验和总变差正则项的低秩分解模型，提高了织物缺陷检测的准确性和性能。Cheng 等[53] 提出了求全局最优解的低秩张量分解算法，用于消除卷积核中的冗余，以此来加速大型卷积神经网络，并显著提升了性能，更方便在边缘移动设备上的部署。Xiao 等[54] 提出了可区分硬件感知的自动低秩压缩框架，通过端到端的方案，在不同数据集和硬件平台上显著提升了压缩效果和模型精确度。Idelbayev 等[55] 提出了一种将每一层的权重矩阵逼近为低秩矩阵的量化方法，实现了更高效的神经网络压缩。

低秩分解在减少模型参数和提高计算效率方面有着不错的效果。但近几年来已不再流行，主要是因为神经网络架构使用较多的 1×1 卷积层，使在使用低秩分解方法时存在一定的困难。

2.4 知识蒸馏

知识蒸馏是将一个大型、复杂的“教师”模型的知识迁移给一个更小、更简单的“学生”模型。该方法通过富含知识的“教师”模型和不同的训练机制，将知识从“教师”模型蒸馏到“学生”模型上，使得“学生”模型高效地学习到来自数据和“教师”模型传递的知识。通过这种方式，学生模型不仅能够提升准确度，而且能同时保持其轻量化的特点，还能减少智能设备在存储和计算资源方面的开销[56-58]。

Hinton 等[59] 于 2015 年首次提出知识蒸馏的概念，通过教师网络产生的 logits（最终 softmax 的输入）与学生网络的 logits 之间的差异，将知识从教师网络迁移到学生网络。Hinton 等在 softmax 层引入了一种 softmax 的温度系数：

$$q_i=\frac{\exp\left(\frac{z_i}{T}\right)}{\sum j\exp\left(\frac{z_j}{T}\right)}\tag{5}$$

其中，q_i 表示第 i 类的输出概率；z_i 和 z_j 表示 softmax 层的输入；T 为温度系数，用于控制输出的平缓程度，T 越高，softmax 层映射曲线越平缓。获得“教师”模型之后，在 softmax 引入温度系数，使用数据集和软标注训练“学生”模型，“学生”模型 softmax 层的温度系数与教师一致，训练结束后，温度系数 $T=1$。若数据集已有标注，则损失函数由两部分组成：

$$L=L_{\text{hard}}+\lambda L_{\text{soft}}\tag{6}$$

通过上述损失函数，学生模型实现了从数据集和“教师”模型获取知识的过程，从而提升了精度，减轻了对大量数据的依赖。基于数据集和“教师”模型的知识蒸馏结构如图 4 所示。

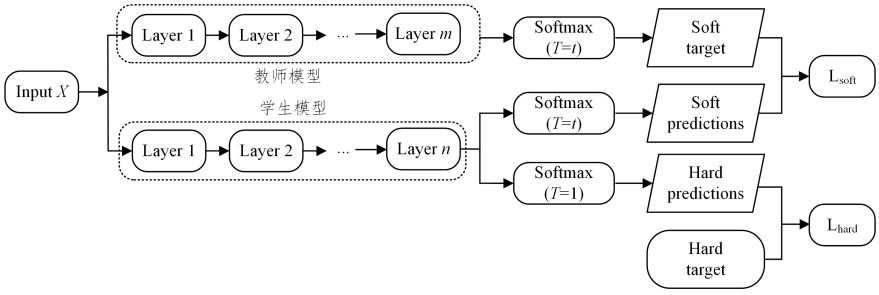

图 4 基于数据集和“教师”模型的知识蒸馏结构图

Fig. 4 Structure of knowledge distillation based on a dataset and a “teacher” model

在知识蒸馏中，Romero 等[60] 提出了 FitNets 方法，引入了中间层对齐的概念，以提高学生模型的性能。Zagoruyko 等[61] 提出了 Attention Transfer 方法，利用教师模型的注意力机制来提升学生模型的性能。随后又提出了多教师蒸馏方案，它是用具有不同结构或训练策略的多个教师模型来引导学生模型的训练。Zhang 等[62] 提出了可信度感知的多教师知识蒸馏（CA-MKD），将多个教师的知识源灵活地整合到学生模型训练中。然而，为了防止一些不成熟的教师可能会向学生传递错误的知识，Choi 等[63] 提出了一种在线角色转换策略，通过将多个网络分为教师组和学生组，并在每次迭代中将学生组中排名靠前的网络晋升为教师组，来实现知识蒸馏的有效性。传统的对抗训练方法对于精简模型来说并不能很好地提供鲁棒性，Qian 等[64] 提出了两阶段对抗知识蒸馏方法，使精简后的模型性能得到有效的提升，以便在计算资源

有限的边缘设备上部署鲁棒的精简深度学习模型。Mishra等[65]使用知识蒸馏中的提前停止技术,在边缘设备上设计和训练轻量级神经网络。Gou等[66]提出了信息协同的知识蒸馏方法。Li等[67]提出了通过知识蒸馏从少样本中实现数据和训练效率的方法。Pham等[68]利用频率注意机制来增强知识蒸馏模型。

在大部分的知识蒸馏方法中,具备明确的"教师"和"学生"的模型身份划分,因此Zhang等[69]提出了"互学习"的训练机制,使用多个学生模型之间的相互学习来提高整体性能。在这种机制下,并不存在明显的"教师"身份模型,而每一个"学生"模型都相当于彼此的"同伴教师",以一种互帮互助的模型进行训练。"学生"网络在训练中相互学习、相互监督。Huang等[70]设计了一种基于多阶段多生成对抗网络(MS-MGANs)的互学习知识蒸馏方法,用不同阶段的老师模型逐步指导学生模型,以获得更好的精度效果。

一般来说,"教师"模型和"学生"模型的结构差异过大时,知识蒸馏的效率会下降。Mirzadeh等[71]提出了分层知识蒸馏方法,该框架引入"助教模型"来桥接教师和学生之间的差距。该方法包含了3个部分的模型:性能强大的教师模型、能力介于教师和学生之间的助教模型和能力较小的学生模型。"助教模型"知识蒸馏的基本步骤如下:首先"助教"从"教师"模型中提取新知识,然后"助教"充当"老师",以"助教相辅"的模式再将知识传给"学生"。相比直接从教师蒸馏到学生,助教模型作为中间层可以提高蒸馏效果,帮助学生模型学习到更细粒度的知识和模型行为,使得知识蒸馏变得更为灵活。

2.5 混合压缩模式

参数剪枝、参数量化、低秩分解和知识蒸馏等深度模型压缩技术,在单独应用时都能够显著提高模型效率并取得不错的效果,但同时它们也有各自的局限性。为此,研究人员们提出了混合压缩模式,意在整合各种模型压缩技术的优点。混合压缩方法将会是轻量化神经网络适配边缘智能的重要研究方向。

Kwon等[72]结合权重加密、非结构化剪枝和参数量化技术,可以在不损失模型性能的情况下实现对神经网络的压缩和优化。Qu等[73]通过权重动态量化减小通信成本,结合知识蒸馏方法加速模型训练,在边缘设备上实现高效的联邦学习。Chang等[74]根据通道的概率分布特征,逐层修剪多余的通道并确定混合精度量化的比特宽度,以实现高效的边缘和移动计算硬件电路。Bai等[75]结合剪枝Pruning和量化Quantization技术,可以在无需微调的情况下实现模型压缩。Wang等[76]将利用泰勒展开式对网络的参数剪枝和知识蒸馏相结合的方法应用于模型压缩,实现轻量级模型在配电系统中的应用。Yu等[77]提出了可微的联合剪枝和量化方案,将神经网络压缩构建为基于梯度的优化问题,自动在模型剪枝和量化之间进行权衡,以实现硬件效率。Kim等[78]提出了一种针对计算资源有限的边缘设备的新型模型压缩方法,称为PQK,包括修剪、量化和知识蒸馏过程。

