软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(1):68−92 [doi: 10.13328/j.cnki.jos.006096]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

深度学习模型压缩与加速综述*

高 晗, 田育龙, 许封元, 仲 盛

(计算机软件新技术国家重点实验室(南京大学),江苏 南京 210023)

通讯作者: 许封元, E-mail: fengyuan.xu@nju.edu.cn

摘 要: 随着训练可用数据量的增长与计算平台处理能力的增强,基于深度学习的智能模型能够完成越来越复杂的任务,其在计算机视觉、自然语言处理等人工智能领域已经取得重大的突破.然而,这些深度模型具有庞大的参数规模,与此相伴的可畏的计算开销与内存需求使其在计算能力受限平台(例如移动嵌入式设备)的部署中遇到了巨大的困难与挑战.因此,如何在不影响深度学习模型性能的情况下进行模型压缩与加速成为研究热点.首先对国内外学者提出的经典深度学习模型压缩与加速方法进行分析,从参数剪枝、参数量化、紧凑网络、知识蒸馏、低秩分解、参数共享和混合方式这 7 个方面分类总结;其次,总结对比几种主流技术的代表性方法在多个公开模型上的压缩与加速效果;最后,对于模型压缩与加速领域的未来研究方向加以展望.

关键词: 深度学习;模型压缩;模型加速;参数剪枝;参数量化;紧凑网络

中图法分类号: TP181

中文引用格式: 高晗,田育龙,许封元,仲盛.深度学习模型压缩与加速综述.软件学报,2021,32(1):68−92. http://www.jos.org.cn/1000-9825/6096.htm

英文引用格式: Gao H, Tian YL, Xu FY, Zhong S. Survey of deep learning model compression and acceleration. Ruan Jian Xue Bao/Journal of Software, 2021,32(1):68−92 (in Chinese). http://www.jos.org.cn/1000-9825/6096.htm

Survey of Deep Learning Model Compression and Acceleration

GAO Han, TIAN Yu-Long, XU Feng-Yuan, ZHONG Sheng

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: With the development of the amount of data available for training and the processing power of new computing platform, the intelligent model based on deep learning can accomplish more and more complex tasks, and it has made major breakthroughs in the field of AI such as computer vision and natural language processing. However, the large number of parameters of these deep models bring awesome computational overhead and memory requirements, which makes the big models must face great difficulties and challenges in the deployment of computing-capable platforms (such as mobile embedded devices). Therefore, model compression and acceleration without affecting the performance have become a research hotspot. This study first analyzes the classical deep learning model compression and acceleration methods proposed by domestic and international scholars, and summarize seven aspects: Parameter pruning, parameter quantization, compact network, knowledge distillation, low-rank decomposition, parameter sharing, and hybrid methods. Secondly, the compression and acceleration performance of several mainstream representative methods is compared on multiple public models. Finally, the future research directions in the field of model compression and acceleration are discussed.

Key words: deep learning; model compression; model acceleration; parameter pruning; parameter quantization; compact network

* 基金项目: 国家自然科学基金(61872180, 61872176); 江苏省“双创计划”; 江苏省“六大人才高峰”高层次人才项目(B 类); 蚂蚁金服科研基金; 中央高校基本科研业务费专项资金(14380069)

Foundation item: National Natural Science Foundation of China (61872180, 61872176); Jiangsu “ShuangChuang” Program; Jiangsu “Six-Talent-Peaks” Program; Ant Financial through the Ant Financial Science Funds for Security Research; Fundamental Research Funds for the Central Universities (14380069)

收稿时间: 2019-10-09; 修改时间: 2020-05-17; 采用时间: 2020-06-04; jos 在线出版时间: 2020-07-27

深度学习模型的压缩和加速是指利用神经网络参数的冗余性和网络结构的冗余性精简模型,在不影响任务完成度的情况下,得到参数量更少、结构更精简的模型.被压缩后的模型计算资源需求和内存需求更小,相比原始模型能够满足更加广泛的应用需求.

本文系统地介绍模型压缩与加速方面的进展.第 1 节主要介绍深度学习模型压缩与加速技术提出的研究背景和研究动机,以及本文的主要贡献.第 2 节主要对目前主流的模型压缩与加速方法进行分类总结,从参数剪枝、参数量化、紧凑网络、知识蒸馏、低秩分解、参数共享、混合方式这 7 个方面探究相关技术的发展历程,并分析其特点.第 3 节主要比较各类压缩与加速技术中一些代表性方法的压缩效果.第 4 节探讨模型压缩与加速领域未来的发展方向.第 5 节对全文进行总结.

