智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

神经网络压缩联合优化方法的研究综述

宁欣, 赵文尧, 宗易昕, 张玉贵, 陈灏, 周琦, 马骏骁

引用本文:
宁欣,赵文尧,宗易昕,张玉贵,陈灏,周琦,马骏骁. 神经网络压缩联合优化方法的研究综述[J]. 智能系统学报, 2024, 19(1): 36–57.
NING Xin, ZHAO Wenyao, ZONG Yixin, et al. An overview of the joint optimization method for neural network compression[J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(1): 36–57.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202306042

DOI: 10.11992/tis.202306042
网络出版地址: https://link.cnki.net/urlid/23.1538.TP.20240102.1655.002

神经网络压缩联合优化方法的研究综述

宁欣[1], 赵文尧[2], 宗易昕[3], 张玉贵[1], 陈灏[4], 周琦[1], 马骏骁[1]
（1. 中国科学院 半导体研究所, 北京 100083; 2. 合肥工业大学 微电子学院, 安徽 合肥 230009; 3. 中国科学院 前沿科学与教育局, 北京 100864; 4. 南开大学 人工智能学院, 天津 300071）

摘　要: 随着人工智能应用的实时性、隐私性和安全性需求增大, 在边缘计算平台上部署高性能的神经网络成为研究热点。由于常见的边缘计算平台在存储、算力、功耗上均存在限制, 因此深度神经网络的端侧部署仍然是一个巨大的挑战。目前, 克服上述挑战的一个思路是对现有的神经网络压缩以适配设备部署条件。现阶段常用的模型压缩算法有剪枝、量化、知识蒸馏, 多种方法优势互补同时联合压缩可实现更好的压缩加速效果, 正成为研究的热点。本文首先对常用的模型压缩算法进行简要概述, 然后总结了“知识蒸馏+剪枝”、“知识蒸馏+量化”和“剪枝+量化”3 种常见的联合压缩算法, 重点分析论述了联合压缩的基本思想和方法, 最后提出了神经网络压缩联合优化方法未来的重点发展方向。

关键词: 神经网络; 压缩; 剪枝; 量化; 知识蒸馏; 模型压缩; 深度学习
中图分类号: TP181　　**文献标志码:** A　　**文章编号:** 1673−4785(2024)01−0036−22

中文引用格式: 宁欣, 赵文尧, 宗易昕, 等. 神经网络压缩联合优化方法的研究综述 [J]. 智能系统学报, 2024, 19(1): 36–57.
英文引用格式: NING Xin, ZHAO Wenyao, ZONG Yixin, et al. An overview of the joint optimization method for neural network compression[J]. CAAI transactions on intelligent systems, 2024, 19(1): 36–57.

An overview of the joint optimization method for neural network compression

NING Xin[1], ZHAO Wenyao[2], ZONG Yixin[3], ZHANG Yugui[1], CHEN Hao[4], ZHOU Qi[1], MA Junxiao[1]
(1. Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China; 2. School of Microelectronics, Hefei University of Technology, Hefei 230009, China; 3. Bureau of Frontier Sciences and Education, Chinese Academy of Sciences, Beijing 100864, China; 4. College of Artificial Intelligence, Nankai University, Tianjin 300071, China)

Abstract: With the increasing demand for real-time, privacy and security of AI applications, deploying high-performance neural network on an edge computing platform has become a research hotspot. Since common edge computing platforms have limitations in storage, computing power, and power consumption, the edge deployment of deep neural networks is still a huge challenge. Currently, one method to overcome the challenges is to compress the existing neural network to adapt to the device deployment conditions. The commonly used model compression algorithms include pruning, quantization, and knowledge distillation. By taking advantage of complementary multiple methods, the combined compression can achieve better compression acceleration effect, which is becoming a hot spot in research. This paper first makes a brief overview of the commonly used model compression algorithms, and then summarizes three commonly used joint compression algorithms: “knowledge distillation + pruning”, “knowledge distillation + quantification” and "pruning + quantification", focusing on the analysis and discussion of basic ideas and methods of joint compression. Finally, the future key development direction of the neural network compression joint optimization method is put forward.
Keywords: neural network; compression; pruning; quantization; knowledge distillation; model compression; deep learning

深度神经网络已经被成功应用在计算机视觉、自然语言处理等任务中, 并在特定应用场景中取得了超越人类水平的成功。随着深度神经网

收稿日期: 2023−06−21.　**网络出版日期:** 2024−01−03.
基金项目: 国家自然科学基金项目 (62373343); 北京市自然科学基金项目(L233036).
通信作者: 张玉贵. E-mail: zhangyugui@semi.ac.cn.

络模型层数的加深、参数量的剧增, 其计算复杂度也在不断增加, 这不但会导致推理速度变慢, 而且在运行时会带来巨大的功耗。如果需要实现模型在工业界的应用落地, 则大多只能在存储容量大、算力充足、供电稳定的服务器、工作站甚至数据中心部署, 然而这会带来网络延迟问题、数据隐私安全问题以及硬件成本问题, 因此对神经网络压缩以实现端侧部署的研究意义重大。

移动互联网的发展使得移动端设备得到了广泛的普及与应用, 正成为人类日常生活不可或缺的一部分。人脸解锁、拍照识物以及语音助手等人工智能应用极大地方便了人们的生产生活, 人们对更加智能、更加隐私、更加安全的设备的需求也在与日俱增。随着智能制造、智能安防、智能家居以及自动驾驶等应用领域的兴起, 在嵌入式设备上部署神经网络以达到自动监测和控制的需求也持续增加。相比于服务器和工作站, 移动端和嵌入式设备具有存储资源少、算力不足、功耗受限等问题, 这限制了深度神经网络在端侧的部署应用。

目前, 关于模型压缩加速算法的文献综述[1-3]的研究重点主要在分别介绍剪枝、量化、知识蒸馏及轻量化网络设计等方法上, 对使用剪枝、量化、知识蒸馏等方法联合压缩加速这类新兴的方法介绍的较为粗略且参考文献的数量少, 随着新方法层出不穷, 其参考价值有所下降。如文献 [1] 并未总结多方法联合压缩; 文献 [2] 虽然总结了几种混合压缩加速方式, 但是未整理知识蒸馏和剪枝结合的研究, 且对知识蒸馏和量化结合的研究未进行详细的剖析; 文献 [3] 虽然列举了部分蒸馏与剪枝、蒸馏与量化结合的联合优化方法, 但是并未揭示这些研究的动机与思想。针对目前对联合优化方法综述的空缺, 本文重点整理总结了剪枝、量化、知识蒸馏 3 种常用模型压缩加速联合优化的方法, 总结归纳了联合优化方法的动机与思想, 希望对相关研究提供新思路。

1　神经网络压缩算法

1.1　神经网络压缩概述

神经网络轻量化的方法有直接设计轻量级网络和压缩现有的神经网络模型 2 种思路。

直接设计轻量级网络分为人工设计轻量级网络模型和基于神经网络架构搜索 (neural architecture search , NAS) 的自动化神经网络架构设计。人工设计轻量级神经网络的思路是设计更加高效的卷积计算方式、构造更高效的神经网络结构, 如谷歌采用深度可分离卷积替代传统卷积提出了 MobileNet v1[4], 并在此基础上引入倒残差结构 (inverted residuals) 和线性激活 (linear bottlenecks) 提出 MobileNet v2[5]; Face++采用分组逐点卷积 (pointwise group convolution) 以及通道混洗 (channel shuffle) 构建了 ShuffleNet v1[6], 在此基础上引入通道分割 (channel split) 降低访存延迟和提高计算并行度提出了 ShuffleNet v2[7]; Iandola 等[8] 构建了 Fire Module 设计出 SqueezeNet。虽然人工设计的高性能轻量化神经网络计算效率极高, 但是人工设计的方法要求设计者具有丰富的知识和经验, 设计难度大、成本高, 且并非是最优解。通过 NAS 在给定候选神经网络架构组成的搜索空间内按照一定的搜索策略寻求最优解的方式能够创造出人工尚未设计的轻量级神经网络, 如谷歌使用 NetAdapt 算法搜索得到卷积核和通道的最佳数量, 并引入挤压和激励 (squeeze-and-excitation, SE) 通道注意力机制和 h-switch 激活函数设计了 MobileNet v3[9]; 谷歌利用强化学习方法学习 NAS 搜索策略构建了 NasNet[10] 和 MnasNet[11], 实现了轻量级网络的自动设计。

神经网络模型压缩的算法主要有剪枝 (pruning)、量化 (quantization) 及知识蒸馏 (knowledge distillation, KD)。

1.2　常用压缩算法

深度学习网络压缩旨在利用神经网络参数和结构的冗余性对现有的模型进行压缩, 在不严重影响模型性能的情况下得到参数量更少、结构更加精简的模型。流行的模型压缩算法有剪枝、量化和知识蒸馏, 上述多种方法进行联合压缩加速能够综合各种方法的优势, 获得更好的压缩比和加速效果, 是近年来逐渐兴起的研究方向。在介绍联合压缩方法之前先分别介绍各类压缩方法。

1.2.1　剪枝

剪枝是指按照一定的准则判断参数重要性并裁剪模型中冗余的参数来缩小网络规模、精简网络结构, 从而达到减少计算量和内存消耗的目的。剪枝通过剔除模型中冗余的参数降低了模型的复杂度, 实现了模型的压缩和加速, 并一定程度上缓解了过拟合的问题。

根据剪枝粒度的不同, 剪枝分为非结构化剪枝和结构化剪枝, 结构化剪枝和非结构化剪枝示意如图 1 所示。非结构化剪枝分为权重剪枝和神经元剪枝, 是细粒度的剪枝方法, 对每一个权重的重要性进行评估并分别移除, 因此可以更好地平衡性能和压缩率, 但计算量通常较大; Lecun

等[12]提最佳脑损伤 (optimal brain damage, OBD) 算法，使用损失函数的 Hessian 矩阵作为参数重要性的判据对权重进行裁剪；Hassibi 等[13]进一步提出最佳脑外科医生 (optimal brain surgeon, OBS) 方法。OBD 和 OBS 都需要计算 Hessian 矩阵，这增加了内存和计算成本。

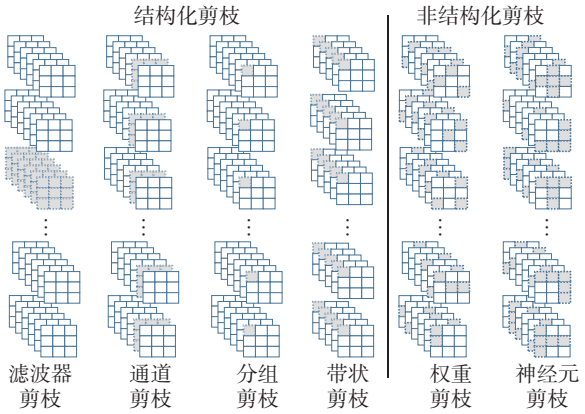
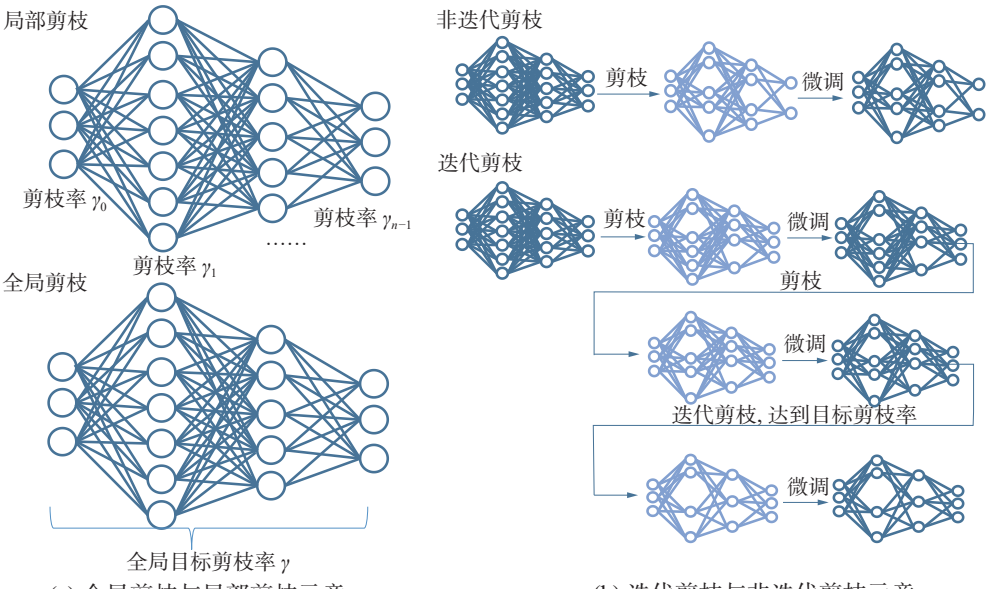


图 1　各类剪枝方法剪枝后效果示意

Fig. 1　Schematic diagram of pruning effect of various pruning methods

由于权重矩阵经过非结构化剪枝后将裁剪的权重位置的参数置零从而产生稀疏矩阵，而对稀疏矩阵进行加速计算需要特定的硬件或者软件库来支持，否则被置零的参数也需要进行计算，因此业界普遍使用结构化剪枝进行模型的压缩加速。结构化剪枝直接移除剪枝对象，随之也去除了这些参数关联的运算，是粗粒度的剪枝方法。按照剪枝的对象，结构化剪枝分为层剪枝、滤波器剪枝、通道剪枝、分组剪枝等。滤波器剪枝删去整个卷积滤波器；Li 等[14]、He 等[15]分别使用 L_1 范数、L_2 范数作为判定标准对模型进行滤波器级别的剪枝。通道剪枝对一组卷积中卷积核的某些通道进行剔除；Liu 等[16]提出了 Network Slimming 方法，对批归一化层引入缩放因子进行 L_1 稀疏化，将较小的通道进行剔除；Luo 等[17]采用贪婪策略修剪下一层激活值影响最小的通道，提出了 Thinet 方法。分组卷积将一组卷积中卷积核相同位置的权重裁剪，当多个滤波器具有相同稀疏模式时，卷积滤波器可表示为一个细化的稠密矩阵，从而能够借助现有的矩阵计算加速库进行计算，Wen 等[18]通过 group lasso 正则化来学习网络结构的稀疏性；Lebedev 等[19]分组修剪卷积核张量将卷积转化为稀疏矩阵的乘法运算进行加速。虽然结构化剪枝通过直接移除整个通道或层的参数来达到模型压缩加速的目的，不需要额外的硬件和软件库的设计，但是由于剪枝粒度较粗，容易造成性能严重损失，因此结构化剪枝的研究重点在于如何平衡性能和压缩率。