2.6 深度模型压缩效果比较

从上文介绍的4种技术中选出一些代表性的压缩技术,并按照文献中的方法对压缩效果进行对比。表3—表6中,Δaccuracy等于压缩后模型的accuracy与原始模型accuracy的差值;Params等于压缩前后参数的比值;FLOPs等于压缩前后FLOPs的比值。在知识蒸馏中,accuracy是学生模型的准确度与教师模型准确度的差值;params是学生模型的参数与教师模型的参数的比值。

表3列出了一些主流的剪枝方法在CIFAR-10数据集上使用VGG-16的压缩效果。从整体来看,accuracy下降程度不大,几乎没有变化。文献[30]在准确度略有提高的前提下,压缩比相对较好,文献[22]达到了最高的压缩比,但accuracy有所下降。

表3 不同的剪枝方法在CIFAR-10数据集上利用VGG-16的压缩效果

Table 3 Compression effects of different pruning methods using VGG-16 on CIFAR-10 dataset

Method	ΔAccuracy/%	Params	FLOPs
Ref.[22]	−0.48	26.3X	6.99X
	−0.35	17.86X	5.85X
Ref.[26]	+1.05	4.2X	2.82X
Ref.[30]	+0.4	14.43X	1.67X
Ref.[31]	+0.39	5.32X	1.72X
Ref.[33]	−0.92	3.57X	3.45X
	−1.42	4.27X	4.17X

表4列出了一些主流的量化方法在CIFAR-10数据集上使用VGG-16的压缩效果。从整体来看,accuracy下降略大。但文献[40]的准确度提高0.53%,它的压缩比达到了10%,效果最好;文献[46]虽然准确度略有下降,但压缩前模型是压缩后模型的10.5倍和15.4倍,内存访问速度也提高了很多;文献[45]虽然accuracy下降,但不需要量化后再训练。

表4 不同的量化方法在CIFAR-10数据集上利用VGG-16的压缩效果

Table 4 Compression effects of different quantization methods using VGG-16 on CIFAR-10 dataset

Method	ΔAccuracy/%	Weight bit	Activation bit
Ref.[38]	−0.15	Mix	32
Ref.[40]	+0.53	Mix	Mix
Ref.[45]	−0.12	9	6
Ref.[46]	−0.34	Mix	16
Ref.[46]	−0.69	Mix	16

表5列出了一些有代表性的知识蒸馏方法在MNIST,GLUB和CIFAR-100数据集上的压缩效果。与其他方法对比,文献[60]Accuracy下降得更多,压缩比更小。

表5 不同知识蒸馏方法的压缩效果

Table 5 Compression effects of different knowledge distillation methods

Dataset	Method	Accuracy	Params
MNIST	Ref.[60]	+0.04	12X
CIFAR-100	Ref.[60]	+1.42	3.6X
GLUB	Ref.[62]	−4.40	2.4X
CIFAR-100	Ref.[63]	+0.15	3.5X
CIFAR-100	Ref.[68]	−0.19	2.8X

表6列出了参数剪枝、参数量化、低秩分解、知识蒸馏和混合压缩5种深度模型技术的一些代表性方法在不同的数据集和不同模型上的效果对比。从表中可以看出,不同的压缩效果与数据集和模型选择有关。

表 6 不同压缩方法的压缩效果

Table 6 Compression effects of different compression methods

Classification	Dataset	Model	Method	ΔAccuracy	Params	FLOPs
Parameter Pruning	CIFAR-10	ResNet-110	Ref. [22]	−0.59	2.56X	2.57X
	CIFAR-10	VGG-16	Ref. [26]	+1.05	4.2X	2.82X
Parameter Quantization	CIFAR-100	ResNet-32	Ref. [40]	+1.33	8.33X	−
	Tiny-ImageNet	ResNet-18	Ref. [46]	−0.88	8.8X	−
Low-Rank Compression	CIFAR-10	ResNet-20	Ref. [54]	+0.07	4.18X	3.59X
	CIFAR-10	VGG-16	Ref. [54]	+1.91	4.18X	5.62X
	CIFAR-10	AlexNet	Ref. [55]	−1.12	4.06X	4.3X
Hybrid Model Compression	CIFAR-10	VGG-16	Ref. [74]	−0.15	10.2X	−
	CIFAR-10	VGG-16	Ref. [75]	−1.44	12X	5X
	Power-Services	ResNet50	Ref. [76]	+2.24	5.24X	12.66X

由表 3—表 6 可以看出，在保证精度的前提下，自适应的自动化剪枝方法和混合精度量化方法都取得了很好的压缩效果，特别是混合方法在压缩比方面表现突出。此外，知识蒸馏作为一种模型压缩方法，仍然具有巨大的发展潜力，未来有望在模型压缩领域发挥更大作用。不同的压缩方法各有其优缺点，具体如表 7 所列。在实际应用中，可以根据具体场景和需求选择合适的方法进行模型压缩来适配资源受限的边端。

表 7 深度神经网络模型压缩方法的总结

Table 7 Summarization of DNNs compression methods

Method	Description	Applied to layers
Parameter Pruning	根据阈值判断参数重要性，删除不重要的参数	卷积层、全连接层
Parameter Quantization	32 位浮点数转换为低位宽度的整数	所有层
Parameter Pruning	将矩阵分解为多个低秩矩阵的乘积形式	卷积层、全连接层
Knowledge distillation	将一个大型的、复杂的教师模型的知识迁移到一个小型的学生模型	所有层
Hybrid Model Compression	组合使用前几种方法	视组合情况而定

3 轻量化的网络架构设计

构建轻量级神经网络来适配资源受限边缘智能的另外一种重要方法是设计轻量化架构。本章将从设计轻量化架构的相关的卷积运算、调整卷积核大小、降低输入通道数、分解卷积运算、调整卷积宽度等方面进行详细阐述。

3.1 卷积层的相关运算

在$k_h \times k_w$卷积中，C_i和C_o分别表示输入特征图通道数和输出特征图通道数，若步长(Stride)为 1 和相同(Same)填充时，考虑偏置运算，则卷积核参数的计算式如下：

$$params = C_o \times (C_i \times k_h \times k_w + 1) \tag{7}$$

特别地，在 $k \times k$ 卷积下，若不考虑偏置，则：

$$params = C_o \times (C_i \times k^2) \tag{8}$$

在卷积网络中，通常将一个“乘加”组合视为一次 FLOP。在$k_h \times k_w$卷积中，若步长(Stride)为 1 和相同(Same)填充时，考虑偏置运算，则：

$$FLOPs = C_i \times C_o \times H \times W \times k_h \times k_w \tag{9}$$

其中，$H \times W$ 为特征图大小。特别地，在 $k \times k$ 卷积下，有：

$$FLOPs = C_i \times C_o \times H \times W \times k^2 \tag{10}$$

3.2 调整卷积核大小

在卷积神经网络早期发展阶段，为确保特征提取效果，许多网络采用了较大尺寸的卷积核。例如，Krizhevsky 等[79]在 AlexNet 网络中使用 11×11 和 5×5 的卷积核，在 ImageNet 图像分类挑战中取得了巨大的成功，使得 AlexNet 一举夺得了冠军，它成功推动了深度学习快速发展。但随着后来大量模型的实验，研究表明使用较大尺寸的卷积核会增加浮点运算的数量，从而引起推理过程的延迟和设备能耗的增加，并且需要更多的存储空间。因此，近年来神经网络在卷积层中较少使用大尺寸的卷积核。