1 简 介

1.1 研究背景

神经网络的概念在 20 世纪 40 年代提出后,发展一直不温不火.直到 1989 年,LeCun 教授提出应用于手写字体识别的卷积神经网络[1],取得了良好效果,才使其得到更广泛的发展和关注.卷积神经网络(CNN)的得名即来自于其使用了卷积运算的结果.如图 1 所示,特征图(feature map)是输入数据的中间抽象表示结果,输入特征图(input feature map)是由 C_{in} 个 $H_{in}\times W_{in}$ 的 2D 输入特征图组合而成,每一个滤波器(filter)与输入特征图的通道数相同,由 C_{in} 个 $d\times d$ 的卷积核(kernel)构成,输出特征图(output feature map)的每个通道(channel)都是由输入特征图与每一个 filter 通过卷积运算而得到.但是由于当时数据集规模较小,容易出现过拟合问题,卷积神经网络并没有引起足够的重视.随着大数据时代的到来,数据集的规模不断扩大,计算硬件,特别是 GPU 的飞速发展,神经网络重新获得关注.2009 年,Deng 等人发布当时世界上最大的通用物体识别数据库——ImageNet 数据库[2].从 2010 年开始,每年都会举办基于该数据库的大规模图像识别比赛——ILSVRC[3].2012 年,Hinton 的研究小组采用深度学习模型 AlexNet[4]赢得了该比赛,突破性地将错误率从 26.2%降到 15.3%.此后,深度学习模型开始广泛用于人工智能的各个领域,在许多任务中得到了超越人类的正确率,在自动驾驶、医疗影像分析、智能家居等领域给予人们的生产和生活以更大的帮助.

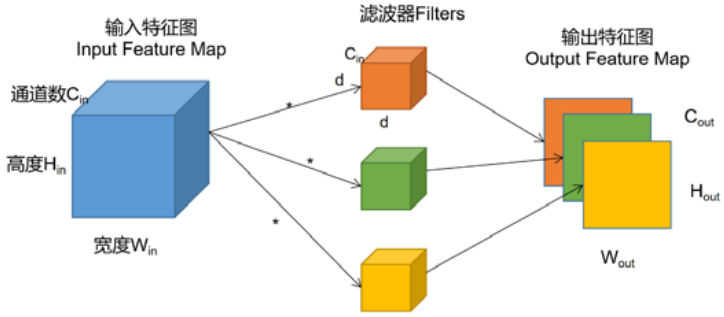

Fig.1 Convolutional operator

图 1 卷积计算

深度学习模型性能提高的同时,计算也越来越复杂,计算开销和内存需求逐渐增加.仅 8 层的 AlexNet[4]需要 0.61 亿个网络参数和 7.29 亿次浮点型计算,花费约 233MB 内存.随后的 VGG-16[5]的网络参数达到 1.38 亿,浮点型计算次数为 1.56 亿,需要约 553MB 内存.为了克服深层网络的梯度消失问题,He 提出了 ResNet[6]网络,首次在 ILSVRC 比赛[3]中实现了低于 5%的 top-5 分类错误,偏浅的 ResNet-50 网络参数就达到 0.25 亿,浮点型计算次数高达 3.9 亿,内存花费约 102MB.庞大的网络参数意味着更大的内存存储,而增长的浮点型计算次数意味着训练成本和计算时间的增长,这极大地限制了在资源受限设备,例如智能手机、智能手环等上的部署.如表 1 所示,深度模型在 Samsung Galaxy S6 的推理时间远超 Titan X 桌面级显卡,实时性较差,无法满足实际应用的需要.

Table 1 Inference time of different deep models[7] (unit: ms)

表 1 不同深度模型的推理时间[7] (单位:毫秒)

模型 \ 设备	Samsung Galaxy S6	Titan X
AlexNet	117	0.54
GoogleNet	273	1.83
VGG-16	1 926	10.67

在深度学习技术日益火爆的背景下,对深度学习模型强烈的应用需求使得人们对内存占用少、计算资源要求低、同时依旧保证相当高的正确率的“小模型”格外关注.利用神经网络的冗余性进行深度学习的模型压缩和加速引起了学术界和工业界的广泛兴趣,各种工作层出不穷.