根据剪枝率的作用范围，剪枝还可以分为全局剪枝和局部剪枝。全局剪枝是对整个网络模型指定一个全局剪枝率，按照一定的标准对所有参数进行裁剪使得网络模型满足整体的剪枝率；局部剪枝则是指定某一层的剪枝率，使得这一层被裁剪的参数量满足剪枝率的要求。此外，根据剪枝是否一次完成可以将剪枝分为迭代剪枝和非迭代剪枝，其示意如图 2 所示。



(a) 全局剪枝与局部剪枝示意　　(b) 迭代剪枝与非迭代剪枝示意

图 2　全局剪枝、局部剪枝、迭代剪枝、非迭代剪枝示意

Fig. 2　Schematic diagram of global pruning , local pruning, iterative pruning and non-iterative pruning

剪枝不仅可以用来进行模型的压缩, 还可以加速训练过程。对训练过程中计算的梯度进行剪枝可以降低反向传播阶段的计算量, 从而起到了加速训练的效果。Ye 等[20] 通过基于分布的阈值确定 (distribution based threshold determination, DBTD) 方法计算过滤阈值, 结合随机剪枝算法对特征值梯度裁剪以降低反向传播过程的计算量, 加速了训练的过程。

总结之前的研究, 得到剪枝的优点主要包括:

1) 降低计算量和内存消耗: 剪枝通过减少神经网络中的参数量来减少计算量和内存消耗, 使网络轻量化从而便于在嵌入式或移动端等资源受限的环境中部署。

2) 降低过拟合风险: 剪枝可以起到稀疏正则化作用, 在合理的剪枝率设置情况下, 剪枝通过去除神经网络中冗余的连接提高模型的泛化能力, 降低过拟合风险, 提高模型性能。

剪枝的缺点主要包括:

1) 训练复杂度提升: 由于剪枝可能会造成模型精度的下降, 而且一些方法[21] 在训练过程中逐步提高剪枝率, 因此需要对神经网络迭代进行剪枝和训练, 消耗更多的计算资源和时间。

2) 人工超参数调节: 剪枝率、剪枝阈值等超参数一般需要人为指定, 这需要算法工程师丰富的经验。较低的剪枝率可能压缩比达不到部署的要求; 较高的剪枝率可能导致模型精度的严重下降。

1.2.2　量化

量化是指降低网络参数的位宽来压缩模型和高效计算。可量化的网络参数包括权重、激活、梯度和误差等, 各类网络参数量化后的位宽既可以统一, 也可以采用一定的策略组合不同的位宽来平衡压缩率与精度。量化后的数据分布可以是均匀的也可以是非均匀的; 均匀量化的数据分布间隔是统一的, 实现较为简单, 但是容易受到量化前数据分布中的离群值的影响产生较大的精度损失; 非均匀量化基于对数分布或聚类方法, 在一定程度上降低量化带来的精度损失。

根据量化后参数的位宽, 量化分为 INT8 量化、极低比特量化、混合精度量化等, 目前业界较为常用的量化方式为 INT8 量化。Tim 等[22] 将 32 位的梯度和激活值量化为特殊的 8 位浮点数实现并行加速; Jacob 等[23] 引入伪量化来模拟量化过程的误差并将权重和激活值量化为 8 位定点数; 低比特量化是基于二值化神经网络[24] 的改进, 典型工作有二值量化[25-27] 和三值量化[28-30], 极低比特量化中乘法运算可以采用移位运算来替代, 极大程度上压缩加速了模型并简化了硬件设计[31]; 极端追求高压缩比和速度而使用极低比特量化会造成模型精度的严重下降, 使用混合精度可以平衡压缩加速和模型精度, Asit 等[32] 提出了宽低精度网络 (wide reduce-precision network, WRPN) 分别将权重和激活量化到 2 bit 和 4 bit; Wang 等[33] 提出混合精度硬件感知自动量化 (hardware-aware automated quantization with mixed precision , HAQ), 引入强化学习自动找出每一层最合理的量化位宽; Zhang 等[34] 提出 Learned quantization(LQ-nets) 使量化器与网络联合训练自适应调节量化位宽。除上述量化方法以外, Gong等[35] 提出使用聚类方式量化参数数量庞大的模型, 将 k-means 聚类用于量化全连接层参数。

根据量化是否涉及训练, 量化分为训练后量化 (post-training quantization , PTQ) 和量化感知训练 (quantization-aware training , QAT), 2 种量化的一般流程如图 3 所示。PTQ 方法[27-29,32] 在对训练好的模型进行量化之后不再微调; 而 QAT 方法[23,25-26,36] 中量化因子可以在训练中动态调整大小以达到最小的精度损失, 因此一般比 PTQ 保持更高的精度。2 类量化方法各有优缺点, PTQ 方法精度损失较大, 但不需要数据或者只要少量的校验数据集进行校准, 常用于非监督学习、小批量数据集的任务中; QAT 方法量化模型效果好, 但需要训练过程复杂。

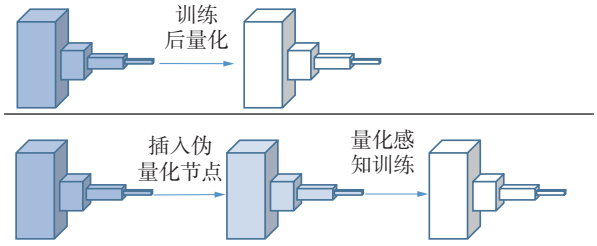


图 3　训练后量化、量化感知训练流程示意

Fig. 3　Schematic diagram of quantization and quantization perception training after training

此外, 量化还可以用于分布式训练加速, 对训练过程中产生的梯度进行量化可以在多级通讯传输梯度时减低传输带宽的压力, 能够更快地将梯度数据传输完毕。文献 [37-39] 都对模型训练过程中的梯度进行量化并且补偿量化误差, 降低通讯延时并加快了收敛速度。

总结之前的研究, 发现量化的优点主要包括:

1) 节约存储资源: 量化降低参数位宽、节约存储资源。以 INT8 量化[22-23] 为例, 量化后模型参数从 32 位浮点数压缩为 8 位整数, 模型占用存

储量理论上缩小为原来的 1/4, 这大大节约了存储资源。

2) 加速模型计算: 在 INT8 量化[22-23] 中, 量化后计算量降低为原来的 1/4, 并且处理器进行整数计算的速度快于浮点数运算速度, 因此模型推理速度有效提升。对于二值量化[25-27] 和三值量化[28-29], 浮点数乘法器可以使用更简单快速的运算器来替代 (如 XNOR 或 pop-count 逻辑门), 这不但提高了运算速度, 而且便于在计算资源受限的移动端或嵌入式设备部署模型。

3) 降低功耗: 一方面, 模型计算量的降低节约了功耗; 另一方面, 模型存储量的减少降低了访存次数和对访存带宽的需求, 这降低了 DRAM 访存带来的延迟和功耗。

量化的缺点主要包括:

1) 模型精度损失: 量化后模型的权重和激活值的位宽下降, 导致模型精度下降, 因此需要根据任务需求权衡模型精度和计算效率。

2) 训练过程复杂: 为保证量化精度损失最小, 在 QAT [23,25-26,36] 的过程中需要先插入伪量化节点再进行训练, 训练过程更为复杂。

1.2.3　知识蒸馏

知识蒸馏是利用大型教师模型的知识来监督小型学生模型训练的方法。与剪枝、量化从待压缩模型中压缩得到轻量化网络模型不同的是, 知识蒸馏的轻量化模型一般需要额外单独设计, 近年来使用剪枝、量化得到的轻量化模型作为学生模型的研究日益增多, 并成为一种新的趋势, 本文将在第 3 节重点介绍。学生模型的选择是知识蒸馏的难点和关键点, 直接影响蒸馏效果, 文献 [40] 归纳了现有研究中的学生模型的选择: "学生网络通常被选择为: 1) 教师网络的简化版本, 层数更少、每层的通道也更少; 2) 保留网络结构的教师网络的量化版本; 3) 具有高效基本操作的小型网络; 4) 具有优化的全局网络结构的小型网络; 5) 与教师相同的网络。"

教师模型向学生模型传递的知识包括预测概率、特征图、注意力映射、结构特征等。知识蒸馏中的知识类型有输出特征知识、中间特征知识、关系特征知识以及结构特征知识。早在 2006 年 Buciluǎ等[41] 就已经提出使用小模型模仿大模型特征来提升小模型性能的方法, 但直到 2015 年 Hinton 等[42] 将温度因子 T 引入 softmax 函数得到软化的概率分布作为小模型的软标签进行模仿后, 知识蒸馏方法才开始得到更为广泛的关注。硬标签与软标签示意如图 4 所示。Mirzadeh 等[43] 发现, 并非教师模型性能越高对学生模型的学习越有利, 当教师模型和学生模型存在容量差异过大时, 学生模型直接模仿教师网络的输出特征知识往往效果不佳, 为此他们提出了助教策略来缩小教师模型与学生模型之间的容量差距以取得更好的蒸馏效果。教师模型的输出特征知识提供的信息有限, 教师模型的中间层同样也含有重要的知识, Romero 等[44] 提出了 FitNets 方法以训练在当时难以优化的深层神经网络, 首次使用了教师模型的特征图作为学生模型的模仿对象; Zagoruyko 等[45] 基于 FitNets 使用注意力图代替特征图, 提出了注意力转移 (attention transfer) 方法。中间特征知识往往凭经验选取, 并且仅使学生模型的中间层模仿相对应的教师模型的中间层; Junho 等[46] 将 2 个特征层的内积定义为 FSP 矩阵并以此表示这 2 个层之间的关系, 学生模型通过教师模型的解决方案流程 (flow of the solution process, FSP) 矩阵学习其关系特征, 首次引入了关系特征知识; Xu 等[47] 综合了输出特征、中间特征和关系特征作为结构特征知识提出了整体知识蒸馏 (integral knowledge distillation, IKD), 将知识蒸馏从分类问题拓展到人体姿态估计问题。随着研究的开展, 知识的选取从最初的输出特征发展到中间特征、关系特征以及融合前 3 种特征的结构特征, 知识蒸馏的应用也从最初的分类任务拓展到其他任务中。

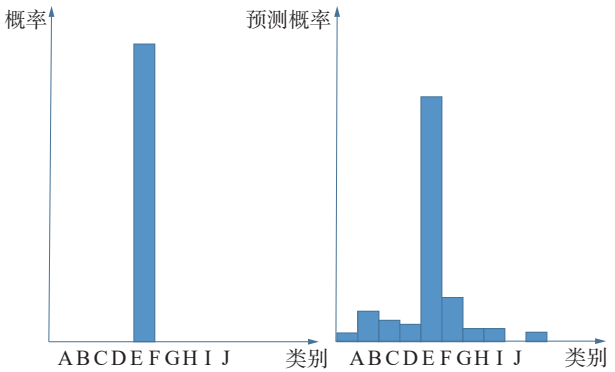


图 4　硬标签与软标签示意

Fig. 4　Hard label and soft label

根据教师模型是否和学生模型一起更新参数可以分为离线蒸馏 (offline distillation)、在线蒸馏 (online distillation) 以及自蒸馏 (self distillation)。离线蒸馏通常需要先得到训练完备的高性能教师模型然后再进行知识蒸馏得到轻量化的学生模型, 因此训练过程一般分为 2 个阶段; 在离线蒸馏学习的过程中教师模型只进行推理而不更新参

数, 大部分研究[41-45]使用离线蒸馏学习范式。离线蒸馏方法简单易行, 但是需要高性能的教师模型, 而且不能保证教师模型与学生模型匹配。针对离线蒸馏的训练周期长、师生模型容量差距[43]的问题, 在线蒸馏对高性能教师模型没有严格要求, 蒸馏过程中教师模型与学生模型同步更新参数, 因此只需要一个阶段就可以完成训练。自蒸馏是学生模型从自身进行蒸馏的特殊的在线蒸馏方法, 无需额外构造教师模型, 通过将深层信息回传给浅层指导训练[48]或迭代前模型指导迭代后模型训练[49]的方式实现模型性能的提升。离线蒸馏、在线蒸馏和自蒸馏如图 5 所示。

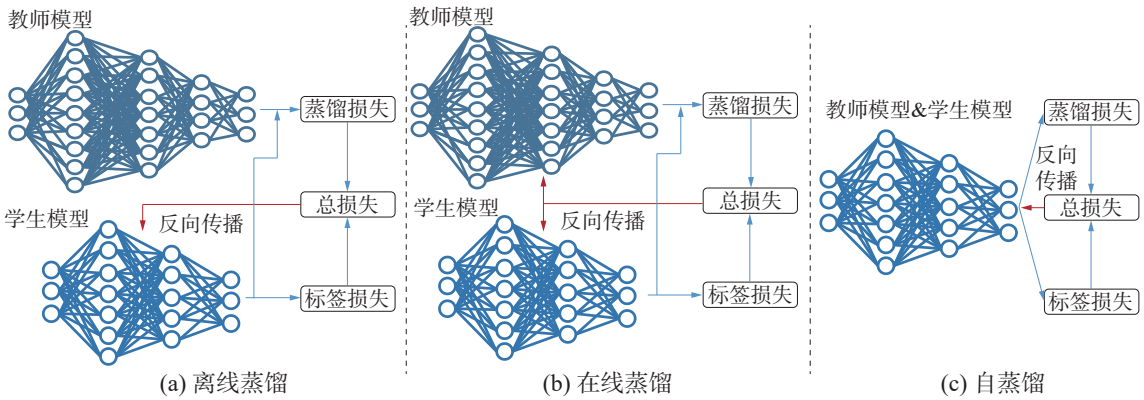


图 5　离线蒸馏、在线蒸馏、自蒸馏示意

Fig. 5　Schematic diagram of off-line distillation, on-line distillation and self-distillation