神经网络中常用的卷积核有 11×11，7×7，5×5，3×3 和 1×1。在调整卷积核大小时，常用的小卷积核为 3×3。Simonyan 等[80]在 VGG 网络中采用了一种替代方法，将 7×7 输入特征图使用三层连续 3×3 卷积核代替 7×7 卷积核，从而得到输出特征图的尺寸都为 1×1，且感受野大小都为 7。相比使用单个 7×7 卷积核，这种方法减少了参数量，因为一个 7×7 卷积核有 49 个参数，而 3 层 3×3 卷积核共有 27 个参数，参数量减少约 44.9%。同样，对于 5×5 的输入特征图，使用连续两层 3×3 卷积核来代替 5×5 卷积核，得到的输出特征图和感受野大小都相同，而相比参数量减少了 28%。输入特征是 $H \times W \times C_i$时，输出特征图 $H \times W \times C_o$，对于 5×5 卷积核的 $FLOPs = C_i \times C_o \times H \times W \times 5^2$，而两层 3×3 卷积核的 $FLOPs = C_i \times C_o \times H \times W \times 2 \times 3^2$，很明显 FLOPs 也减少 28%。上述结果说明在相同大小的感受野条件下，大卷积用小卷积代替，小卷积的参数量更少和 FLOPs 更小，同时小卷积核可以捕捉并学习更细节和局部的特征，引入更多的非线性变换能力，表达能力更强，并降低了过拟合的风险[81]。

Li 等[82]使用小卷积核和移位操作的方法解决了大卷积核在硬件操作和兼容性方面的问题，同时实现了与大卷积核相同的效果。该方法显著提高了普通 CNN 的准确性，并明显减少了计算需求。Rong 等[83]采用连续小卷积神经网络(CSCNN)对脑电图信号进行分类，结果表明 CSCNN 具有较高的准确率，同时具有较少的参数。

3.3 降低输入通道数

对于标准的卷积操作，由 3.1 节的公式可知，输入和输出通道数对运算量的大小存在影响，因此在对网络性能影响不大的情况下，可以减少输入通道数来降低卷积运算量。通过适当设置 1×1 的卷积核数量，可以实现对通道维度的降维和升维，降维和升维取决于所采用 1×1 卷积核的个数。

Iandola 等[84]设计了 SqueezeNet 网络结构，该结构使用“火焰模块”(Fire Module)作为基础模块。该基础模块由

一个 Squeeze 层和一对并行的 Expand 层组成，如图 5 所示。图 5 中，输入特征尺寸 $H\times W$，通道数是 M，经过 Squeeze 层与 Expand 层，然后进行融合处理。其中 Squeeze 层使用 1×1 卷积进行降维，通道数少于上一层的通道数，达到压缩的目的；在 Expand 层并行使用 1×1 和 3×3 卷积，获得不同感受野的特征图，实现减少参数量和计算量，最后经过 Concat 层将两个特征图进行拼接，并将其作为最终输出，S_1，e_1 和 e_2 是可调参数。1×1 卷积在模型中的占比较多，参数量是 3×3 卷积参数的 1/9，可减少模型的运算量。SqueezeNet 首先将图像输入 Covn 1，然后使用 8 个 Fire Module，最后一个卷积为 Covn 10，一共使用 3 个 Pool 层。最终该模型与 AlexNet 相比，Top-1 和 Top-5 精度相当，但是 SqueezeNet 模型参数仅为 AlexNet 的 1/50。同时，采用混合模型压缩技术，可以进一步压缩该模型的大小。例如用深度压缩技术，在性能几乎不损失的情况下，模型可以压缩到 0.47 MB[85]。

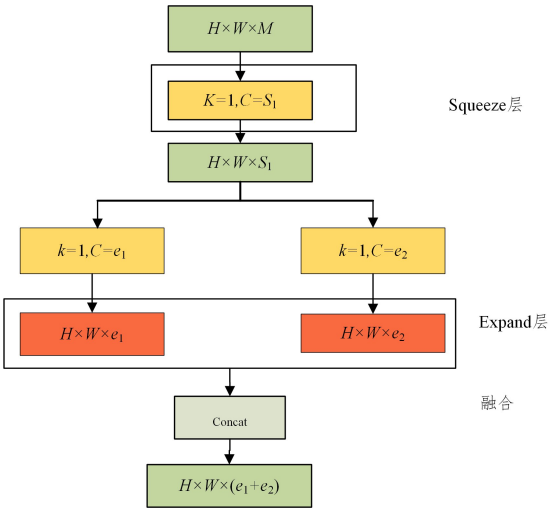

图 5 Fire Module 的结构

Fig. 5 Structure of Fire Module

Kathirgamaraja 等[86]在资源受限的边缘设备上成功实现了适用于 ImageNet 的 SqueezeNet v1.1 结构。Minu 等[87]将 SqueezeNet 用于航空图像分类，最高准确率达到了 98.97%，运行时间也大为缩短。

3.4 分解卷积运算

标准卷积考虑了空间结构和通道两个维度，导致模型参数量大且存在冗余。为了解决这个问题，可以将标准卷积在空间维度和通道维度上进行分解。通常有两种分解方法：第一种是深度可分离卷积，它将标准卷积在空间结构和通道上进行解耦；第二种是空间可分离卷积，即把 $k\times k$ 卷积分成 $k\times 1$ 和 $1\times k$ 的非对称卷积。通过对卷积运算进行分解，可以减少网络参数量，同时加快推理速度。

深度可分离卷积[88]将常规卷积分解为深度卷积(Depthwise Convolution，DW)和逐点卷积(Pointwise Convolution，PW)两个过程，如图 6 所示。首先，深度神经网络卷积是对输入特征图的每个通道独立地进行卷积操作，即将每个输入通道与对应通道数的卷积核进行卷积运算，而不是使用具有多个通道的卷积核。由于深度卷积通道间缺少特征融合，并且没有改变通道数，因此需继续连接一个逐点的 1×1 的卷积，将其作用在所有通道上，来融合不同通道间的特征。由 3.1 节中的公式可知，深度可分离卷积与标准卷积的计算量的比值为：

$$\rho=\frac{1}{C_o}+\frac{1}{k^2} \tag{11}$$

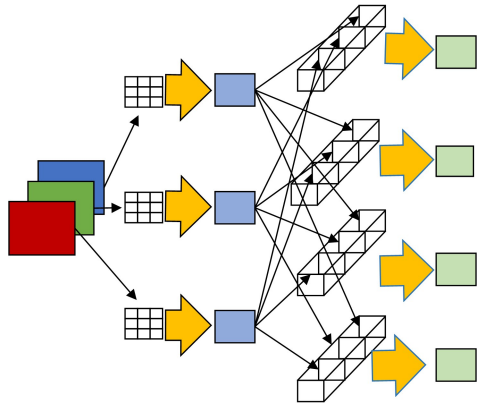

图 6 深度可分离卷积

Fig. 6 Depthwise separable convolution

在 MobileNet 网络中，Howard 等[88]利用深度可分离卷积对标准卷积进行了分解。在 MobileNet 模型中，采用的是 3×3 卷积逐通道卷积核，使用 BN 和 ReLU 6 激活函数。由式(11)可得，深度可分离卷积的计算量是标准卷积的 1/9 至 1/8，结构框图如图 7 所示。MobileNet 模型是特别为资源受限的移动和嵌入式设备设计的，以在保持合理的准确度的同时最小化模型的大小和计算量。

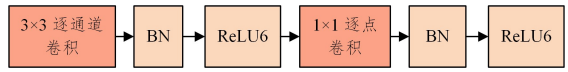

图 7 MobileNet 的结构

Fig. 7 Structure of MobileNet

为了进一步控制模型的大小与计算量，MobileNet V1 还设置了两个超参数[88]，即控制特征图通道数的宽度乘子 α 和控制特征图的尺寸分辨率乘子 ρ，则计算量为：

$$\alpha C_i\times\rho H\times\rho W\times 9+\alpha C_i\times\alpha C_o\times\rho H\times\rho W \tag{12}$$