1.2 研究动机

综述能为读者省去大量阅读时间,以高屋建瓴的视角对该领域技术进行了解.然而截止到目前,在技术不断推陈出新的背景下,关于模型压缩的综述文章数量不多且年代久远,分类简单,难以展示新的趋势和方法.表 2 是本文与目前国内外最新相关综述进行方法分类的种类以及与该分类下的文章数量进行对比的情况,从中可以看出:无论是方法分类还是涉及到的文章数量,已有的综述文章都难以展示新的趋势,对参数剪枝、参数量化和紧凑网络这 3 类方法介绍得都较为粗略,对于混合方式这一新型加速方法未给出详细介绍,不能满足新进入这一领域的初学者了解整体发展方向的需求.

Table 2 Literature classification and quantity of the reviews

表 2 综述的文献分类与数量

	文献[8]	文献[9]	文献[10]	文献[11]	文献[12]	本文
参数剪枝	19	14	2	11	12	**42**
参数量化	7	13	2	10	27	**40**
低秩分解	3	10	6	7	8	**14**
参数共享	–	3	2	6	/	**14**
紧凑网络	–	4	4	5	7	**22**
知识蒸馏	12	7	–	8	3	**18**
混合方式	–	–	–	–	–	**11**

根据图 2 所示的文章发表年份来看,文献[8–11]的最新文章发表于 2017 年,对近年来热门研究方向和新方法的介绍较少.

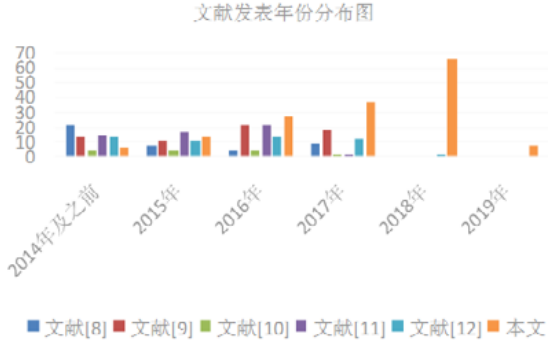

Fig.2 Article publication time and quantity of the reviews

图 2 综述的文章发表时间与数量

根据我们的最新整理,2018 年之后,发表在各大顶级会议上的文章达到 64 篇,占本文统计文章总数的大约 40%,其中,文献[13]首先提出在裁剪权重时加入能耗、延迟等硬件限制作为优化约束,为后续工作[14–16]提供了启发.Network Trimming[17]将激活值为 0 的通道数量作为判断 filter 是否重要的标准,是结构化剪枝领域最有影响

力的工作,开创了设置 filter 评价因子的技术分支.文献[18]提出的依据参数对应损失函数(loss)的梯度来自适应确定每个参数量化位数的方法,打破了固有的手工确定量化位数的观念,引领了新的自适应量化技术体系.由此可以看出:近年来出现的热门文章提供了不少新的研究方向,极大地促进了模型压缩与加速领域的发展,非常值得收录到我们的综述中,从而为读者带来新的思考.

1.3 主要贡献

对比模型压缩与加速领域已有的综述文章,本文提出的技术分类更加齐全,收录的文章更新颖、热门,对于主流研究方向进行了重点介绍和分析.本文调研了近年来发表在国际顶级会议上的近 200 篇文章,对主流模型压缩与加速方法分类进行了总结和详细分析;同时对一些具有代表性的方法在公开模型上进行了性能对比,探讨了模型压缩与加速领域未来的研究方向.希望本文能给研究者对模型压缩与加速领域有一个全面的了解,抓住热门研究方向,推动未来模型压缩与加速的研究,促进深度学习模型的实际应用.

2 压缩方法概览

本节主要介绍目前主流的模型压缩与加速方法,见表 3,从压缩参数和压缩结构两个角度可以将压缩方法分成以下 7 类.