总结之前的研究, 我们发现知识蒸馏的优点主要包括:

1) 提高学生模型精度: 学生模型同时学习教师模型的知识和标签训练后的精度往往高于直接使用标签训练的小模型的精度, 甚至能够达到接近教师模型的精度, 有"青出于蓝而胜于蓝"的效果。

2) 无需人工设置压缩率: 与剪枝、量化等轻量化算法通过压缩原始模型得到轻量化模型不同, 知识蒸馏一般需要额外构造轻量化模型作为学生模型, 不需要人工设置压缩率。

3) 便于与其他轻量化算法结合: 经过知识蒸馏后得到的高性能的轻量化模型还能继续使用剪枝、量化等方法进行进一步的压缩加速以达到更高的压缩率; 或者使用知识蒸馏恢复压缩后模型的精度。

知识蒸馏的缺点主要包括:

1) 训练过程复杂: 离线蒸馏一般需要先训练得到高性能教师模型然后再使用教师模型对学生模型进行知识蒸馏, 这增大了训练成本和时间。除此之外, 知识蒸馏需要针对不同的任务构造合适的损失函数以达到知识传递的效果, 这增加了训练的计算复杂度。

2) 使用场景局限: 知识蒸馏使用场景受限有网络结构和应用场景两方面原因。在网络结构方面, 知识蒸馏算法通常用于压缩结构化的神经网络如卷积神经网络, 但对于非结构化神经网络如循环神经网络或图神经网络知识蒸馏算法需要针对其特殊结构改进; 在应用场景方面, 由于知识的定义和蒸馏损失函数的构造上的困难, 知识蒸馏方法最广泛的应用仍是在分类任务。

3) 模型匹配与超参数调节问题: 当教师模型和学生模型容量差异过大时学生模型的性能可能反而下降[46], 因此师生模型的匹配是知识蒸馏面临的挑战。此外, 知识蒸馏中的超参数如温度因子、权重因子的设定和调节也需要一定的经验。

1.3　模型压缩加速评价指标

1.3.1　压缩程度评价指标

参数减少量是评价模型参数量减少的绝对值, 压缩率是评价模型参数量减少的相对值。前者为模型压缩后减少的参数量, 后者为原模型参数的存储量与压缩后模型参数的存储量的比值。压缩率定义为

$$\phi(M, M') = \frac{\mu}{\mu'} \tag{1}$$

式中: μ为压缩前模型的存储量, μ'为压缩后模型的存储量。

1.3.2　加速效果评价指标

浮点运算数 (floating point operations, FLOPs) 是评价模型计算复杂度的指标, 可以用来衡量模型加速效果的绝对值。对于计算机视觉任务而言, 每秒传输帧数 (frames per second, FPS) 是衡量模型每秒处理的图片数的重要指标。此外, 在同一硬件设备上的推理时间可以用来衡量模型加速的相对值, 加速率定义为

$$\varphi(M,M')=\frac{v}{v'} \tag{2}$$

式中：v为压缩前模型的推理时间，v'为压缩后模型的推理时间。

1.3.3　模型性能评价指标

不同任务对模型性能有不同的评价指标，模型的性能不严重损失是模型压缩的一个重要前提。常见的计算机视觉任务有分类、目标检测、语义分割等，详细介绍各个任务的性能评价指标并不是本文的重点，因此在这里只列举分类任务和目标检测任务的常用指标。

在分类任务中常使用准确率和错误率来评判模型的性能，准确率是指预测结果中排名靠前的k个值的准确率，k一般取 1 或 5；相反的，错误率表示预测结果中排名靠前的k个值的错误率。

在目标检测任务中常使用 mAP@k 作为模型性能的评价指标，平均精度 (mean average precision, mAP) 表示在交并比 (intersection over union, IoU) 阈值为k的条件下各个检测类平均精度 (average precision , AP) 的平均值。AP 是综合了准确度 (precision) 和召回率 (recall) 的每个检测类别的评价指标，设某一类别 C 的精度 (precision) 与召回率 (recall) 的函数为$P(x)$，则 AP 定义为

$$A_{\mathrm{PC}}=\int_0^1 P(R)\mathrm{d}R \tag{3}$$

IoU 是衡量网络模型预测框与真实框重合程度的指标，定义为

$$I_{\mathrm{oU}}=\frac{S_{\cap}}{S_{\cup}} \tag{4}$$

式中：模型预测框 P 与真实框 GT 重叠区域的面积为$S_{\cap}$，模型预测框 P 与真实框 GT 覆盖的总面积为$S_{\cup}$。

2　神经网络压缩联合优化方法

尽管各种压缩方法都从独特的角度对模型压缩和加速做了深入的研究，但目前大多数研究都局限于各自的方向中，而涉及到如何有效结合各种方法进行联合优化的问题时，因其问题的复杂度较高还未形成成熟的研究。本文总结大量文献所提出的联合压缩加速方法，将其按照组合类型分为“知识蒸馏+剪枝”、“知识蒸馏+量化”和“剪枝+量化”3 类，将在下文一一阐述。

2.1　知识蒸馏+剪枝

知识蒸馏与剪枝联合进行模型压缩加速主要应用于卷积神经网络[50-61]，近年来也出现了对 Transformer 模型[62-63]及强化学习代理网络[64,65]的压缩加速。结合方式主要分为先剪枝后蒸馏[50-58,65]、先蒸馏后剪枝[51,59]、边蒸馏边剪枝[60-63]3 类。

本文调研的剪枝与知识蒸馏相结合的文献中所使用的剪枝、知识蒸馏方法及两者之间的结合方式统计如表 1 所示。

表 1　本文所整理的“知识蒸馏+剪枝”文献方法汇总

Table 1　Summary of the literature methods of “knowledge distillation + pruning” organized in this paper

年份	文献标题	剪枝方法	知识蒸馏方法	结合方式
2018	Compression of Acoustic Model via Knowledge Distillation and Pruning	全局权重剪枝方法	KD[42]	B
2019	Learning Slimming SSD through Pruning and Knowledge Distillation	Network Slimming[16]	DML(Deep Mutual Learning)	A-2
2019	Semantic Segmentation Optimization Algorithm Based on Knowledge Distillation and Model Pruning	基于BN层L_1 norm的非结构化通道剪枝	KD[42]	C-2
2020	Automatic Optimization of super Parameters Based on Model Pruning and Knowledge	Level pruner pruning	KD[42]	A-2
2020	Knapsack Pruning with Inner Distillation	*Knapsack Pruning	*IKD	A-2
2020	The Optimization Method of Knowledge Distillation Based on Model Pruning	L_1 Filter pruning、L_2 Filter pruning、Lottery Ticket pruning、Level pruning、AGP pruning、FPGM pruning	KD[42]	A-2
2020	A Lossless Lightweight CNN Design for SAR Target Recognition	逐层渐进通道剪枝	KD[42]	A-3
2021	Boosting Lightweight CNNs Through Network Pruning and Knowledge Distillation for SAR Target Recognition	*基于注意力的全局通道结构化剪枝	*基于桥接的知识蒸馏	A-2
2021	Improving the Accuracy of Pruned Network Using Knowledge Distillation	基于L_1 norm的滤波器剪枝	KD[42]	D

续表 1

年份	文献标题	剪枝方法	知识蒸馏方法	结合方式
2021	Combining Weight Pruning and Knowledge Distillation for CNN Compression	*基于APoZ的迭代神经元剪枝	*基于师生特征层余弦相似性的知识蒸馏	A-1
2021	HKDP: A Hybrid Approach on Knowledge Distillation and Pruning for Neural Network Compression	*基于SGD的全局迭代剪枝	*基于MSE的阶段知识蒸馏	B
2021	Joint-DetNAS: Upgrade Your Detector with NAS, Pruning and Dynamic Distillation	*基于L_1 norm的通道剪枝、添加层、重排层的NAS方法	*Dynamic Distillation: 找到最佳匹配教师	C-2
2021	Joint Structured Pruning and Dense Knowledge Distillation for Efficient Transformer Model Compression	*DISP	*DKD	C-1
2021	Knowledge from the Original Network Restore a Better Pruned Network with Knowledge Distillation	基于L_1 norm的非结构化剪枝	KD[42]; AT、SP	C-1
2021	Learning Slimming SAR Ship Object Detector Through Network Pruning and Knowledge Distillation	Network Slimming[16]	*FIR-KD	A-2
2022	A Knowledge-Distillation-Integrated Pruning Method for Vision Transformer	*PDIP	*基于自注意力机制、输出特征向量的知识蒸馏	C-1
2022	An Efficient Method for Model Pruning Using Knowledge Distillation with Few Samples	Network Sliming[16]	*PFDD	A-2
2022	LRP-based Policy Pruning and Distillation of Reinforcement Learning Agents for Embedded Systems	*LRP-based policy pruning	Policy Distillation[64]	A-2
2022	PPCD-GAN: Progressive Pruning and Class-Aware Distillation for Large-Scale Conditional GANs Compression	*基于PP-Res块的渐进权重剪枝	*类感知蒸馏	C-2
2022	Prune Your Model Before Distill It	非结构化权重剪枝	KD[42]	A-1

注: 1) 标注*表示文章新提出的方法; 2) 结合方式编号如下: A-1: 先剪枝后蒸馏, 学生模型单独构造, 教师模型为剪枝后模型; A-2: 先剪枝后蒸馏, 学生模型为剪枝后模型, 教师模型为剪枝前模型; A-3: 先剪枝后蒸馏, 学生模型为剪枝后模型, 教师模型单独构造; B: 先蒸馏后剪枝, 对知识蒸馏得到的学生模型进行剪枝; C-1: 边剪枝边蒸馏, 学生模型为剪枝后模型, 教师模型为剪枝前模型; C-2: 边剪枝边蒸馏, 学生模型为剪枝后模型, 教师模型单独构造; D: 先蒸馏再剪枝后蒸馏: 第1次蒸馏中, 学生模型为教师模型的通道减半的紧凑版本; 第2次蒸馏中, 学生模型为剪枝后模型, 教师模型为剪枝前模型。

2.1.1　先剪枝后蒸馏

先剪枝后蒸馏方法按照知识蒸馏中师生模型的选择细分为 3 种方法, 如图 6 所示。

深度神经网络在剪枝后精度往往会下降, 微调能够在一定程度上恢复剪枝后的模型精度, 但当裁剪掉过多的参数时, 剪枝后模型容量会大幅度下降以至于无法恢复到原来的精度。学生网络的设计是知识蒸馏的难点和关键点, 直接影响知识蒸馏后模型的性能。文献 [66] 通过实验证明: 学生模型和教师模型网络结构的相似性与最终的蒸馏性能呈正相关, 即教师模型与学生模型结构越相似, 蒸馏效果越好。基于上述现象, 可以对教师模型剪枝来构造结构相似的学生模型, 使用知识蒸馏替代微调的过程弥补剪枝造成的精度损失, 实现剪枝和知识蒸馏方法的互补结合。在这

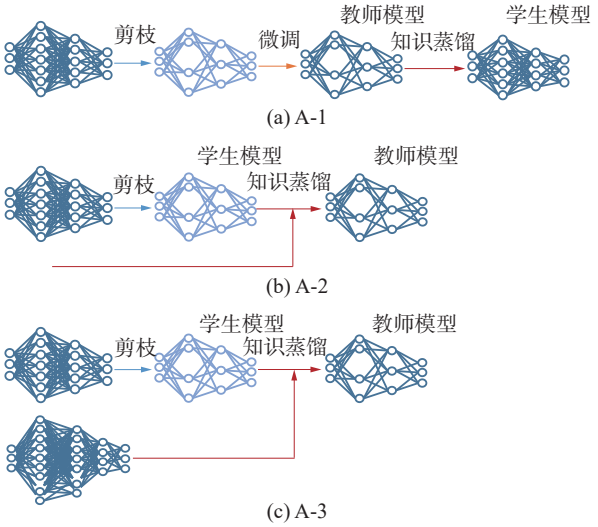


图 6　先剪枝后蒸馏方法分类示意

Fig. 6　Schematic diagram of pruning before distillation method classification

类方法的知识蒸馏过程中，通常使用剪枝后模型作为学生模型，剪枝前模型作为教师模型以达到更好的知识传递效果[50-55,65]，但也有个别方法使用额外构造的高精度教师模型[54]。很多研究[16,50-51,65,67-68]组合已提出的剪枝和知识蒸馏方法来进行联合压缩和加速，如 Li 等[67]使用未剪枝的单次多检测器 (single shot multiple detector, SSD) 作为教师模型对 Network Slimming 剪枝[16]后 SSD 目标检测模型进行在线蒸馏以提高检测精度。针对手动调参的局限性，Wu 等[50]在剪枝和知识蒸馏联合压缩的基础上引入了超参数自动优化算法，利用现有的模型剪枝技术设计学生模型并在知识蒸馏过程中使用 NNI(neural network intelligence) 自动参数调整工具自动优化知识蒸馏中的 3 个重要的超参数：温度因子、权重因子及剪枝率，以自动得到最优的实验压缩效果，克服手动调参的局限性。Xu 等[65]在深度强化学习中结合剪枝和策略蒸馏[68]，对深度 Q 网络 (deep Q-network, DQN) 代理网络进行压缩以部署在资源有限的嵌入式系统中。Wu 等[51]探究了 6 种不同的剪枝方法与知识蒸馏的结合，实验证明了先剪枝后蒸馏的方法的效果与剪枝方法、模型结构、数据集大小相关，需要多方面联合优化，为后续针对特定应用设计剪枝与知识蒸馏联合压缩方法提供了理论支持。