其中，$\alpha,\rho\in(0,1]$。通过调节 α 和 ρ，模型在精度、参数和计算量之间达到最优化。最终，在精度大致相等的情况下，MobileNet 模型大小约是 VGG-16 的 3.1%，计算量约是其 3.7%[23]。当输入图像的分辨率为 160×160 时，MobileNet 的精度比 AlexNet 高 4%，计算量减少了 9.4%[79]。为了克服 MobileNet V1 的局限性，Sandler 等[89]提出了 MobileNet V2，该版本创新性地引入了 Inverted Residual Block 结构，将线性瓶颈层和下采样思想相结合，其 Inverted Residual Block 采用轻量级的深度可分离卷积。Koonce 等[90]在 MobileNet V2 的基础上进一步改进了模型，添加了 SENet(Squeeze-and-Excitation)模块[91]以及 Hard-swish 激活函数，称为 MobileNet V3。这些改进显著增强了模型在运行速度、准确度和功耗方面的性能，使得其非常适合在计算能力较弱的边缘设备上使用。

Chollet[92]提出了 Xception 轻量化网络，它的 Extreme Inception 模块与深度可分离卷积几乎是等价的，唯一区别是 Depth wise Convolution 与 Point wise Convolution 的计算

顺序不同，但顺序对模块性能的影响不大。

采用 $k\times1$ 和 $1\times k$ 的卷积组合代替 $k\times k$ 的卷积，则参数量和 FLOPs 都是原来的 $2/k$。Gholami 等[93]对 Fire Module 进行改进，提出了 SqueezeNext 轻量化网络结构，如图 8(a)所示。该结构首先通过两阶段的 Squeeze 对通道数进行缩减，每个阶段的 Squeeze 都将通道数减半，Squeeze 称为 Bottleneck module；在将 Fire Module 的 Expand 层使用两个低秩的卷积核 3×1 和 1×3 代替一个满秩的卷积核 3×3 时，Expand 变为低秩滤波(Low Rank Filters)。同时移除了该层的拼接 1×1 卷积，添加了 1×1 卷积来恢复与输入同样的通道数。卷积结构过程中使用 1×1 的卷积个数来灵活控制通道数量，从而降低模型的计算量。在 SqueezeNext 保持了与 AlexNet 相当的 Top-5 精度的条件下，参数数量约为 2%。

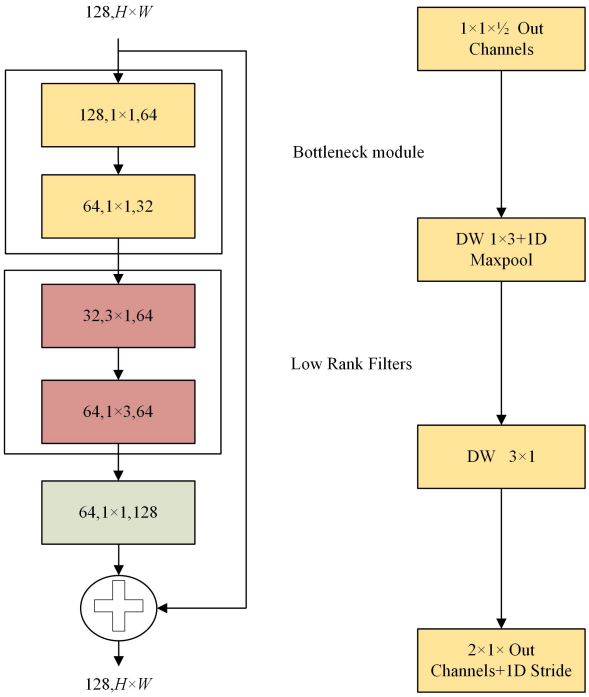

图 8 SqueezeNext 和 EffNet 结构

Fig. 8 Structures of SqueezeNext and EffNet

Freeman 等[94]提出了 EffNet 结构，如图 9 所示。主要把 MobileNet 的深度卷积(DW)的 3×3 卷积核进一步改进为 1×3 和 3×1 的一维卷积核(空间上的分离)，并在两个卷积核中间再增加一个一维的 Max pooling。

3.5 调整卷积宽度

将标准卷积在空间和通道上进行解耦后，增加了模型中的卷积操作。其中，1×1 卷积占用了大量的计算资源和模型参数，同时导致不同通道之间的联系约束增大。如 MobileNet 在 1×1 卷积中占据模型总参数量的 74.59%，计算花费了 94.86%[95]。另外深度可分离卷积 $FLOPs=H\times W\times C_i\times(C_o+9)$，当 C_o 远大于 9 时，模型的主要计算都花费在 1×1 的逐点卷积上。为此可以调整卷积结构的宽度，将输入和卷积核分成两个或多个组，然后每个组并行进行独立的卷积计算，以实现模型的轻量化。

分组卷积的原理与深度可分离卷积的操作类似，深度可分离卷积实现了通道的解耦，而分组卷积则是对数据进行了分离。在分组卷积中，若卷积核个数是 c，将卷积核平均分成 g 组，则每一组的卷积核个数为 $\frac{c}{g}$，对卷积核进行分组后，也需要对输入特征的通道进行分组，输入特征分组的组数和卷积核分组后的组数保持一致。

Xie 等[96]融合 Inception[97]网络结构和 ResNet[98]的 BottleNeck 的思想，提出了 ResNeXt 模块，如图 9 所示。ResNeXt 模块的核心是分组卷积，图中用 * 表示。分组卷积层行执行 32 组卷积，并且每个分支采用完全相同的卷积结构。这种设计不会显著增加模型的计算成本，并更适合于边缘智能硬件的设计原则。ResNeXt 在 ImageNet 数据集上的表现优于 ResNet-200[99]和 Inception-v3[100]，特别是相比 ResNet-200，ResNeXt-101，在复杂度只增加 50%的情况下实现了更高的精度[99]。

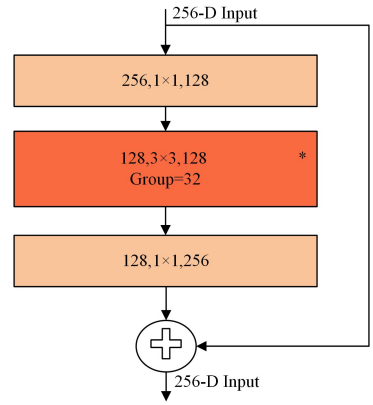

图 9 ResNeXt 结构

Fig. 9 Structure of ResNeXt

分组卷积虽然降低了计算成本，但阻塞了通道之间的信息并削弱了表征能力。从优化网络角度出发，Zhang 等[95]在 ShuffleNet 结构中提出了逐点分组卷积(Pointwise Group Convolution)和通道混洗(Channel Shuffle)。图 11 给出了常规的两个组卷积操作，可以看到输出特征仅由一部分输入特征通道计算得出，组和组之间没有信息的交互，可能降低了特征之间的交流和信息跨通道的组织能力。为了使得信息在组之间流动，通道混洗恰好实现了这个过程，如图 12 所示。通道混洗主要是将每一个组的特征重新排列，再分散到不同的组，使得输出的特征包含每一组的特征，再进行下一组卷积。Reshape，Transpose 和 Flatten 这 3 个步骤便可以实现通道混洗。