Table 3 Summarization of methods for deep learning models compression and acceleration

表 3 深度学习模型压缩与加速方法总结

类别	技术	描述
压缩参数	参数剪枝	设计关于参数重要性的评价准则,基于该准则判断网络参数的重要程度,删除冗余参数
	参数量化	将网络参数从 32 位全精度浮点数量化到更低位数
	低秩分解	将高维参数向量降维分解为稀疏的低维向量
	参数共享	利用结构化矩阵或聚类等方法映射网络内部参数
压缩结构	紧凑网络	从卷积核、特殊层和网络结构这 3 个级别设计新型网络
	知识蒸馏	将较大的教师模型的信息提炼到较小的学生模型
混合方式	混合方式	组合使用前述几种方法

2.1 参数剪枝

参数剪枝是指在预训练好的大型模型的基础上,设计对网络参数的评价准则,以此为根据删除“冗余”参数.根据剪枝粒度粗细,参数剪枝可分为非结构化剪枝和结构化剪枝.非结构化剪枝的粒度比较细,可以无限制地去掉网络中期望比例的任何“冗余”参数,但这样会带来裁剪后网络结构不规整、难以有效加速的问题.结构化剪枝的粒度比较粗,剪枝的最小单位是 filter 内参数的组合,通过对 filter 或者 feature map 设置评价因子,甚至可以删除整个 filter 或者某几个 channel,使网络“变窄”,从而可以直接在现有软/硬件上获得有效加速,但可能会带来预测精度(accuracy)的下降,需要通过对模型微调(fine-tuning)以恢复性能.

2.1.1 非结构化剪枝

LeCun 在 20 世纪 80 年代末提出的 OBD(optimal brain damage)算法[19]使用 loss 对参数求二阶导数,以判断参数的重要程度.在此基础上,Hassibi 等人不再限制于 OBD 算法[19]的对角假设,提出了 OBS(optimal brain surgeon)算法[20],除了将次重要权重值置 0 以外,还重新计算其他权重值以补偿激活值,压缩效果更好.与 OBS 算法[20]类似,Srinivas 等人[21]提出了删除全连接层稠密的连接,不依赖训练数据,极大地降低了计算复杂度.最近,Dong 等人[22]提出了逐层 OBS 算法,每一层都基于逐层 loss 函数对相应参数的二阶导数独立剪枝,修剪后,经过轻量再训练以恢复性能.

如图 3 所示,卷积层和全连接层的输入与输出之间都存在稠密的连接,对神经元之间的连接重要性设计评价准则,删除冗余连接,可达到模型压缩的目的.Han 等人[23]提出:根据神经元连接权值的范数值大小,删除范数值小于指定阈值的连接,可重新训练恢复性能.为了避免错误删除重要连接,Guo 等人[24]提出了 DNS(dynamic network surgery)方法,恢复被误删的重要连接.Lin 等人[25]利用生物学上的神经突触概念,定义突触强度为 Batch

Normalization(BN)层放缩因子γ和 filter 的 Frobinus 范数的乘积,用突触强度来表示神经元之间连接的重要性.不同于其他方法在预训练模型上做剪枝,Lee 等人提出的 SNIP(single-shot network pruning)方法[26]在模型初始化阶段,通过对训练集多次采样判断连接的重要性,生成剪枝模板再进行训练,无需迭代进行剪枝-微调的过程.

Fig.3 Pruning connections

图 3 裁剪连接

除了对神经元之间的连接进行评估以外,也可以如图 4 所示,直接对神经元权重进行评估,相比原始权重,3 个 filter 各自进行权重置零操作(即删去某几个小方块),置零的神经元可能各不相同.行列式点过程(determinantal point process,简称 DPP)[27]常用来解决机器学习中的子集选择问题,Mariet 等人[28]将 DPP 应用于神经元的选择,再通过重新加权将删除神经元的信息直接融合到剩余神经元中,这种方法不需要再微调模型.受 Kingma 等人提出的变分 dropout 技术[29]的启发,Molchanov 等人[30]将其用于模型压缩,同时对卷积层和全连接层进行稀疏化.另外,正则化项作为机器学习中 loss 函数的惩罚项常用于对某些参数进行限制,所以关于权重参数的正则化项也可以用于惩罚次重要参数的存在,达到模型压缩的目的.由于参数的 L0 范数不可微分,很难与 loss 共同优化,Louizos 等人[31]对权重设置非负随机门来决定哪些权重设置为 0,转化为可微问题,门上参数可以与原始网络参数共同优化.Tartaglione 等人[32]量化权重参数对于输出的敏感度,将其作为正则化项,逐渐降低敏感度较低的参数值.延迟、能耗等硬件约束条件也可以作为模型压缩的惩罚项,Chen 等人[13]引入硬件约束(例如延迟),使任务目标(如分类精度)最大化,基于权重大小删除范数值较低的权重.Yang 等人[14]利用加权稀疏投影和输入遮蔽来提供可量化的能耗,将能耗预算作为网络训练的优化约束条件,并且由于手工设置的压缩阈值对网络的自适应性不好,使用能恢复误删重要连接的动态剪枝法可获得稀疏网络.Carreira-Perpinán 等人[33]提出交替使用“学习”和“压缩”步骤,探索使 loss 最小化的权重子集的方法.Liu 等人[34]证明卷积可以通过 DCT 域乘法来实现,然后对 filter 的 DCT 系数进行动态剪枝.