上述研究中剪枝和知识蒸馏都利用已提出的方法，许多研究都针对特定的应用场景设计了新的剪枝或知识蒸馏方法以实现更好的模型压缩加速效果。在模型中有非连续卷积 (如倒残差结构[9]) 的情况下，并不能直接应用常规剪枝方法，Aflalo 等[52]将背包问题表述为一种新的剪枝方法，并提出了一种新的知识蒸馏方法内知识蒸馏 (inner knowledge distillation, IKD) 用于弥补剪枝造成的精度损失；IKD 方法利用剪枝前后师生结构相似性使用网络内层的特征图作为知识引导学生模型学习教师模型的内层映射；在他们的方法中，先对原始模型背包剪枝得到轻量化的学生模型，再使用原始模型作为教师模型进行 IKD 恢复精度。针对少样本知识蒸馏的场景，Zhou 等[53]提出了渐进特征分布蒸馏 (progressive feature distribution distillation, PFDD) 旨在恢复少样本场景下模型剪枝造成的精度损失；PFDD 方法使用特征图作为蒸馏知识，使用中间特征层的 Gram 矩阵计算最大平均差异 (maximum mean discrepancy, MDD) 作为蒸馏损失来训练学生模型以匹配教师模型的特征图，并使用渐进式训练策略重点关注损失函数中占比最大的损失，以充分利用样本信息。在该方法中，先使用 Network Slimming 剪枝[16]得到学生模型，再使用原始模型作为教师模型对学生模型进行 PFDD 以恢复精度损失。

广泛应用于军事和民用的孔径合成雷达 (synthetic aperture radar , SAR) 自动目标识别 (automatic target recognition , ATR) 对网络模型的实时性和准确性有严格要求，对于 SAR ATR 场景部署的网络模型往往需要轻量化网络设计、剪枝、蒸馏等多种方法联合进行压缩加速，且需要针对其特殊性对传统压缩方法进行改造。Zhang 等[54]联合通道剪枝、知识蒸馏和权重共享实现了 SAR 目标检测模型的压缩加速，他们首先使用逐层通道迭代剪枝得到轻量化模型，然后使用高性能预训练模型作为教师模型对剪枝后模型知识蒸馏，最后使用 k-means 实现权值共享进一步压缩模型。与先前方法不同的是，该方法知识蒸馏阶段使用高性能 SAR 目标检测网络作为教师模型，而非剪枝前的原始模型。Chen 等[55]针对 SAR ATR 场景，使用稠密连接和非对称卷积设计了一种新型的 DC-ACM YOLOv3(densely connected and ACM-assisted YOLOv3) 目标检测器，并命名为 Tiny YOLOv3-Lite，然后使用通道剪枝降低参数量，并提出了一种新的知识蒸馏方法特征相互关系知识蒸馏 (feature inter-relationship knowledge distillation, FIR KD) 进一步提升模型精度。Wang 等[56]使用 SE 模块构建了轻量化的 CA-Net，提出了一种基于注意力的结构化剪枝方法用于压缩 CA-Net，并提出了一种基于桥接的知识蒸馏方法恢复剪枝后 CA-Net 的精度。

现有的先剪枝后蒸馏的联合压缩方法[50-56,65]发展趋势为从直接使用已提出的剪枝、知识蒸馏方法进行联合发展[16,50-51,65,67-68]为针对特定应用场景设计相应的剪枝、知识蒸馏方法进行联合[51,53-56]，都使用剪枝后的模型作为学生模型进行知识蒸馏，剪枝的目的是解决知识蒸馏中学生模型的构造问题，蒸馏的目的则是弥补由于剪枝造成的精度损失，2 种方法联合发挥了更好的压缩效果。

教师模型的选择直接决定了知识蒸馏的效果，盲目组合高性能教师模型和轻量化的学生模型往往效果不尽人意，当复杂教师模型和简单学生模型之间容量差异过大时，知识蒸馏效果往往不佳[43]。文献 [69] 也表明，当学生模型没有足够的函数拟合能力时，受过较少训练的教师模型蒸馏效果更好，而使用剪枝后模型作为学生模型会进一步拉大教师模型与学生模型的容量差距，这可能影响知识蒸馏的效果。基于以上观点，文献 [57-58] 使用剪枝后的模型作为教师模型以降低

师生模型容量差异。Aghli 等[57] 提出了一种联合剪枝和知识蒸馏用于压缩具有残差结构的神经网络的方法, 针对残差块连接的特点, 使用激活值的平均零点百分比 (average percentage of zeros , APoZ) 作为神经元重要性判据进行剪枝; 并提出了一种新的知识蒸馏方法, 将剪枝后的模型作为教师模型、额外构造的轻量化的网络作为学生模型, 通过分层最小化师生特征图之间的余弦相似度损失以使得学生模型模仿教师模型的特征图。Park 等[58] 提出了 Prune then distill 方法, 使用剪枝后的模型作为教师模型对额外构造的轻量化学生模型进行知识蒸馏。

上述先剪枝后蒸馏的方法中师生组合方式截然不同, 前者[50-27,65] 使用剪枝后的模型作为知识蒸馏中的学生模型, 而后者[57-58] 使用剪枝后的模型作为知识蒸馏中的教师模型, 产生上述不同的根本原因在于指导思想不同。前者使用知识蒸馏来弥补剪枝造成的精度损失, 其工作中心在于尽可能提高剪枝率; 而后者使用剪枝来降低知识蒸馏中师生模型的容量差异, 其工作中心在于尽可能提高蒸馏效果。

2.1.2　先蒸馏后剪枝

先蒸馏后剪枝方法通常对知识蒸馏得到的学生模型进一步压缩以得到更加轻量化的模型, 如图 7 所示。

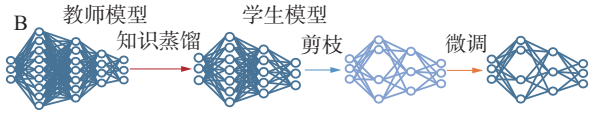


图 7　先蒸馏后剪枝方法分类示意

Fig. 7　Schematic diagram of classification by distillation before pruning

文献 [59,70] 使用先蒸馏后剪枝的方法来联合 2 种压缩方法, 即先进行知识蒸馏得到高精度的学生模型, 然后对学生模型进行剪枝以进一步压缩模型。Li 等[70] 联合剪枝和知识蒸馏压缩语音识别系统的声学模型, 即先使用知识蒸馏得到高精度的学生模型, 然后对学生模型进行权重剪枝以进一步压缩模型参数, 最后对剪枝后的模型进行微调恢复模型精度, 剪枝和微调迭代进行以最大程度压缩模型。Che 等[59] 提出了联合剪枝和知识蒸馏的方法知识蒸馏和剪枝联合方案 (hybrid approach on knowledge distillation and pruning, HKDP), 使用一种分阶段知识蒸馏方法以得到高性能的学生模型, 然后使用一种基于随机梯度下降 (stochastic gradient descent, SGD) 的迭代剪枝方法进一步压缩学生模型; 在分阶段知识蒸馏中, 学生模型在一轮训练中只优化某一阶段的网络参数, 其他参数冻结; 在 SGD 迭代剪枝中, 每一轮剪枝后微调以恢复模型精度, 迭代修剪直到模型被压缩到目标剪枝率。文献 [59,70] 在剪枝阶段使用硬标签进行微调, 缺乏教师模型的监督信息, 因而模型性能恢复有限。Prakosa 等[71] 提出了先蒸馏再剪枝后蒸馏的方法, 即先知识蒸馏得到高性能的学生模型, 再剪枝进一步压缩学生模型, 最后使用知识蒸馏恢复剪枝后学生模型的精度。

先剪枝后蒸馏[50-58,65] 和先蒸馏后剪枝[67,72] 2 类方法在联合顺序上截然相反, 其根本原因在于算法设计的出发点不同。先剪枝后蒸馏旨在利用知识蒸馏弥补由于剪枝造成的精度损失, 而先蒸馏后剪枝旨在进一步压缩本就轻量级的学生模型。

2.1.3　边剪枝边蒸馏

边剪枝边蒸馏方法按照知识蒸馏中师生模型的选择细分为 2 种方法, 如图 8 所示。

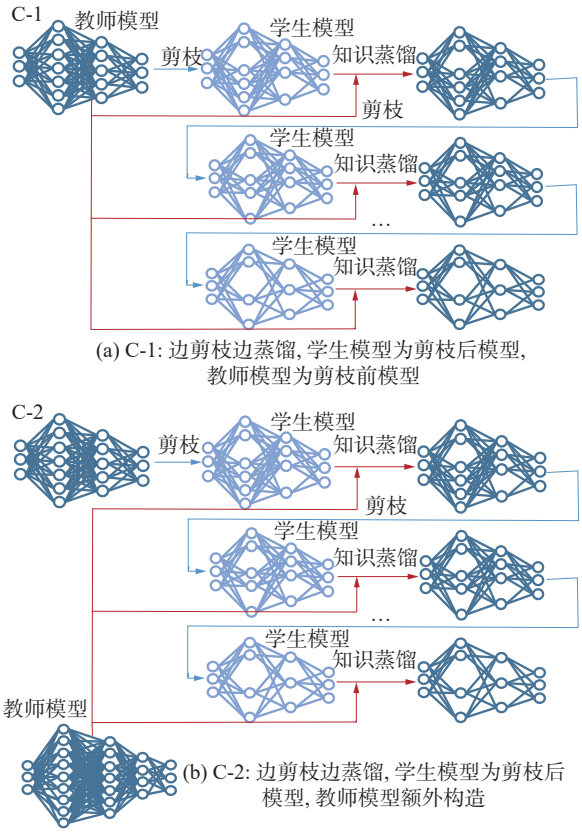


图 8　边剪枝边蒸馏分类示意

Fig. 8　Schematic diagram of distillation classification while pruning

上述研究所提出的方法中, 无论是先剪枝后蒸馏[50-58,65]、先蒸馏后剪枝[67,72] 还是先蒸馏再剪枝后蒸馏[71], 其剪枝和蒸馏阶段分别进行, 学生模型在剪枝的过程中缺乏高精度模型的监督, 使得模型剪枝后往往难以保持精度, 且训练过程复杂、周期长、计算成本高, 因此有研究[60-63] 提出了端到端的压缩加速方案, 即边剪枝边蒸馏。在边剪枝边蒸馏的方案中, 所使用的剪枝方案一般为

迭代剪枝，知识蒸馏中学生模型一般都是剪枝后的模型，教师模型可以是剪枝前的模型[60,62-63]也可以是额外构造的高性能模型[61]，使用知识蒸馏替代每轮剪枝后的微调过程，实现剪枝和知识蒸馏的结合。Chen 等[60]提出了边剪枝边蒸馏的联合压缩方法，在迭代剪枝的一轮修剪中，先剔除模型中冗余参数并将其作为学生模型，再使用原始模型作为教师模型进行知识蒸馏以恢复模型精度。Yao 等[61]结合知识蒸馏和通道剪枝来压缩语义分割网络，所使用的通道剪枝方法使用批量归一化 (batch normalization, BN) 层的 L_1 范数作为评价通道重要性指标，边剪枝边蒸馏逐步压缩模型。Cui 等[62]结合结构化剪枝和密集知识蒸馏提出了联合模型压缩 (joint model compression, JMC) 方法用于大型 Transformer 压缩，提出了一种新的直接重要性剪枝感知结构化剪枝 (direct importance-aware structured pruning , DISP) 方法，和一种采用多对一层映射策略的密集知识蒸馏 (dense knowledge distillation , DKD) 以更全面地利用分层语言知识进行蒸馏；JMC 方法将 DISP 和 DKD 联合，在一轮迭代剪枝中，先使用 DISP 修剪层数较少的学生模型，再使用 DKD 恢复模型精度，不断迭代直到达到目标剪枝率。由于自注意力机制的引入，视觉自注意力模型 (vision transformer, ViT) 模型结构复杂、参数冗余度低，直接使用卷积神经网络 (convolutional neural network, CNN) 的剪枝方法来压缩 ViT 会造成精度严重损失，Xu 等[63]将知识蒸馏引入 ViT 的剪枝过程中，提出了知识蒸馏集成剪枝 (knowledge dsitillation integrated pruning , KDIP)，在待剪枝层之前引入一个重要性得分学习模块来评估删减参数矩阵的每个维度，以确保删减模型冗余的维度，并使用知识蒸馏弥补剪枝引起的精度损失，反复迭代修剪直到达到目标剪枝率。

综上所述，知识蒸馏作为一种高性能轻量化模型的训练方法可以有效弥补剪枝造成的精度损失，同时剪枝可以作为知识蒸馏中学生模型[50-56,65]或教师模型[57-58]的构造来源。绝大多数的研究都针对特定场景对剪枝或蒸馏方法进行了优化，但是剪枝和知识蒸馏的结合方式基本分为先剪枝后蒸馏、先蒸馏后剪枝、边剪枝边蒸馏 3 类。

2.2　知识蒸馏+量化

知识蒸馏与量化联合进行模型压缩加速结合方式主要有先量化后蒸馏[72-78]和边量化边蒸馏[74,77]两大类。根据学生模型的来源，先量化后蒸馏这一类可以分为学生模型单独设计[76]和学生模型由量化得到[30,72,74-75,77-78]2 种情况，