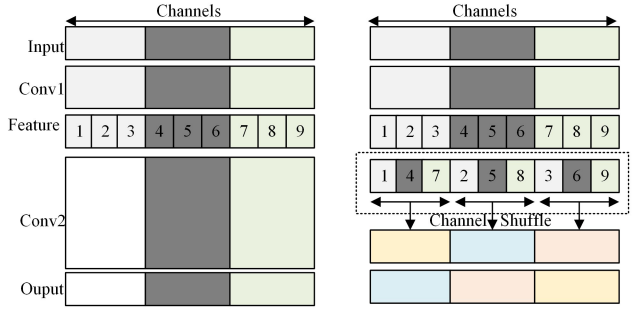

图 10 分组卷积与通道混洗

Fig. 10 Group convolution and channel shuffle

利用上述的通道混洗操作，ShuffleNet 构建了以下网络的基本模块，如图 11 所示。图 11(a)是基本的 ShuffleNet 单元，使用 1×1 组卷积，然后进行通道混洗。图 11(b)在图 11(a)的基础上，使用了带降采样的 ShuffleNet 单元，在旁路使用了步长为 2 的 3×3 平均池化进行降采样，在主路中使用 3×3 卷积步长为 2 来实现降采样，最后使用 Contat 实现了通道数的增加，替换了 Add 操作。给定输入大小 $H\times W\times C_i$、输出通道C_o和卷积组数 g，则在基本 ResNet 模块，有：

$$FLOPs=H\times W\times C_o\times(2\ C_i+9\ C_o) \tag{13}$$

图 10 给出的 ResNext 模块如下：

$$FLOPs=H\times W\times C_o\times\left(2\ C_i+\frac{9\ C_o}{g}\right) \tag{14}$$

图 13(a)给出的 ShuffleNet 结构如下：

$$FLOPs=H\times W\times C_o\times\left(\frac{2\ C_i}{g}+9\ C_o\right) \tag{15}$$

很明显，当组数相对较多时，FLOPs 减少，而且组数越多，具有信道混洗的模型在性能上优于对应的模型，这表明了跨组信息交换的重要性。最终，在精度相同的情况下，ShuffleNet V1 比 ResNet 快 13 倍。因此，ShuffleNet V1 适合部署在资源有限的边缘的移动端上。

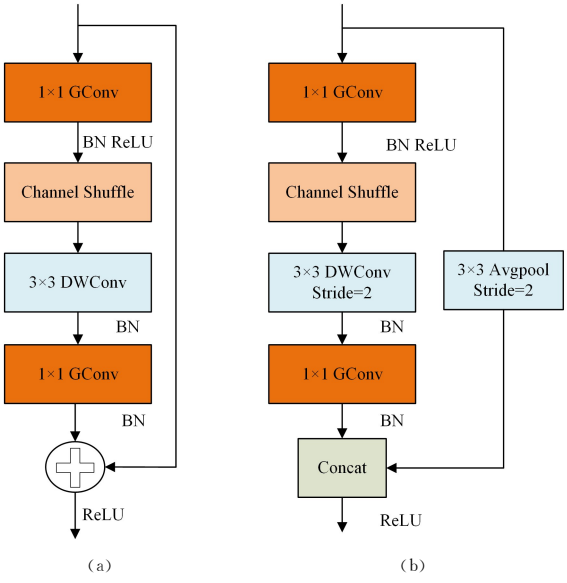

图 11 ShuffleNet 结构

Fig. 11 Structure of ShuffleNet

在衡量轻量化模型的性能时，Ma 等[101]发现仅仅依赖 FLOPs 是不够的，还应该考虑内存访问时间(Memory Access Cost，MAC)和系统处理并行的硬件条件。Ma 等提出了设计高性能轻量化模型的 4 个基本原则[101]：

1)卷积层的 C_i 与 Co 相等时，MAC 最小，速度最快；

2)使用过多的分组卷积会增加计算负担；

3)网络的碎片化会降低并行处理的效率，模型中的分支数量越少，模型速度越快；

4)逐元素(Element Wise)操作使 MAC 较高，应减少逐元素操作。

根据以上原则，对上面 ShuffleNet 的基本单元进行了改进，Ma 等[101]提出了 ShuffleNet V2 版本，如图 12 所示。图 12(a)将输入特征分成两部分，一部分执行深度可分离卷积，将计算结果与另外一通道 Concat，最后进行通道混洗，完成信息互通；图 12(b)给出了去掉 Channel Split 操作的结构，最后的 Concat 时通道翻倍。ShuffleNet V2 中都没有使用到 1×1 的组卷积。结果 ShuffleNet V2 比 ShuffleNet V1 和 Xception 分别约快 63%和 25%。总体上，该模型在速度和精度的平衡上达到了当前的最佳水平，非常适合于部署在智能物联网受限的边端。

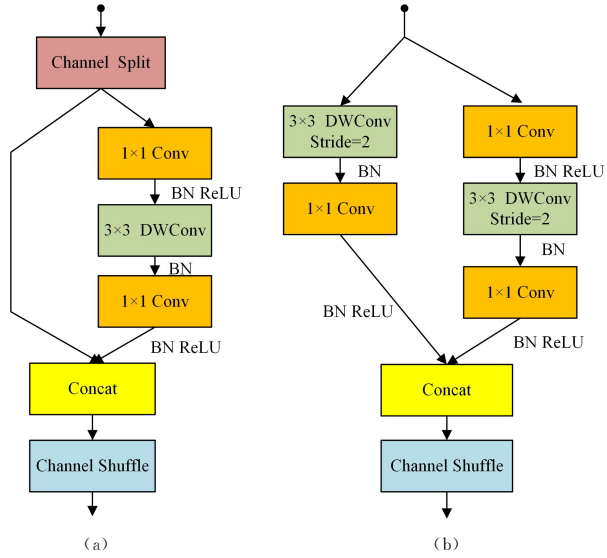

图 12 ShuffleNet V2 结构

Fig. 12 Structure of ShuffleNet V2

3.6 各种轻量化神经网络模型的效果

卷积操作是深度学习架构的重要组成部分。借助调整卷积核的尺寸、减少输入通道数、分解卷积操作，以及优化卷积层宽度等手段，可以有效地实现神经网络的轻量化。这样的轻量化网络在性能影响不大的基础上，能减少参数数量和计算负担，以此适配资源受限的边缘智能设备。为了测试 SqueezeNet，MobileNet，ShuffleNet 等多种轻量级模型在不同边缘智能终端设备上的表现，使用 Oxford102 数据集，该数据集专门用于细粒度图像分类，含有 102 种不同类别的花朵共 8189 张图片。为了防止模型在这个数据集上过拟合，在模型中统一添加了 L2 正则化项，强度设为 1×10^{-5}。此外，模型的训练采用了交叉熵损失函数和基于 Adam 算法的优化器，设定学习率为 0.0001。FLOPs 的值是在 batch 为 1、输入通道为 3、输入图像分辨为 224×224 的条件下计算得到的。轻量化模型的效果如表 8 所列。

从表 8 可以看出，轻量化的神经网络模型在各种边缘智能设备上的性能因设备的硬件配置和操作系统等因素而异，尤其是在计算能力有限的智能边缘设备上，性能的差异尤其显著。在相同条件下，尽管 Xception，Inception-V3，MobileNet-V2 和 SqueezeNet 相对于其他网络架构，在分类准确度方面表现相对较差，但 MobileNet-V2 和 SqueezeNet 却在参数量和模型尺寸方面具备明显优势。得益于深度可分离卷积和通道混洗操作，相比之下，ShuffleNet-V2 和 MobileNet-V3Large 在参数量、模型大小、准确度、推理时间等方面展现出了更卓越的表现。总体来说，对比轻量化模型和传统的深度学习模型可以发现，轻量模型在减少计算复杂度方面和

模型大小占有显著优势,在准确度方面也并不比传统的模型如 AlexNet,VGG-16 逊色。特别是 ShuffleNet-V2 和 MobileNet-V3Large 不仅在参数量、计算复杂度和存储上做了进一步优化,而且在准确度方面,它们比传统的 AlexNet 和 VGG-16 网络高出 7%以上。这表明进行模型轻量化并非总是需要以牺牲精度为代价,通过降低计算复杂度和减少参数量,模型的准确度、训练时间和推理时间也能够得到提升。