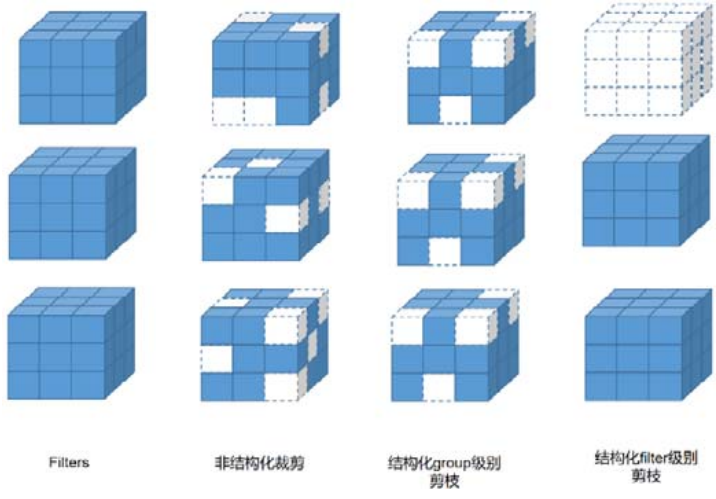

Fig.4 Parameter pruning

图 4 参数剪枝

2.1.2 结构化剪枝

(1) group 级别剪枝

如图 4 所示,group 级别剪枝是指对每一层的 filter 设置相同的稀疏模式(即图中每个立方体都删去相同位置的小方块),变成结构相同的稀疏矩阵.Wen 等人[35]利用 group Lasso 回归进行正则化规约,探索 filter、channel 等不同层次的结构稀疏性.Alvarez 等人[36]提出不需要预训练模型,加入组稀疏正则化项,而是在网络训练的同时自动选择各层神经元数目.Figurnov 等人[37]提出 Perforatedcnns,使用不同策略遮蔽激活值,被遮蔽的值用邻近值表示.Lebedev 等人[38]利用文献[19]中提出的 OBD 算法,将卷积操作视作矩阵乘法计算,以 group 方式稀疏化卷积核,变为稀疏矩阵乘法,提高运算速度.Zhou 等人[39]提出引入稀疏约束,减少最后一个全连接层的参数数量.

(2) filter 级别剪枝

filter 级别剪枝也可以看作 channel 级别剪枝.如图 4 所示,删去该层的某些 filter(即图中删去整个立方体),相当于删去其产生的部分 feature map 和原本需要与这部分 feature map 进行卷积运算的下一层部分 filter.对 filter 的评价准则可分为以下 4 种.

- 基于 filter 范数大小

Li 等人[40]提出计算 filter 的 L1 范数,过滤掉较小 L1 范数的 filter 对应的 feature map,剪枝后再训练.Yang 等人[15]利用 Chen 等人的工作[41]提出的模型能耗工具 Eyeriss 计算每一层能耗,对能耗大的层优先剪枝;同时,为了避免不正确的剪枝,保留剪枝后精确度下降最大的权重.Yang 等人在其另一项工作[42]中提出的 Netadapt 同样也是将硬件度量指标(延迟和能耗等)作为剪枝评价准则,但与文献[15]不同的是:使用经验度量来评估,不需要对平台有详细的了解.算法在移动平台上自动迭代对预训练网络进行剪枝,直到满足资源预算.He 等人[43]提出设置剪枝概率删去 L2 范数最小的几个卷积核,即将该 filter 置 0.其特殊之处在于:每次训练完一个 epoch 进行剪枝,但在上一个 epoch 中被剪枝的 filter 在当前 epoch 训练时仍然参与迭代.