本文调研的量化与知识蒸馏相结合的文献中所使用的量化、知识蒸馏方法及两者之间的结合方式统计如表 2 所示。

表 2　本文所整理的“知识蒸馏+量化”文献方法汇总

Table 2　Summarizes the literature methods of "knowledge distillation + quantization" organized in this paper

年份	文献题目	量化方法	知识蒸馏方法	结合方式
2017	Apprentice: Using Knowledge Distillation Techniques to Improve Low-precision Network Accuracy	三值量化[48](2W8A); WRPN[28](4W8A)	文章提出3种方案： 方案1：DML 方案2、3：KD[42]	A-3
2018	Model Compression Via Distillation and Quantization	*DQ	*QD	B
2018	Quantization Mimic: Towards Very Tiny CNN for Object Detection	均匀量化	*基于量化后特征图的知识	A-1
2019	Empirical Analysis of Knowledge Distillation Technique for Optimization of Quantized Deep Neural Networks	三值量化	KD[45]	A-3
2019	QKD: Quantization-aware Knowledge Distillation	一种线性均匀QAT方法	CS阶段：DML TU阶段：KD[42]	A-3
2019	A Gradually Distilled CNN for SAR Target Recognition	三值量化	*GD渐进蒸馏	A-3
2020	Data-Free Network Quantization with Adversarial Knowledge Distillation	QAT方法	Adversarial Knowledge Distillation	A-2
2021	Explore a Novel Knowledge Distillation Framework for Network Learning and Low-Bit Quantization	二值/三值量化	*SKD,*DLBQ	B
2021	Lossless AI: Toward Guaranteeing Consistency between Inferences Before and After Quantization via Knowledge Distillation	QAT方法	KD[42]	A-2

续表 2

年份	文献题目	量化方法	知识蒸馏方法	结合方式
2022	Understanding and Improving Knowledge Distillation for Quantization-aware Training of Large Transformer Encoders	二值/三值QAT量化	*基于注意力输出损失的知识蒸馏	A-3

注:1)标注*表示文章新提出的方法; 2)结合方式编号如下: A-1: 先量化后蒸馏, 学生模型单独构造, 教师模型为量化后模型; A-2: 先量化后蒸馏, 学生模型为量化后模型, 教师模型为量化前模型; A-3: 先量化后蒸馏, 学生模型为量化后模型, 教师模型单独构造; B: 边量化边蒸馏: 学生模型为量化模型, 教师模型单独构造。

2.2.1　先量化后蒸馏

先量化后蒸馏方法按照知识蒸馏中师生模型的选择细分为 3 种方法, 如图 9 所示。

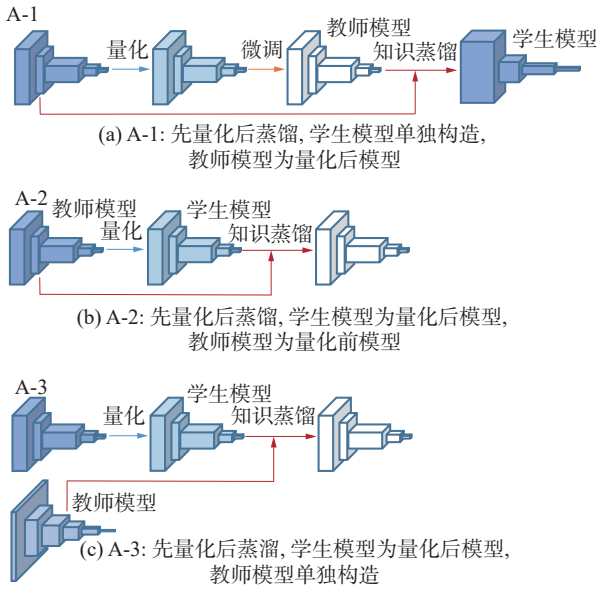


图 9　先量化后蒸馏方法分类示意

Fig. 9　Schematic diagram of quantization before distillation method classification

文献 [72] 证明知识蒸馏能够很大程度上弥补由于量化导致的模型参数位宽减少而造成的模型精度损失, 因此知识蒸馏常用来代替量化后的微调过程以获得更高精度的轻量化网络模型, 基于这种思想提出的联合优化方法往往需要先量化压缩得到低位宽模型再使用知识蒸馏弥补量化造成的精度损失, 是两阶段的模型压缩加速方法。Mishra 等[73] 首次结合知识蒸馏与量化提出了 Apprentice 方法, 针对不同的训练过程提出了 3 种方案, 论证了 3 种方案各自的优缺点并通过实验证明了方案 3 在恢复极低比特量化精度上效果最优。量化蒸馏 (quantized distillation, QD)[79] 方法在 4、8 bit 量化时具有良好的性能, 但在 2 bit 及以下量化的时候具有严重精度损失, Min 等[74] 针对 QD 在三值网络上的不足与 SAR 目标检测应用场景的特殊性, 提出了渐进蒸馏 (gradually distillation , GD) 训练方法, 设计了一种轻量化的三值微型卷积神经网络 (micro convolutional neural network, MCNN), 并使用未训练的全精度教师模型与学生模型 MCNN 进行在线蒸馏, 得到了高精度的实时 SAR 目标检测器。

文献 [40] 发现, 并非教师模型性能越高对学生模型的学习越有利, 当复杂教师模型和简单学生模型之间容量差异过大时, 学生模型直接模仿教师网络的输出特征知识往往效果不佳, 而使用量化后的低位宽网络作为学生模型会进一步拉大教师模型与学生模型的容量差距, 这严重影响知识蒸馏的效果, 因此许多研究的重点在于如何降低两者之间的差距以达到更好的知识蒸馏效果。针对模型容量差距问题和 Apprentice 方法[73] 中的初始化问题, Kim 等[75] 提出了量化感知知识蒸馏 (quantization-aware knowledge distillation, QKD) 方法使用 SS+CS+TU 3 阶段训练以得到更高精度的量化后低位宽模型, 缓解了模型容量差距和初始化问题的影响, 得到了更高效的轻量化网络。Okuno 等[76] 提出了“无损 AI”(lossless AI) 的模型压缩概念, 并提出一种知识蒸馏方法将 BN 层统计数据冻结以进行推理结果的对齐, 并使用量化前的全精度模型作为教师模型、量化后的模型作为学生模型进行知识蒸馏, 从而保证模型压缩前后推理结果之间的一致性。

大多数模型压缩方法[61,70,72,74,76-78,80] 将量化后的低位宽模型用于部署, 而 Wei 等[81] 提出的量化模仿 (quantization mimic) 方法中最终部署的是全精度的轻量化学生模型。Quantization Mimic 方法[81] 使用量化后的低位宽网络作为教师模型、全精度的轻量化网络作为学生模型以降低教师模型与学生模型之间的容量差距, 从而达到更好的知识蒸馏效果。此外, Wei 等[81] 还使用量化对教师模型和学生模型的输出特征图进行离散化以促进 2 个网络输出特征图之间的匹配, 从而降低学生模型的模仿难度。与 3.2.1 节先剪枝后蒸馏方法的师生组合类似, 上述先量化后蒸馏的方法中师生组合方式截然不同, 前者[64,73-78] 使用量化后的模型作为知识蒸馏中的学生模型, 而后者[76] 使用量化后的模型作为知识蒸馏中的教师模型。其根本原因也在于基本思想不同, 前者使用知识蒸馏弥补由于量化造成的精度损失, 以尽可能达到更低的

量化位宽，其重点在于量化压缩；后者使用量化来降低师生容量差距，以尽可能地提高蒸馏精度，其重点在于知识蒸馏。

在知识蒸馏中，一个常见的问题是如何构造学生模型使之能够与教师模型匹配从而达到更好的蒸馏效果，而量化正是构造轻量化学生模型的常见方法，现有的大多数研究[64,73-77]都使用量化的方式获得轻量化的模型作为知识蒸馏中的学生模型。文献 [77] 将使用全精度模型作为教师模型指导量化后的低位宽模型作为学生模型进行知识蒸馏的方法定义为自蒸馏量化。不同于大多数研究需要额外构造高性能的全精度教师模型，自蒸馏量化[75,77,80]在整个模型压缩的过程中只需要一个全精度模型，无需另外构造并训练一个高性能网络作为教师模型，这拓展了“量化+知识蒸馏”方法的使用范围。

如上文所述，知识蒸馏的使用场景受限于知识的定义和损失函数的构造通常应用在计算机视觉中的分类问题中，若要拓展到其他领域，需要根据应用场景定义知识和构造相应的损失函数。Kim 等[78]针对 Transformer 的特点改进了先前用于 QAT 的知识蒸馏技术，他们使用注意力图损失替代注意力得分损失并逐层引导量化的 Transformer 模型的知识蒸馏，实现了 Transformer 模型在 2 bit 及以下位宽 QAT 量化的最佳精度。

随着人们对数据隐私和安全保护意识的提高，在某些特定应用场景下用于模型压缩的数据集的获取变得越来越困难，但是对大规模冗余模型轻量化的需求仍然存在，这就需要无数据模型压缩技术。针对数据隐私和安全性导致的无数据集情况下的模型压缩问题，Choi 等[80]将对抗学习与知识蒸馏、量化相结合，提出了对抗性知识蒸馏量化方法，使用生成器合成的样本作为全精度教师模型和由教师模型量化得来的学生模型的输入，并将生成的数据与教师模型中的原始数据进行批量归一化层的匹配以保证生成器生成与原始数据相似的对抗性样本，用于学生模型的知识蒸馏。

2.2.2　边量化边蒸馏

边量化边蒸馏方法中教师模型通常为原始模型，学生模型为量化后模型，如图 10 所示。

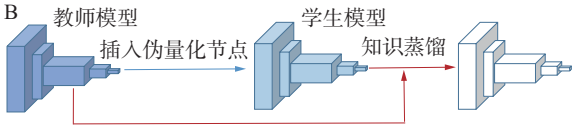


图 10　边量化边蒸馏方法分类示意

Fig. 10　Schematic diagram of the classification of side quantization side distillation method

上述量化和知识蒸馏结合的方法量化和知识蒸馏的过程分别进行，训练周期长、成本高，有研究[74,77]提出了边量化边蒸馏训练方法。这类方法中的量化一般使用 QAT，使用知识蒸馏作为量化模型感知训练的方法，实现量化和知识蒸馏的结合。Polino 等[79]分别针对量化后低位宽模型的知识蒸馏和全精度模型的量化提出了量化蒸馏 (quantized distillation, QD) 和可微分量化 (differentiable quantization, DQ)；QD 使用待压缩的模型 A 和高精度高延迟的模型 B 等 2 个模型，使用模型 B 作为教师模型对量化后的模型 A 进行迭代知识蒸馏最终得到用于部署的学生模型；DQ 则是一种的 PTQ 方法，通过在量化后微调弥补量化造成的精度损失。Si 等[77]提出了低比特量化蒸馏 (distillation for low bit quantization, DLBQ) 方法使用特殊的知识蒸馏框架来提高极低比特量化模型的精度，为克服容量差距的问题，他们通过教师模型和学生模型共享全连接层参数来降低知识迁移的难度，此外，通过分析教师模型输出中重要的数值提出了选择性知识蒸馏，以进行更有针对性的知识蒸馏。DLBQ 虽然实现了极低比特量化模型的知识蒸馏，但是由于共享全连接层的原因必须确保教师模型和学生模型在全连接层之前共享相同的输出通道，因此该方法仅适用于网络深度不同，但最终卷积通道数相同的情况。QD 方法[79]和 DLBQ 方法[77]都实现了边量化边蒸馏的模型压缩加速，即在模型压缩的过程中量化和蒸馏同时进行，但量化模型位宽仍然取决于量化方法的选择，因此不一定是平衡压缩率和精度的最优解。

综上所述，知识蒸馏作为一种训练高性能的轻量化模型的方法通常用于弥补由于量化造成的精度损失，同时量化是构造知识蒸馏所需的轻量化学生网络的一种方式[72,74-75,79-80]。针对知识蒸馏中模型容量差距问题可以使用量化后模型作为教师模型[76]或者设计合适的训练方案[73,75-76,78]以保证知识更好的传递。特别的，在考虑数据安全与隐私性的领域，可以使用对抗学习生成样本来进行知识蒸馏[78]。

总结 3.2 节和 3.3 节所概述的“知识蒸馏+剪枝”和“知识蒸馏+量化”的联合方法，两者具有许多共同的基本思想。首先，知识蒸馏作为一种轻量化网络的高性能训练方法，能够有效弥补由于剪枝或量化造成的精度损失，同时剪枝或量化可以为知识蒸馏提供足够轻量化的学生模型，因此

大部分研究[30,50,65]使用剪枝或量化压缩后的模型作为学生模型, 使用压缩前的模型作为教师模型进行知识蒸馏以获取高性能轻量化网络。但是由于知识蒸馏中的师生模型容量差距过大会影响蒸馏效果, 有些研究[57-58,76]使用剪枝或量化后的模型作为教师模型以降低容量差距、提升蒸馏效果。由于知识蒸馏和剪枝、量化两阶段结合训练周期长、成本高, 且简单的组合 2 种压缩方法往往达不到最佳的压缩加速效果, 不少研究[60-63,78-79]提出了边剪枝或量化边蒸馏的端到端的压缩加速方法, 其中的剪枝或量化往往都是迭代进行, 并引入蒸馏损失指导模型的训练过程。

2.3　剪枝+量化

剪枝与量化联合进行模型压缩加速主要应用于卷积神经网络[64,72,80,82-86], 近年来也出现了应用于语音增强[87]、自然语言处理[88]和强化学习[64]的情景。剪枝与量化的结合方式主要分为先剪枝后量化[80,83,84,87-89]和边剪枝边量化[64,85-86]。剪枝操作根据其剪枝粒度的不同可分为结构化剪枝和非结构化剪枝。量化操作根据其量化精度的不同可分为固定精度量化, 即所有权重量化精度一致; 动态精度量化, 即网络不同部分各自选择适合的精度; 以及离散值量化, 即保留少部分精度不变的离散值作为量化后的权重值, 其余权重均量化为这些离散值。