表 8 轻量化模型效果的对比

Table 8 Comparison of lightweight model effects

Model	Parameter	FLOPs	Size	Top-1 Acc	Top-5 Acc	Train-time/m	Val-time/s	Processor/ Operating System
SqueezeNet	1.25×10^6	0.82×10^9	4.78×10^6	0.9118	0.9843	113.10	457.0	2 核 CPU:Intel Xeon(R)Gold 5218R/windows server 2019
MobileNet-V3Large	5.48×10^6	0.23×10^9	21.10×10^6	0.9745	0.9951	147.90	331.0	
ResNet-50	25.55×10^6	4.13×10^9	95.70×10^6	0.9069	0.9814	49.10	640.0	
SqueezeNet	1.25×10^6	0.82×10^9	4.78×10^6	0.9216	0.9704	11.07	87.1	* 8 核 CPU:AMD Ryzen7 3700X,GPU:GTX1660 Super/windows 11
ShuffleNet-V2	2.28×10^6	0.15×10^9	8.79×10^6	0.9510	0.9922	8.10	64.0	
MobileNet-V2	$\mathbf{3.50\times10^6}$	$\mathbf{0.33\times10^9}$	$\mathbf{13.50\times10^6}$	**0.8667**	**0.9745**	**19.60**	**192.0**	
MobileNet-V3Large	5.48×10^6	0.23×10^9	21.10×10^6	0.9598	0.9902	10.67	64.5	
DenseNet-121	7.98×10^6	2.89×10^9	30.80×10^6	0.9696	0.9951	14.50	65.0	
ResNet-18	11.69×10^6	1.82×10^9	44.60×10^6	0.8951	0.9794	55.27	772.0	
Xception	22.86×10^6	4.60×10^9	87.40×10^6	0.8343	0.9049	47.60	142.0	
ResNext	25.03×10^6	4.28×10^9	95.70×10^6	0.8755	0.9676	20.25	238.0	
Inception-V3	23.83×10^6	2.86×10^9	103×10^6	0.8892	0.9755	45.80	211.0	
AlexNet	57.42×10^6	0.71×10^9	233×10^6	0.8471	0.9559	14.42	75.0	
VGG-16	138.36×10^6	15.47×10^9	552×10^6	0.8755	0.9725	44.95	157.0	
MobileNet-V3 Large	5.48×10^6	0.23×10^9	21.10×10^6	0.9412	0.9922	54.98	211.0	2 核 CPU:Intel Core i5/MAC OS
EfficientNet-b1	0.61×10^6	7.79×10^9	30.10×10^6	0.9676	0.9941	253.17	762.0	

4 展望与总结

虽然神经网络轻量化已取得令人瞩目的成就,并已被应用在许多资源有限的设备上,但这些模型的设计与优化对设计者提出了较高的要求,它们大多基于实验结果提供了一种适用的方案,需要更多的人为干预才能实现;需要不断试错、对比性能和积累经验;需要在模型大小、运算速度、准确率和设备资源之间进行权衡。针对以上情况,未来一段时间研究轻量级深度神经网络模型以此来适配边缘智能的研究还将继续深入,可以从以下 4 个方面着手。

1)结合自动化机器学习(AutoML)技术和神经网络架构搜索(NAS)算法,可以降低设计和优化针对边缘智能应用的轻量级神经网络难度,同时减少人为干预。

2)目前,常见的注意力机制有 SENet,CBAM[102](Convolutional Block Attention Module),ECANet[103](Efficient Channel Attention Network)等,它们在提升神经网络性能方面表现出色。表 8 中在带 * 条件下,以 MobileNet-V2 为例,结合注意力机制与轻量级神经网络所得的模型性能如表 9 所列。与表 8 中的加粗部分对比,引入注意力机制的模型在维持参数规模几乎不变的情况下显著提高了整体性能。未来的研究可注重于开发简化且高效的新型注意力机制,并将其与不断改进的轻量级神经网络技术相结合,构建高效的轻量级深度神经网络,以满足智能边缘设备在各种应用环境下的性能和快速计算需求。

表 9 轻量化模型与注意力机制结合的效果

Table 9 Effectiveness of combining lightweight models with attention mechanisms

Model	Parameter	Top-1 Acc	Top-5 Acc	Train-time/min	Val-time/s
CBAM MobileNet-V2	2.35×10^6	0.9578	0.9931	5.6	36
SE MobileNet-V2	2.35×10^6	0.9686	0.9931	23.0	23

3)目前,Zhang 等[104]利用 Vision Transformer 作为教师网络,ShuffleNet-V2-x0.5 作为学生网络,取得了 99.00%的高准确率,而学生模型中仅有 25.51×10^6 FLOPs 和 0.20×10^6 个网络参数,应用于轮胎胎面花纹和凹陷图像的分类。Basha 等[105]将 MobileNet v1 剪枝用于 ARM 设备上的推断时间减少至 84 ms。这些例子表明,结合模型压缩和轻量化网络架构,能表现出优于采用复杂网络模型的综合性能。未来研究可同时优化轻量化网络的架构设计与模型压缩技术,有助于促进轻量级神经网络模型在边缘智能上的广泛应用。

4)探索硬件与轻量化深度神经网络算法的协同设计,开发能够适用于不同计算环境的跨模态轻量化模型。

本文首先介绍了边缘智能发展的历程,并阐述了深度神经网络难以适配边缘智能的现实需求,需对深度神经网络进行轻量化。然后,对目前深度网络轻量化的两个主流方面分别进行了详尽分析;并分别对压缩技术中的一些代表性方法和轻量化模型在不设备上的运行结果进行比较和分析。最后,展望了轻量化深度神经网络适配边缘智能领域的未来发展方向。希望本文能够让研究者更好地了解于深度神经网络轻量化模型,促进和推动轻量化深度神经网络适配边缘智能领域的未来发展。

参 考 文 献

[1] LEI B,ZHOU J,MA M,et al. DQN based Blockchain Data Storage in Resource-constrained IoT System[C]// 2023 IEEE Wireless Communications and Networking Conference(WCNC). IEEE,2023:1-6.

[2] CAO K, LIU Y, MENG G, et al. An overview on edge computing research[J]. IEEE Access, 2020, 8: 85714-85728.

[3] GHAZNAVI M, JALALPOUR E, SALAHUDDIN M A, et al. Content delivery network security: A survey[J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2166-2190.

[4] DAS R, INUWA M M. A review on fog computing: issues, characteristics, challenges, and potential applications[J]. Telematics and Informatics Reports, 2023, 10(1): 1-20.

[5] BABAR M, KHAN M S, ALI F, et al. Cloudlet computing: recent advances, taxonomy, and challenges[J]. IEEE Access, 2021, 9: 29609-29622.

[6] ZHENG F, ZHU D W, ZANG W Q, et al. Edge Computing: Review and Application Research on New Computing Paradigm [J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(4): 541-553.

[7] SHI W, CAO J, ZHANG Q, et al. Edge computing: Vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[8] ZHANG X, WANG Y, LU S, et al. OpenEI: An open framework for edge intelligence[C]// 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019: 1840-1851.

[9] ZHOU Z, CHEN X, LI E, et al. Edge intelligence: Paving the last mile of artificial intelligence with edge computing[J]. Proceedings of the IEEE, 2019, 107(8): 1738-1762.

[10] DENG S, ZHAO H, FANG W, et al. Edge intelligence: The confluence of edge computing and artificial intelligence[J]. IEEE Internet of Things Journal, 2020, 7(8): 7457-7469.

[11] SUBHASHINI R, KHANG A. The role of Internet of Things (IoT) in smart city framework[M]//Smart Cities. CRC Press, 2023: 31-56.

[12] SINGH A, SAINI K, NAGAR V, et al. Artificial intelligence in edge devices[M]// Advances in Computers. Elsevier, 2022, 127: 437-484.

[13] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[14] SUBRAMANIAN M, LV N P, VE S. Hyperparameter optimization for transfer learning of VGG16 for disease identification in corn leaves using Bayesian optimization[J]. Big Data, 2022, 10(3): 215-229.

[15] JAWOREK-KORJAKOWSKA J, KLECZEK P, GORGON M. Melanoma Thickness Prediction Based on Convolutional Neural Network With VGG-19 Model Transfer Learning[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE Computer Society, 2019: 2748-2756.

[16] SHAFIQ M, GU Z. Deep residual learning for image recognition: A survey[J]. Applied Sciences, 2022, 12(18): 8972.

[17] PANDA M K, SUBUDHI B N, VEERAKUMAR T, et al. Modified ResNet-152 Network With Hybrid Pyramidal Pooling for Local Change Detection[J]. IEEE Transactions on Artificial Intelligence, 2023, 1(1): 1-14.

[18] YANG Z, ZHANG H. Comparative Analysis of Structured Pruning and Unstructured Pruning[C]// International Conference on Frontier Computing. Singapore: Springer Nature Singapore, 2021: 882-889.

[19] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv: 1510.00149, 2015.

[20] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient inference[J]. arXiv: 1611.06440, 2016.

[21] DONG X, YANG Y. Network Pruning via Transformable Architecture Search[J]. arXiv: 1905.09717, 2019.

[22] SAKAI Y, ETO Y, TERANISHI Y. Structured pruning for deep neural networks with adaptive pruning rate derivation based on connection sensitivity and loss function[J]. Journal of Advances in Information Technology, 2022, 13(3): 295-300.

[23] CHOI Y, EL-KHAMY M, LEE J. Compression of deep convolutional neural networks under joint sparsity constraints[J]. arXiv: 1805.08303, 2018.

[24] WANG M, TANG J, ZHAO H, et al. Automatic Compression of Neural Network with Deep Reinforcement Learning Based on Proximal Gradient Method[J]. Mathematics, 2023, 11(2): 338.

[25] ZHAO R, LUK W. Efficient Structured Pruning and Architecture Searching for Group Convolution[C]// IEEE/CVF International Conference on Computer Vision Workshop (ICCVW 2019). IEEE Computer Society, 2019: 1961-1970.

[26] XU K, WANG Z, GENG X, et al. Efficient joint optimization of layer-adaptive weight pruning in deep neural networks[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 17447-17457.

[27] ANWAR S, HWANG K, SUNG W. Structured pruning of deep convolutional neural networks[J]. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2017, 13(3): 1-18.

[28] LOUATI H, LOUATI A, BECHIKH S, et al. Embedding channel pruning within the CNN architecture design using a bi-level evolutionary approach[J]. The Journal of Supercomputing, 2023, 79(14): 16118-16151.

[29] XIA M, ZHONG Z, CHEN D. Structured pruning learns compact and accurate models[J]. arXiv: 2204.00408, 2022.

[30] ECCLES B J, RODGERS P, KILPATRICK P, et al. DNNShifter: An efficient DNN pruning system for edge computing[J]. Future Generation Computer Systems, 2024, 152: 43-54.

[31] SUN X, SHI H. Towards Better Structured Pruning Saliency by Reorganizing Convolution[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024: 2204-2214.

[32] QIAN Y, HUANG W, YU Q, et al. Robust Filter Pruning Guided by Deep Frequency-Features for Edge Intelligence[J/OL]. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4691079.

[33] BASHAS H S, FARAZUDDIN M, PULABAIGARI V, et al. Deep model compression based on the training history[J]. Neurocomputing, 2024, 573: 127257.

[34] KIM S, HOOPER C, WATTANAWONG T, et al. Full stack optimization of transformer inference: a survey[J]. arXiv: 2302.

14017,2023.
[35] RASTEGARI M,ORDONEZ V,REDMON J,et al. Xnor-net:Imagenet classification using binary convolutional neural networks[C]// European Conference on Computer Vision. Cham:Springer International Publishing,2016:525-542.
[36] VORABBI L,MALTONI D,SANTI S. On-Device Learning with Binary Neural Networks[C]// International Conference on Image Analysis and Processing. Cham:Springer Nature Switzerland,2023:39-50.
[37] LIU B,LI F,WANG X,et al. Ternary weight networks[C]// 2023 IEEE International Conference on Acoustics,Speech and Signal Processing(ICASSP 2023). IEEE,2023:1-5.
[38] RAZANI R,MORIN G,SARI E,et al. Adaptive binary-ternary quantization[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021:4613-4618.
[39] CHEN W,QIU H,ZHUANG J,et al. Quantization of deep neural networks for accurate edge computing[J]. ACM Journal on Emerging Technologies in Computing Systems(JETC),2021,17(4):1-11.
[40] TMAMNA J,AYED E B,FOURATIR,et al. Bare-Bones particle Swarm optimization-based quantization for fast and energy efficient convolutional neural networks[J]. Expert Systems,2023(12):1.
[41] SCHAEFER C J S,JOSHI S,LI S,et al. Edge inference with fully differentiable quantized mixed precision neural networks[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024:8460-8469.
[42] ZHANG H,YAO B,SHAO W,et al. Mixed Precision Quantized Neural Network Accelerator for Remote Sensing Images Classification[C]// 2023 IEEE 16th International Conference on Electronic Measurement & Instruments(ICEMI). IEEE,2023:172-176.
[43] LI B,WANG L,WANG Y,et al. Mixed-Precision Network Quantization for Infrared Small Target Segmentation[J]. IEEE Transactions on Geoscience and Remote Sensing,2024,62:3346904.
[44] WANG Y Z,GUO B,WANG H L,et al. Adaptive Model Quantization Method for Intelligent Internet of Things Terminal[J]. Computer Science,2023,50(11):306-316.
[45] HUANG C,LIU P,FANG L. MXQN:Mixed quantization for reducing bit-width of weights and activations in deep convolutional neural networks[J]. Applied Intelligence,2021,51:4561-4574.
[46] KUNDU S,WANG S,SUN Q,et al. Bmpq:bit-gradient sensitivity-driven mixed-precision quantization of dnns from scratch[C]// 2022 Design,Automation & Test in Europe Conference & Exhibition(DATE). IEEE,2022:588-591.
[47] LOUIZOS C,REISSER M,BLANKEVOORT T,et al. Relaxed quantization for discretized neural networks[J]. arXiv:1810.01875,2018.
[48] LANE N D,BHATTACHARYA S,GEORGIEV P,et al. Deepx:A software accelerator for low-power deep learning inference on mobile devices[C]// 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks(IPSN). IEEE,2016:1-12.
[49] FANG S,KIRBY R M,Zhe S. Bayesian streaming sparse Tucker decomposition[C]// Uncertainty in Artificial Intelligence. PMLR,2021:558-567.
[50] ERICHSON N B,MANOHAR K,BRUNTON S L,et al. Randomized CP tensor decomposition[J]. Machine Learning:Science and Technology,2020,1(2):025012.
[51] SWAMINATHAN S,GARG D,KANNAN R,et al. Sparse low rank factorization for deep neural network compression[J]. Neurocomputing,2020,398:185-196.
[52] BAO X,LIANG J,XIA Y,et al. Low-rank decomposition fabric defect detection based on prior and total variation regularization[J]. The Visual Computer,2022,38(8):2707-2721.
[53] CHENG T,TONG X,ZHANG Y,et al. Convolutional neural networks with low-rank regularization[J]. arXiv:1511.06067,2015.
[54] XIAO J,ZHANG C,GONG Y,et al. HALOC:Hardware-Aware Automatic Low-Rank Compression for Compact Neural Networks[J]. arXiv:2301.09422,2023.
[55] IDELBAYEV Y,CARREIRA-PERPIÑáN M A. Low-rank compression of neural nets:Learning the rank of each layer[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020:8049-8059.
[56] MENG X F,LIU F,LI G,et al. Review of Knowledge Distillation in Convolutional Neural Network Compression[J]. Journal of Frontiers of Computer Science and Technology,2021,15(10):1812-1829.
[57] 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.
[58] SI Z F,QI H G. Survey on knowledge distillation and its application[J]. Journal of Image and Graphics,2023,28(9):2817-2832.
[59] HINTON G,VINYALS O,DEAN J. Distilling the knowledge in a neural network[J]. arXiv:1503.02531,2015.
[60] ROMERO A,BALLAS N,KAHOU S E,et al. Fitnets:Hints for thin deep nets[J]. arXiv:1412.6550,2014.
[61] ZAGORUYKO S,KOMODAKIS N. Paying more attention to attention:Improving the performance of convolutional neural networks via attention transfer[J]. arXiv:1612.03928,2016.
[62] ZHANG H L,CHEN D F,WANG C. Confidence-aware multi-teacher knowledge distillation[C] // IEEE International Conference on Acoustics,Speech and Signal Processing(ICASSP 2022). IEEE,2022.
[63] CHOI J,CHO H,CHEUNG S,et al. ORC:Network group-based knowledge distillation using online role change[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023:17381-17390.
[64] QIAN Y G,MA J,HE N N,et al. Two-stage Adversarial Knowledge Transfer for Edge Intelligence[J]. Journal of Software,2022,33(12):4504-4516.
[65] MISHRA R,GUPTA H P. Designing and training of lightweight neural networks on edge devices using early halting in knowledge distillation[J]. IEEE Transactions on Mobile Com-