本文调研的剪枝与量化相结合的文献中所使用的剪枝、量化方法及两者之间的结合方式统计如表 3 所示。

表 3　本文所整理的“剪枝+量化”文献方法汇总

Table 3　Summary of the literature methods of “pruning + quantification” organized in this paper

年份	文献题目	剪枝方法	量化方法	结合方式
2015	Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding	基于权重大小的非结构化剪枝[21]	*聚类量化	A-1-3
2018	CLIP-Q: Deep Network Compression Learning by In-Parallel Pruning-Quantization	*并行剪枝量化操作: 基于权重的非结构化剪枝	*并行剪枝量化操作: 平均值量化	B-1-3
2019	Increasing Compactness of Deep Learning Based Speech Enhancement Models with Parameter Pruning and Quantization Techniques	*参数剪枝	*参数量化	A-2-3
2019	A High Energy-Efficiency FPGA-Based LSTM Accelerator Architecture Design by Structured Pruning and Normalized Linear Quantization	基于带偏置的置换块对角掩码矩阵进行剪枝[90]	*归一化的线性量化方法	A-2-1
2020	Quantization and Pruning for Neural Network Compression and Regularization	基于权重大小的非结构化剪枝	逐通道量化[91]	A-1-2
2020	APQ: Joint Search for Network Architecture, Pruning and Quantization Policy	*设计超网络进行细粒度剪枝	*设计量化预测器得到量化精度	B-2-1
2021	Learning Low Resource Consumption CNN Through Pruning and Quantization	*输出输入通道联合的稀疏度正则化剪枝方法	*增量式量化	A-2-1
2021	Quantization-Aware Pruning Criterion for Industrial Applications	*均匀变分网络量化器: 基于稀疏变分随机失活模块剪枝	*均匀变分网络量化器: 动态精度量化	B-1-2
2022	Variational Channel Distribution Pruning and Mixed-Precision Quantization for Neural Network Model Compression	*变分通道分布剪枝	*基于变分通道分布的混合精度量化	A-2-2
2022	YOLO-Based Face Mask Detection on Low-End Devices Using Pruning and Quantization	基于L1范数的结构化剪枝[14]	量化为8 bit固定精度; 动态范围量化[92]	A-2-1

注:1) 标注*表示文章新提出的方法; 2) 结合方式编号: A-1-1: 先剪枝后量化, 剪枝操作为非结构化剪枝, 量化操作为动态精度量化; A-1-2: 先剪枝后量化, 剪枝操作为非结构化剪枝, 量化操作为离散值量化; A-2-1: 先剪枝后量化, 剪枝操作为结构化剪枝, 量化操作为固定精度量化; A-2-2: 先剪枝后量化, 剪枝操作为结构化剪枝, 量化操作为动态精度量化; A-2-3: 先剪枝后量化, 剪枝操作为结构化剪枝, 量化操作为离散值量化; B-1-2: 边剪枝边量化, 剪枝操作为非结构化剪枝, 量化操作为动态精度量化; B-1-3: 边剪枝边量化, 剪枝操作为非结构化剪枝, 量化操作为离散值量化; B-2-1: 边剪枝边量化, 剪枝操作为结构化剪枝, 量化操作为固定精度量化。

2.3.1　先剪枝后量化

先剪枝后量化结合方式按照剪枝方法分为结构化剪枝和非结构化剪枝 2 类，又按照量化方法细分为 5 种方式，如图 11 所示。

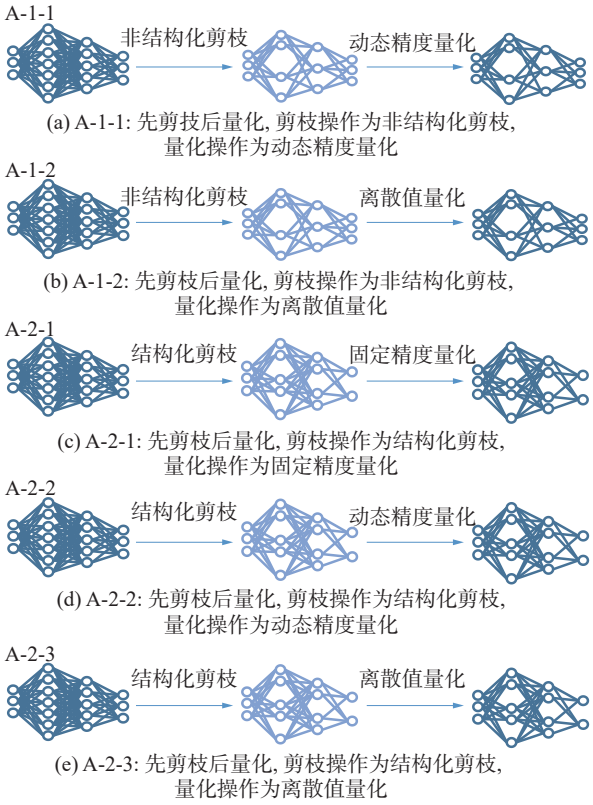


图 11　先剪枝后量化方法分类示意

Fig. 11　Schematic diagram of classification by pruning before quantization

1) 不同粒度的剪枝方法。

剪枝操作根据剪枝粒度的不同可分为非结构化剪枝和结构化剪枝两大类。早期的方法中更多使用非结构化剪枝，其以权重为剪枝单位，通过权重大小来衡量权重的重要程度，对于值较小的权重则将其剪枝。该方法的优势在于剪枝后能更好地维持原网络的高性能，而劣势在于计算量相对较大。剪枝程度通常使用设置的超参数阈值来控制，在超参数的选择上也有很多方案，Han 等[89]设置权重阈值，对低于该阈值的权重进行剪枝；而 Paupamah 等[82]通过实验调整设置合适的剪枝率，对网络不断剪枝直到当前剪枝率已达到预先设定值。相比而言，设置剪枝率阈值能够更有效地控制网络的压缩程度。此外，值得注意的是，Paupamah 等[82]的研究还表明剪枝操作可用于轻量化网络结构的进一步压缩和复杂网络过拟合的抑制。

目前使用更多的剪枝方法为结构化剪枝，其以卷积核通道或整个卷积核等整体结构为剪枝单位。其优势主要在于计算量及所需要的权重存储空间相对较小。通常的结构化剪枝方法使用一定的指标来衡量整体结构的重要性，对于不重要的结构进行剪枝，如 Liberatori 等[83]使用卷积核权重的 L1 范数作为衡量指标，对于一定量 L1 范数值较小的卷积核进行剪枝，剪枝的卷积核数量由预设的超参数剪枝率控制。

目前除上述提到的常用结构化剪枝方法外，也有一系列各具特色的新结构化剪枝方法被提出，以用于满足更高的需求。Chang 等[84]提出了一种基于批归一化层不同通道规模系数的分布来进行剪枝的方法变分通道分布剪枝 (variational channel distribution prune, VCD prune)。该方法考察批归一化层不同通道规模系数的分布，将该分布拟合为正态分布，并考虑该正态分布的方差。方差大意味着该通道重要性相对较低，应进行剪枝。对于剪枝程度，使用超参数方差阈值进行控制。Qi 等[87]提出了一种输出输入通道联合的稀疏度正则化剪枝方法。该方法考虑了 2 个连续层中前一层输出通道和下一层对应输入通道之间的关联性，构造了输入输出通道联合正则项，通过正则项约束来控制权重值从而进行剪枝。在具体剪枝过程中，通道权重的平方和作为相应的重要性衡量指标。每一迭代轮次均设置特定的剪枝率，在达到当前轮次剪枝率后，对网络进行重训练以提升其预测准确率。此外，对于如 ResNet 等包含多分支连接的网络，仅考虑主通路层的关联性，而暂时忽略其他分支连接。Wu 等[88]提出基于稀疏度参数的参数剪枝方法，使用某通道中小于均值的权重数占通道总权重数的比例来表示稀疏度，稀疏度趋近于 1，证明该通道重要性相对越低，故应被剪枝。在进行剪枝时，使用稀疏度阈值作为超参数控制剪枝程度。

可以看到，上述结构化剪枝方法虽各具特色，但其共同点在于提出了不同于传统结构化剪枝方法的重要性衡量指标。这些指标相对于传统结构化剪枝方法更为精细，有助于降低剪枝操作对预测精度带来的损失。传统非结构化剪枝注重精度，而结构化剪枝注重计算量与存储空间的降低，这些新方法则是非结构化剪枝与结构化剪枝的折中。

此外，还有一些新剪枝方法能够在特定情况下进行高效剪枝。Zheng 等[85]提出基于带偏置的置换块对角掩码矩阵的剪枝方法。块对角掩码矩阵会以掩码形式保留当前块对角线位置的权重，而将非对角线位置的权重置 0，而偏置代表块对角掩码矩阵当前对角线的偏移量。该方法较一般的结构化剪枝方法精度更高，同时偏移的方式便于基于 FPGA 的压缩加速设备的硬件实现，在获得精度的同时也实现了高效。

2) 不同精度的量化方法。

量化方法根据量化精度的不同可分为固定精度量化、动态精度量化和离散值量化 3 大类。目

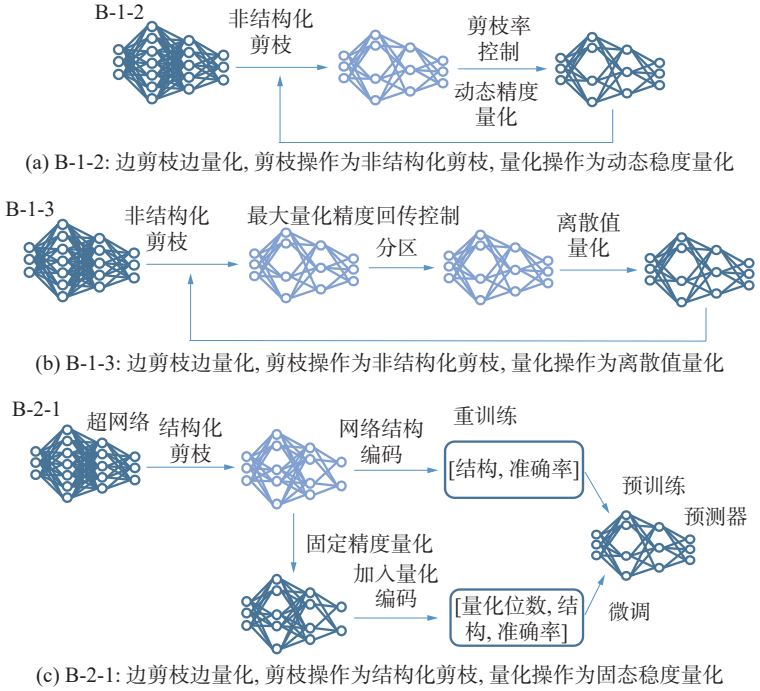
前最常用的量化方法即为固定精度量化, 主要优势在于便于实现, 且通常量化为 8 bit 整型, 便于权重的存储, 但也可根据实际情况调节, 如文献[83] 中同时使用了 8 bit 固定精度量化和动态精度量化。一般在使用固定精度量化方法时, 不仅会进行量化操作, 还包含其他修正步骤, 以避免直接量化带来模型性能的大幅下降。Zheng 等[85] 在使用固定精度量化方法时, 首先对权重进行归一化, 再对其进行对称均匀量化。量化后, 使用饱和操作对权重进行进一步修正。Qi 等[87] 基于固定精度量化提出了增量式量化方法, 即量化与重训练交替。在选取待量化权重时, 可通过量化误差最小化方法或网络预测准确率波动最小化方法。在重训练过程中, 需要先将浮点数权重值转化为固定精度再进行重训练。

动态精度量化与固定精度量化类似, 其区别在于动态精度量化会使用特定方法对网络不同结构部分计算对应的最适合的量化精度。Paupamah 等[82] 使用的动态精度量化方法以通道为量化精度选择单位, 每个通道统一使用最适合当前通道的量化位数。Chang 等[84] 提出基于批归一化层不同通道规模系数分布的混合精度量化方法。该方法与本文所使用的剪枝方法均基于所拟合的批归一化层规模系数分布来进行, 即方差越大, 通道能容忍的量化误差也越大, 因而可以降低量化位数。可以看到, 动态精度量化的过程更为复杂, 计算量比固定精度量化大, 但降低了量化操作对网络性能的影响, 同时减少了次要权重的存储精度, 也节约了存储空间。

离散值量化与上述 2 种量化方法差别较大, 该方法以原精度存储少量权重作为离散值, 而其余权重均量化为上述离散值, 仅存储其对应的索引, 从而实现模型压缩。离散值量化相较固定精度量化精度更高, 而相较动态精度量化计算量和权重所需存储空间都很少。离散值的选择方法也有多种, 最为常见的情况是, 在基于聚类的离散值量化方法中, 选择聚类中心作为离散值。Han 等[89] 提出的聚类量化方法和 Wu 等[88] 提出的参数量化方法就是典型的基于聚类的量化方法, 上述方法也均使用聚类中心作为离散值。在使用基于聚类的量化方法时, 聚类的质量最终会影响量化的质量, 因此上述研究也涉及如何对其中的聚类过程进行优化。首先, 对于聚类类别数, Wu 等[88] 通过实验方法选取类别数以实现模型精度和计算代价的较好平衡。其次, 对于聚类中心初始化, Han 等[89] 提出了随机初始化、基于密度的初始化及线性初始化 3 种方法, 并通过实验证明了线性初始化方法的最优性。值得注意的是, 由于离散值量化的量化策略不同于前 2 种量化方法, 其前向传播和反向传播过程与前 2 种方法相比也有一定差异。Han 等[89] 指出, 前向传播过程会记录每个权重的值和索引, 在反向传播时相同索引的权重的梯度值进行相加, 合成为一个梯度值, 再用该值对离散化的权重值进行更新, 最终训练得到一组离散化权重。