puting, 2024(25): 4665-4677.
[66] GOU J, HU Y, SUN L, et al. Collaborative knowledge distillation via filter knowledge transfer[J]. Expert Systems with Applications, 2024, 238: 121884.
[67] LI T, LI J, LIU Z, et al. Few sample knowledge distillation for efficient network compression[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 14639-14647.
[68] PHAM C, NGUYEN V A, LE T, et al. Frequency Attention for Knowledge Distillation[C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024: 2277-2286.
[69] ZHANG Y, XIANG T, HOSPEDALES T M, et al. Deep mutual learning[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4320-4328.
[70] HUANG Z H, YANG X Y, YU J, et al. Mutual Learning Knowledge Distillation Based on Multi-stage Multi-generative Adversarial Network[J]. Computer Science, 2022, 49(10): 169-175.
[71] MIRZADEH S I, FARAJTABAR M, LI A, et al. Improved knowledge distillation via teacher assistant[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2020: 5191-5198.
[72] KWON S J, LEE D, KIM B, et al. Structured compression by weight encryption for unstructured pruning and quantization [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1909-1918.
[73] QU X, WANG J, XIAO J. Quantization and knowledge distillation for efficient federated learning on edge devices[C]// 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2020: 967-972.
[74] CHANG W T, KUO C H, FANG L C. Variational Channel Distribution Pruning and Mixed-Precision Quantization for Neural Network Model Compression[C]// 2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT). IEEE, 2022: 1-3.
[75] BAI S, CHEN J, SHEN X, et al. Unified Data-Free Compression: Pruning and Quantization without Fine-Tuning[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 5876-5885.
[76] WANG W, ZHOU X, JIANG C, et al. A Lightweight Identification Method for Complex Power Industry Tasks Based on Knowledge Distillation and Network Pruning[J]. Processes, 2023, 11(9): 2780.
[77] YU P H, WU S S, KLOPP J P, et al. Joint pruning & quantization for extremely sparse neural networks[J]. arXiv: 2010.01892, 2020.
[78] KIM J, CHANG S, KWAK N. PQK: model compression via pruning, quantization, and knowledge distillation[J]. arXiv: 2106.14681, 2021.
[79] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012(25): 1-9.
[80] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv: 1409.1556, 2014.
[81] FANG L L, WANG X. Brain tumor segmentation based on the dual-path network of multi-modal MRI images[J]. Pattern Recognition, 2022(124): 108434.
[82] LI D C, LI L, CHEN Z Z, et al. Shift-ConvNets: Small Convolutional Kernel with Large Kernel Effects[J]. arXiv: 2401.12736, 2024.
[83] RONG Y Y, WU X, ZHANG Y M. Classification of motor imagery electroencephalography signals using continuous small convolutional neural network[J]. International Journal of Imaging Systems and Technology, 2020, 30(3): 653-659.
[84] 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.
[85] LI, X, LONG R J, YAN J, et al. TANet: a tiny plankton classification network formobile devices[J]. Mobile Information Systems, 2019(3): 1-8.
[86] KATHIRGAMARAJA P, KAMALAKKANNAN K, RATNASEGAR N, et al. Edgenet: Squeezenet like convolution neural network on embedded fpga[C]// 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2018: 81-84.
[87] MINU S, SUBASHKA R. Optimal Squeeze Net with Deep Neural Network-Based Arial Image Classification Model in Unmanned Ae-rial Vehicles[J]. Traitement duSignal, 2022, 39(1): 275-281.
[88] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv: 1704.04861, 2017.
[89] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4510-4520.
[90] KOONCE B, KOONCE B. MobileNetV3[J]. Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, 2021(5): 125-144.
[91] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7132-7141.
[92] CHOLLET F. XCEPTION: Deep learning with depthwise separable convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1251-1258.
[93] GHOLAMI A, KWON K, WU B, et al. Squeezenext: Hardware-aware neural network design[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops. 2018: 1638-1647.
[94] FREEMAN I, ROESE-KOERNER L, KUMMERT A. Effnet: An efficient structure for convolutional neural networks[C]//

2018 25th IEEE International Conference on Image Processing (ICIP). IEEE,2018:6-10.

[95] ZHANG X,ZHOU X,LIN M,et al. Shufflenet:An extremely efficient convolutional neural network for mobile devices[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:6848-6856.

[96] XIE S,GIRSHICK R,DOLLÁR P,et al. Aggregated residual transformations for deep neural networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017:1492-1500.

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

[98] HE K,ZHANG X,REN S,et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:770-778.

[99] HE K,ZHANG X,REN S,et al. Identity mappings in deep residual networks[C]// Computer Vision—ECCV 2016:14th European Conference, Amsterdam Springer International Publishing,2016:630-645.

[100] SZEGEDY C,VANHOUCKE V,IOFFE S,et al. Rethinking the inception architecture for computer vision[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:2818-2826.

[101] MA N,ZHANG X,ZHENG H T,et al. Shufflenet v2:Practical guidelines for efficient cnn architecture design[C]// Proceedings of the European Conference on Computer Vision(ECCV). 2018:116-131.

[102] WOO S,PARK J,LEE J Y,et al. Cbam:Convolutional block attention module[C]// Proceedings of the European Conference on Computer Vision(ECCV). 2018:3-19.

[103] WANG Q,WU B,ZHU P,et al. ECA-Net:Efficient channel attention for deep convolutional neural networks[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020:11534-11542.

[104] ZHANG F,LI D,LI S,et al. A Lightweight Tire Tread Image Classification Network[C]// 2022 IEEE International Conference on Visual Communications and Image Processing(VCIP). IEEE,2022:1-5.

[105] BASHA S H S,GOWDA S N,DAKALA J. A simple hybrid filter pruning for efficient edge inference[C]// 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022). IEEE,2022:3398-3402.

XU Xiaohua, born in 1980, associate professor. His main research interests include deep learning and edge computing.

ZHOU Zhangbing, born in 1974, Ph.D, professor. is a member of CCF (No. 28475M). His main research interests include service computing and edge computing.

(责任编辑:喻藜)