2.3.2　边剪枝边量化

边剪枝边量化方法按照剪枝和量化方法的结合方式细分为 3 种方法, 如图 12 所示。



(a) B-1-2: 边剪枝边量化, 剪枝操作为非结构化剪枝, 量化操作为动态稳度量化

(b) B-1-3: 边剪枝边量化, 剪枝操作为非结构化剪枝, 量化操作为离散值量化

(c) B-2-1: 边剪枝边量化, 剪枝操作为结构化剪枝, 量化操作为固态稳度量化

图 12　边剪枝边量化方法分类示意

Fig. 12　Schematic diagram of side pruning and side quantization method classification

先剪枝后量化的结合方式下，剪枝操作与量化操作依然是两阶段操作，一个阶段进行优化时无法兼顾另一个阶段，很难保证 2 个阶段同时达到最优。因此，一系列边剪枝边量化的方法被提出，这些方法中剪枝和量化操作相互关联、共同优化，最终得到最优的剪枝量化联合策略。

Tung 等[86] 提出并行剪枝量化策略，该策略将剪枝过程和量化过程结合，每次迭代会对剪枝和量化同时进行优化。每次迭代过程中，并行剪枝量化操作包含 3 个步骤。首先，进行非结构化剪枝操作。其次，进行分区操作，对未剪枝区域进行分区。最后，进行离散值量化操作，同一区间的权重均量化为一个离散值，该离散值由该区间内所有权重求均值得到。该方法在每一迭代轮次内部依然是先剪枝后量化的过程，但每一迭代轮次的并行剪枝量化操作完成后均进行重训练以实现精度恢复，在每一次重训练的过程中均实现了对于剪枝和量化过程的同步优化，因而优于一般的先剪枝后量化策略。当然，该方法相比一般的先剪枝后量化的过程付出的计算代价也是更大的。

针对传统神经网络设计过程中网络结构搜索，及网络压缩过程中剪枝、量化各阶段分离的问题，Wang 等[64] 提出了将网络结构搜索、剪枝、量化 3 阶段合并的轻量化网络设计策略。该方法基于超网络架构设计和精度预测器设计实现。超网络架构设计即神经网络中如卷积核大小等超参数可以变动，而剪枝过程随着超网络训练使得内部结构改变从而实现。这里使用结构化剪枝，对卷积核通道进行剪枝。对于精度预测器设计，其输入为网络结构策略编码，输出为预测精度估计。在精度预测器的微调阶段，对于原网络加入量化过程，使精度预测器由网络策略编码、量化策略编码得到预测准确率。使用该预测器，可以同时得到最优的网络结构策略和量化策略，即最优的剪枝操作和量化操作。该方法通过学习网络架构设计的方式可协同进行剪枝和量化策略的调优，其中蕴含着元学习的思想。

Gil 等[93] 提出了一种高效协同完成剪枝与量化操作的深度神经网络压缩方法，即均匀变分网络量化器（uniform variational network quantizer, UVNQ）。其中的剪枝方法为基于网络正则化随机失活技术（dropout）的非结构化剪枝，即为每个神经元设置对应的随机失活几率以进行剪枝。其中的量化方法为动态精度量化，即对每个权重使用不同的量化精度，其量化精度由上述训练得到的剪枝几率来控制。本文提出的方法作为协同完成剪枝与量化操作的方法，其协同主要体现在剪枝与量化操作之间的关联性。在训练开始前，需要给出一个代表最大量化精度的超参数，该超参数为剪枝几率设置了边界值，以保证在训练过程中网络表现不会大幅下降；在训练结束剪枝完成后，量化操作的量化精度也要通过剪枝几率计算得到。

从上述文献可以看出，边剪枝边量化策略通常比先剪枝后量化策略计算量大，这正是两阶段协同调优以降低网络压缩后性能损失的代价。在实现剪枝量化协同的方式上，Tung 等[86] 和 Gil 等[93] 的研究更多是先剪枝后量化方法的延申，将先剪枝后量化的过程多次迭代，或是使先进行的剪枝操作和后进行的量化操作产生关联；而 Wang 等[64] 的研究相当于是各种剪枝策略与各种量化策略的组合，相较而言考虑了更多协同调优的可能性，也更易取得较好的效果，但同样地计算量也会更大。

综上所述，剪枝操作通过调整模型结构进行模型压缩，而量化操作通过调整参数存储量进行模型压缩。对于剪枝与量化的结合，可以按照次序进行操作[80,83-84,87-89]，也可以协同进行[64,85-86]，两阶段操作互相影响，共同优化。大部分研究都根据需求对剪枝或量化方法进行了改进，但剪枝和量化的基本结合方式还是可以分为先剪枝后量化、边剪枝边量化两大类。

3　结束语

神经网络在端侧部署离不开模型压缩加速技术，日益高涨的产业需求正推动其蓬勃发展。但目前的方法在产品化实际部署水平上还有很大的发展空间，下面是几个值得关注和探讨的重点研究方向。

3.1　多方法深度联合

现有的联合模型压缩方法往往以一种方法为主、另一种方法为辅，使用多阶段的训练完成模型的轻量化压缩，如先剪枝后蒸馏、先量化后蒸馏的方法，虽然这种结合方式可以直接使用已经提出的成熟的模型压缩方法，但是往往具有训练过程复杂、周期长的缺点。现有的端到端的训练方法如边剪枝边蒸馏、边量化边蒸馏往往需要设计新的模型压缩方法，因此探究将多种成熟的模型压缩方法联合同时进行压缩的端到端的训练框架是未来研究的方向。此外，对于同时结合剪枝、量化和知识蒸馏进行模型压缩的研究尚未成熟，有待进一步研究。

3.2　最优方案高效选择

如何选择最优的模型压缩方案和合适的超参数, 在保证精度不严重损失的前提下最大程度的实现网络模型的压缩和加速, 从而尽可能缩短开发周期、增加产品的市场竞争力, 正成为业界的核心需求。因此探究如何更快速高效地实现压缩加速, 而非重复尝试不同的压缩方案和超参数组合, 正成为产业应用落地急需解决的问题。

3.3　自适应模型压缩

现有的模型压缩方法中的各种超参数都需要手工调节, 如量化后模型的位宽、剪枝模型的剪枝率、知识蒸馏的温度因子等, 这些超参数的选择需要经验, 且不一定是平衡模型压缩率和精度的最优解。因此研究不依赖于人工设定超参数的自动模型压缩方法很值得探究。

3.4　多场景模型压缩

一方面, 现有的模型压缩方法往往需要有标签的数据进行微调以恢复模型精度, 这将延长模型训练周期。在无监督学习和数据隐私安全性高的场景下往往无法获得具有标签的数据, 因此无数据或少数据的模型压缩方法也有待研究。另一方面, 目前压缩加速方法多应用于计算机视觉领域的卷积神经网络, 而还有大量其他结构的模型应用于不同的场景, 如应用于自然语言处理的长短期记忆模型 (long short-term memory , LSTM)、用于知识图谱领域的图神经网络 (group neural network , GNN) 及对抗学习领域的生成对抗网络 (generative adversarial network , GAN) 等, 针对这些模型的压缩加速方法需要对现有的方法进行改进, 有待进一步研究。

3.5　软硬件协同设计

目前大多数模型压缩加速算法仅从软件层面对模型进行优化, 并未考虑硬件平台是否支持其特殊的运算, 这导致了算法仅仅压缩模型而不能加速模型。如大多数硬件中算数逻辑单元 (arithmetic and logic unit , ALU) 的数据位宽固定, 对于不同量化位宽的模型计算并不能很好地支持, 往往需要符号位拓展才能适配, 业界常用 INT8 量化作为取舍; 此外, 大多数硬件对非结构化剪枝后的稀疏化矩阵的计算并不支持, 而非结构化剪枝能够细粒度地剔除冗余参数, 在模型压缩的同时保持模型精度, 业界常用结构化通道剪枝作为取舍。基于上述现象, 软硬件协同设计在一开始就考虑软硬件的功能划分和适配问题, 能够减少产品的迭代次数、加快上市时间, 是未来业界的研究热点。

参考文献:

[1] 黄震华, 杨顺志, 林威, 等. 知识蒸馏研究综述 [J]. 计算机学报, 2022, 45(3): 624–653.
HUANG Zhenhua, YANG Shunzhi, LIN Wei, et al. Knowledge distillation: a survey[J]. Chinese journal of computers, 2022, 45(3): 624–653.

[2] 高晗, 田育龙, 许封元, 等. 深度学习模型压缩与加速综述 [J]. 软件学报, 2021, 32(1): 68–92.
GAO Han, TIAN Yulong, XU Fengyuan, et al. Survey of deep learning model compression and acceleration[J]. Journal of software, 2021, 32(1): 68–92.

[3] 邵仁荣, 刘宇昂, 张伟, 等. 深度学习中知识蒸馏研究综述 [J]. 计算机学报, 2022, 45(8): 1638–1673.
SHAO Renrong, LIU Yuang, ZHANG Wei, et al. A survey of knowledge distillation in deep learning[J]. Chinese journal of computers, 2022, 45(8): 1638–1673.

[4] HOWARD A G, ZHU Menglong, CHEN Bo, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017−04−17)[2023−06−21]. https://arxiv.org/abs/1704.04861.pdf.

[5] SANDLER M, HOWARD A, ZHU Menglong, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 4510−4520.

[6] ZHANG Xiangyu, ZHOU Xinyu, LIN Mengxiao, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 6848−6856.

[7] MA Ningning, ZHANG Xiangyu, ZHENG Haitao, et al. ShuffleNet V2: practical guidelines for efficient CNN architecture design[C]//European Conference on Computer Vision. Cham: Springer, 2018: 122−138.

[8] IANDOLA F N, HAN Song, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size[EB/OL]. (2016−11−04)[2023−06−21]. https://arxiv.org/abs/1602.07360.pdf.

[9] HOWARD A, SANDLER M, CHEN Bo, et al. Searching for MobileNetV3[C]//2019 IEEE/CVF International Conference on Computer Vision . Piscataway: IEEE, 2020: 1314−1324.

[10] ZOPH B, VASUDEVAN V, SHLENS J, et al. Learning transferable architectures for scalable image recognition[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 8697−8710.

[11] TAN Mingxing, CHEN Bo, PANG Ruoming, et al. MnasNet: platform-aware neural architecture search for mo-

bile[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 2815−2823.

[12] LECUN Y, DENKER J, SOLLA S. Optimal brain damage[J]. Advances in neural information processing systems, 1989, 2: 598–605.

[13] HASSIBI B, STORK D G. Second order derivatives for network pruning: optimal brain surgeon[C]//Advances in Neural Information Processing Systems 5. New York: ACM, 1992: 164−171.

[14] LI Hao, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB/OL]. (2017−03−10)[2023−06−21]. https://arxiv.org/abs/1608.08710.pdf.

[15] HE Yang, KANG Guoliang, DONG Xuanyi, et al. Soft filter pruning for accelerating deep convolutional neural networks[EB/OL]. (2018−08−21)[2023−06−21]. https://arxiv.org/abs/1808.06866.pdf.

[16] LIU Zhuang, LI Jianguo, SHEN Zhiqiang, et al. Learning efficient convolutional networks through network slimming[C]//2017 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 2755−2763.

[17] LUO Jianhao, WU Jianxin, LIN Weiyao. ThiNet: a filter level pruning method for deep neural network compression[C]//2017 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 5068−5076.

[18] WEN Wei, WU Chunpeng, WANG Yandan, et al. Learning structured sparsity in deep neural networks[C]// In Proceedings of the 30th International Conference on Neural Information Processing Systems. Curran Associates Inc. Red Hook, NY, USA, 2016: 2082–2090.

[19] LEBEDEV V, LEMPITSKY V. Fast ConvNets using group-wise brain damage[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway : IEEE, 2016: 2554−2564.

[20] YE Xucheng, DAI Pengcheng, LUO Junyu, et al. Accelerating CNN Training by Pruning Activation Gradients[C]// In Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part XXV. Springer-Verlag, Berlin: Heidelberg, 2020: 322–338.

[21] HAN Song, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[EB/OL]. (2015−10−31)[2023−06−21]. https://arxiv.org/abs/1506.02626.pdf.

[22] DETTMERS T. 8-bit approximations for parallelism in deep learning[EB/OL]. (2016−02−19)[2023−06−21]. https://arxiv.org/abs/1511.04561.pdf.

[23] JACOB B, KLIGYS S, CHEN Bo, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 2704−2713.

[24] LIN Xiaofan, ZHAO Cong, PAN Wei. Towards accurate binary convolutional neural network[EB/OL]. (2016−02−19)[2023−06−21]. https://arxiv.org/abs/1711.11294.pdf.

[25] COURBARIAUX M, BENGIO Y, DAVID J P. BinaryConnect: training deep neural networks with binary weights during propagations[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. New York: ACM, 2015: 3123−3131.

[26] HOU Lu, Yao Quanming, KWOK J T. Loss-aware binarization of deep networks[J]. (2018−05−10)[2023−06−21]. https://arxiv.org/abs/1611.01600v1.pdf.

[27] JUEFEI-XU F, BODDETI V N, SAVVIDES M. Local binary convolutional neural networks[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 4284−4293.

[28] ZHU Chenzhuo, HAN Song, Mao Huizi, et al. Trained ternary quantization[EB/OL]//(2017−02−23)[2023−06−21]. https://arxiv.org/abs/1612.01064v2.pdf.

[29] ACHTERHOLD J, KOEHLER J M, SCHMEINK A, et al. Variational network quantization[C]//International Conference on Learning Representations. Ithaca:ICLR, 2018: 1−18.

[30] MELLEMPUDI N, KUNDU A, MUDIGERE D, et al. Ternary neural networks with fine-grained quantization[EB/OL]. (2017−05−30)[2023−06−21]. https://arxiv.org/abs/1705.01462.pdf.

[31] BOROUMAND A, GHOSE S, KIM Y, et al. Google workloads for consumer devices: mitigating data movement bottlenecks[C]//Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2018: 316−331.

[32] MISHRA A, COOK J, NURVITADHI E, et al. WRPN: training and inference using wide reduced-precision networks[EB/OL]. (2017−04−10)[2023−06−21]. https://arxiv.org/abs/1704.03079.pdf.

[33] WANG Kuan, LIU Zhijian, LIN Yujun, et al. HAQ: hardware-aware automated quantization with mixed precision[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 8604−8612.

[34] ZHANG Dongqing, YANG Jiaolong, YE D, et al. LQ-nets: learned quantization for highly accurate and compact deep neural networks[C]//European Conference on Computer Vision. Cham: Springer, 2018: 373−390.

[35] GONG Yunchao, LIU Liu, YANG Ming, et al. Com-

pressing deep convolutional networks using vector quantization[EB/OL]. (2014−12−18)[2023−06−21]. https://arxiv.org/abs/1412.6115.pdf.

[36] TAILOR S A, FERNANDEZ-MARQUES J, LANE N D. Degree-quant: quantization-aware training for graph neural networks[EB/OL]. (2021−03−15)[2023−06−21]. https://arxiv.org/abs/2008.05000.pdf.

[37] SEIDE F, FU Hao, DROPPO J, et al. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs[C]//Interspeech 2014. ISCA: ISCA, 2014: 1−5.

[38] LI Conglong, AHMAD AWAN A, TANG Hanlin, et al. 1-bit LAMB: communication efficient large-scale large-batch training with LAMB's convergence speed[C]//2022 IEEE 29th International Conference on High Performance Computing, Data, and Analytics (HiPC). Piscataway: IEEE, 2023: 272−281.

[39] ALISTARH D, GRUBIC D, LI J Z, et al. QSGD: communication-efficient SGD via gradient quantization and encoding[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 1707−1718.

[40] GOU Jianping, YU Baosheng, MAYBANK S J, et al. Knowledge distillation: a survey[J]. International journal of computer vision, 2021, 129(6): 1789–1819.

[41] BUCILUǍ C, CARUANA R, NICULESCU-MIZIL A. Model compression[C]//Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM, 2006: 535−541.

[42] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[EB/OL]. (2015−03−09)[2023−06−21]. https://arxiv.org/abs/1503.02531.pdf.

[43] MIRZADEH S I, FARAJTABAR M, LI Ang, et al. Improved knowledge distillation via teacher assistant[J]. Proceedings of the AAAI conference on artificial intelligence, 2020, 34(4): 5191–5198.

[44] REMERO A, BALLAS N, EBRAHIMI K, et al. FitNets: hints for thin deep Nets[EB/OL].(2015−03−27)[2023−06−21]. https://arxiv.org/abs/1412.6550.pdf.

[45] ZAGORUYKO S, KOMODAKIS N.Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer[EB/OL]. (2017−02−12)[2023−06−21]. https://arxiv.org/abs/1612.03928.pdf.

[46] YIM J, JOO D, BAE J, et al. A gift from knowledge distillation: fast optimization, network minimization and transfer learning[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017: 7130−7138.

[47] XU Xixia, ZOU Qi, LIN Xue, et al. Integral knowledge distillation for multi-person pose estimation[J]. IEEE signal processing letters, 2020, 27: 436–440.

[48] ZHANG Linfeng, SONG Jiebo, GAO Anni, et al. Be your own teacher: improve the performance of convolutional neural networks via self distillation[C]//2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2020: 3712−3721.

[49] ZHANG Feng, HU Hong, DAI Hanbin, et al. Self-evolutionary pose distillation[C]//2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing. Piscataway: IEEE, 2020: 240−244.

[50] WU Min, MA Weihua, LI Yue, et al. Automatic optimization of super parameters based on model pruning and knowledge distillation[C]//2020 International Conference on Computer Engineering and Intelligent Control. Piscataway: IEEE, 2021: 111−116.

[51] WU Min, MA Weihua, LI Yue, et al. The optimization method of knowledge distillation based on model pruning[C]//2020 Chinese Automation Congress. Piscataway: IEEE, 2021: 1386−1390.

[52] AFLALO Y, NOY A, LIN Ming, et al. Knapsack pruning with inner distillation[EB/OL]. (2020−06−03)[2023−06−21]. https://arxiv.org/abs/2002.08258.pdf.

[53] ZHOU Zhaojing, ZHOU Yun, JIANG Zhuqing, et al. An efficient method for model pruning using knowledge distillation with few samples[C]//ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2022: 2515−2519.

[54] WANG Bo, JIANG Qingji, SONG Dawei, et al. SAR vehicle recognition via scale-coupled Incep_Dense Network (IDNet)[J]. International journal of remote sensing, 2021, 42(23): 9109–9134.

[55] CHEN Shiqi, ZHAN Ronghui, WANG Wei, et al. Learning slimming SAR ship object detector through network pruning and knowledge distillation[J]. IEEE journal of selected topics in applied earth observations and remote sensing, 2020, 14: 1267–1282.

[56] WANG Zhen, DU Lan, LI Yi. Boosting lightweight CNNs through network pruning and knowledge distillation for SAR target recognition[J]. IEEE journal of selected topics in applied earth observations and remote sensing, 2021, 14: 8386–8397.

[57] AGHLI N, RIBEIRO E. Combining weight pruning and knowledge distillation for CNN compression[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Piscataway: IEEE, 2021: 3185−3192.

[58] PARK J, NO A. Prune your model before distill it[M]. Cham: Springer Nature Switzerland, 2022: 120−136.

[59] CHE Hongle, SHI Qirui, CHEN Juan, et al. HKDP: a hybrid approach on knowledge distillation and pruning for neural network compression[C]//2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing. Piscataway: IEEE, 2022: 188−193.

[60] CHEN Liyang, CHEN Yongquan, XI Juntong, et al. Knowledge from the original network: restore a better pruned network with knowledge distillation[J]. Complex & intelligent systems, 2022, 8(2): 709–718.

[61] YAO Weiwei, ZHANG Jie, LI Chen, et al. Semantic segmentation optimization algorithm based on knowledge distillation and model pruning[C]//2019 2nd International Conference on Artificial Intelligence and Big Data. Piscataway: IEEE, 2019: 261−265.

[62] CUI Baiyun, LI Yingming, ZHANG Zhongfei. Joint structured pruning and dense knowledge distillation for efficient transformer model compression[J]. Neurocomputing, 2021, 458: 56–69.

[63] XU Bangguo, ZHANG Tiankui, WANG Yapeng, et al. A knowledge- distillation - integrated pruning method for vision transformer[C]//2022 21st International Symposium on Communications and Information Technologies. Piscataway: IEEE, 2022: 210−215.

[64] WANG Tianzhe, WANG Kuan, CAI Han, et al. APQ: joint search for network architecture, pruning and quantization policy[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 2075−2084.

[65] XU Rui, LUAN Siyu, GU Zonghua, et al. LRP-based policy pruning and distillation of reinforcement learning agents for embedded systems[C]//2022 IEEE 25th International Symposium on Real-Time Distributed Computing. Piscataway: IEEE, 2022: 1−8.

[66] LIU Yu, JIA Xuhui, TAN Mingxing, et al. Search to distill: pearls are everywhere but not the eyes[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 7536−7545.

[67] SATTLER F, MARBAN A, RISCHKE R, et al. CFD: communication-efficient federated distillation via soft-label quantization and delta coding[J]. IEEE transactions on network science and engineering, 2021, 9(4): 2025–2038.

[68] RUSU A A, COLMENAREJO S G, GULCEHRE C, et al. Policy distillation[EB/OL]. (2016−01−07)[2023−06−21]. https://arxiv.org/abs/1511.06295.pdf.

[69] CHO J H, HARIHARAN B. On the efficacy of knowledge distillation[C]//2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2020: 4793−4801.

[70] LI Chenxing, ZHU Lei, XU Shuang, et al. Compression of acoustic model via knowledge distillation and pruning[C]//2018 24th International Conference on Pattern Recognition. Piscataway: IEEE, 2018: 2785−2790.

[71] PRAKOSA S W, LEU J S, CHEN Zhaohong. Improving the accuracy of pruned network using knowledge distillation[J]. Pattern analysis & applications, 2021, 24(2): 819–830.

[72] SHIN S, BOO Y, SUNG W. Knowledge distillation for optimization of quantized deep neural networks[C]//2020 IEEE Workshop on Signal Processing Systems (SiPS). Piscataway: IEEE, 2020: 1−6.

[73] MISHRA A, MARR D. Apprentice: using knowledge distillation techniques to improve low-precision network accuracy[EB/OL]. (2017−11−15)[2023−06−21]. https://arxiv.org/abs/1711.05852.pdf.

[74] MIN Rui, LAN Hai, CAO Zongjie, et al. A gradually distilled CNN for SAR target recognition[J]. IEEE access, 2019, 7: 42190–42200.

[75] KIM J, BHALGAT Y, LEE J, et al. QKD: quantization-aware knowledge distillation[EB/OL]. (2019−11−28)[2023−06−21]. https://arxiv.org/abs/1911.12491.pdf.

[76] OKUNO T, NAKATA Y, ISHII Y, et al. Lossless AI: toward guaranteeing consistency between inferences before and after quantization via knowledge distillation[C]//2021 17th International Conference on Machine Vision and Applications (MVA). Piscataway: IEEE, 2021: 1−5.

[77] SI Liang, LI Yuhai, ZHOU Hengyi, et al. Explore a novel knowledge distillation framework for network learning and low-bit quantization[C]//2021 China Automation Congress. Piscataway: IEEE, 2022: 3002−3007.

[78] KIM M, LEE S, HONG S J, et al. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders[C]//Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022: 6713−6725.

[79] POLINO A, PASCANU R, ALISTARH D. Model compression via distillation and quantization[EB/OL]. (2018−02−15)[2023−06−21]. https://arxiv.org/abs/1802.05668.pdf.

[80] CHOI Y, CHOI J, EL-KHAMY M, et al. Data-free network quantization with adversarial knowledge distillation[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE, 2020: 3047−3057.

[81] WEI Yi, PAN Xinyu, QIN Hongwei, et al. Quantization

mimic: towards very tiny CNN for object detection[C]// European Conference on Computer Vision. Cham: Springer, 2018: 274−290.

[82] PAUPAMAH K, JAMES S, KLEIN R. Quantisation and pruning for neural network compression and regularisation[C]//2020 International SAUPEC/RobMech/PRASA Conference. Piscataway: IEEE, 2020: 1−6.

[83] LIBERATORI B, MAMI C A, SANTACATTERINA G, et al. YOLO-based face mask detection on low-end devices using pruning and quantization[C]//2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). Piscataway: IEEE, 2022: 900−905.

[84] CHANG Wanting, KUO C H, FANG Lichun. Variational channel distribution pruning and mixed-precision quantization for neural network model compression[C]//2022 International Symposium on VLSI Design, Automation and Test. Piscataway: IEEE, 2022: 1−3.

[85] ZHENG Yong, YANG Haigang, HUANG Zhihong, et al. A high energy-efficiency FPGA-based LSTM accelerator architecture design by structured pruning and normalized linear quantization[C]//2019 International Conference on Field-Programmable Technology. Piscataway: IEEE, 2020: 271−274.

[86] TUNG F, MORI G. CLIP-Q: deep network compression learning by In-parallel pruning-quantization[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 7873−7882.

[87] QI Qi, LU Yan, LI Jiashi, et al. Learning low resource consumption CNN through pruning and quantization[J]. IEEE transactions on emerging topics in computing, 2022, 10(2): 886–903.

[88] WU J Y, YU Cheng, FU S W, et al. Increasing compactness of deep learning based speech enhancement models with parameter pruning and quantization techniques[J]. IEEE signal processing letters, 2019, 26(12): 1887–1891.

[89] HAN Song, MAO Huizi, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding[EB/OL]. (2016−02−15)[2023−06−21]. https://arxiv.org/abs/1510.00149.pdf.

[90] DENG Chunhua, LIAO Siyu, XIE Yi, et al. PermDNN: efficient compressed DNN architecture with permuted diagonal matrices[C]//2018 51st Annual IEEE/ACM International Symposium on Microarchitecture. Piscataway: IEEE, 2018: 189−202.

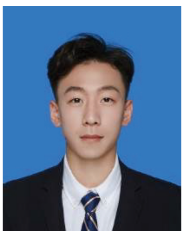
[91] KRISHNAMOORTHI R. Quantizing deep convolutional networks for efficient inference: a whitepaper[EB/OL]. (2018−06−21)[2023−06−21]. https://arxiv.org/abs/1806.08342.pdf.

[92] LIANG Tailin, GLOSSNER J, WANG Lei, et al. Pruning and quantization for deep neural network acceleration: a survey[J]. Neurocomputing, 2021, 461: 370–403.

[93] GIL Y, PARK J H, BAEK J, et al. Quantization-aware pruning criterion for industrial applications[J]. IEEE transactions on industrial electronics, 2022, 69(3): 3203–3213.

作者简介:

宁欣, 青年研究员, IEEE/CCF/CAAI 高级会员, 主要研究方向为计算视觉、神经网络理论与优化计算。主持国家自然科学基金等项目 5 项。发表学术论文 100 余篇。E-mail: ningxin@semi.ac.cn。



赵文尧, 本科生, 主要研究方向为神经网络轻量化算法和硬件加速。E-mail: 2020214817@mail.hfut.edu.cn。

张玉贵, 助理研究员, IEEE 和 CCF 会员, 主要研究方向为计算机视觉、模型优化加速和中医数字化。参与国家重点研发计划项目 2 项、国家自然科学基金项目 4 项、工信部揭榜挂帅项目 1 项。发表学术论文 20 余篇。E-mail: zhangyugui@semi.ac.cn。