- 自定义 filter 评分因子

Hu 等人[17]提出了 Network trimming 方法,他们认为激活值为 0 的神经元是冗余的,所以统计每一个 filter 中激活值为 0 的数量,将其作为判断一个 filter 是否重要的标准.Liu 等人[44]根据 BN 层放缩因子γ来判断 channel 的重要性.Huang 等人的工作[45]可以看作是文献[44]的泛化,引入了额外的放缩因子对 channel 加以评价.Ye 等人[46]在文献[45]的基础上进行优化,提出了基于 ISTA 和重标技术的梯度学习算法.Dai 等人[47]提出了基于变分信息瓶颈剪枝方法,在每一层只提取与任务相关的信息,将冗余神经元的激活值推向 0.He 等人[48]利用强化学习(reinforcement learning)提供压缩策略,相比于手动启发式方法,效果更好.

- 最小化重建误差

设神经网络中某一卷积层权重为 W,通道数为 C,输入为 X,输出为 Y,忽略偏置项 B,则有:

$$Y=\sum_{c=1}^{C}\sum_{k_1=1}^{K1}\sum_{k_2=1}^{K_2}W_{c,k_1,k_2}\times C_{c,k_1,k_2} \tag{1}$$

令:

$$\hat{X}_c=\sum_{k_1=1}^{K1}\sum_{k_2=1}^{K_2}W_{c,k_1,k_2}\times C_{c,k_1,k_2} \tag{2}$$

则有:

$$Y=\sum_{c=1}^{C}\hat{X}_c \tag{3}$$

令 S 作为从 C 个通道中取得的最优子集,裁剪过程其实就是使子集 S 的最终输出与原始 C 个通道的最终输出 Y 的差别最小.即:

$$\arg\min_{S}\left(Y-\sum_{j\in S}\hat{X}_j\right) \tag{4}$$

Luo 等人[49]提出了 Thinet,“贪婪地”剪去对下一层激活值影响最小的 channel.He 等人[50]并没有像文献[49]那样直接使用贪心策略,而是通过 Lasso 回归对 channel 进行选择性删除,然后利用最小二乘法重构 feature map.Yu 等人[51]定义最后一个与 softmax 层相连的 hidden layer 为 final response layer(FRL),通过特征选择器来

确定各个特征的重要性得分,反向传播,得到整个网络各层的得分,再根据裁剪比率进行裁剪.裁剪的原则是,FRL 输出的重建误差最小.Zhuang 等人[52]引入额外的识别感知 loss,辅助选择真正有助于识别的 channel,联合重建误差共同优化.

- 其他方法

Molchanov 等人[53]将剪枝问题当作一个优化问题,从权重参数中选择一个最优组合,使得 loss 的损失最小,认为剪枝后预测精度衰减小的参数是不重要的.Lin 等人[54]工作的独特之处在于:能够全局性地评估各个 filter 的重要度,动态地、迭代地剪枝,并且能够重新调用之前迭代中错误剪枝的 filter.Zhang 等人[55]将剪枝问题视为具有组合约束条件的非凸优化问题,利用交替方向乘法器(ADMM)分解为两个子问题,可分别用 SGD 和解析法求解.Yang 等人[16]的工作与文献[55]的工作相比,加入能耗作为约束条件,通过双线性回归函数进行建模.

2.2 参数量化

参数量化是指用较低位宽表示典型的 32 位浮点网络参数,网络参数包括权重、激活值、梯度和误差等等,可以使用统一的位宽(如 16-bit、8-bit、2-bit 和 1-bit 等),也可以根据经验或一定策略自由组合不同的位宽.参数量化的优点是:(1) 能够显著减少参数存储空间与内存占用空间,将参数从 32 位浮点型量化到 8 位整型,从而缩小 75%的存储空间,这对于计算资源有限的边缘设备和嵌入式设备进行深度学习模型的部署和使用都有很大的帮助;(2) 能够加快运算速度,降低设备能耗,读取 32 位浮点数所需的带宽可以同时读入 4 个 8 位整数,并且整型运算相比浮点型运算更快,自然能够降低设备功耗.但其仍存在一定的局限性,网络参数的位宽减少损失了一部分信息量,会造成推理精度的下降,虽然能够通过微调恢复部分精确度,但也带来时间成本的增加;量化到特殊位宽时,很多现有的训练方法和硬件平台不再适用,需要设计专用的系统架构,灵活性不高.

2.2.1 二值化

二值化是指限制网络参数取值为 1 或−1,极大地降低了模型对存储空间和内存空间的需求,并且将原来的乘法操作转化成加法或者移位操作,显著提高了运算速度,但同时也带来训练难度和精度下降的问题.

(1) 二值化权重

由于权重占据网络参数的大部分,一些研究者提出对网络权重进行二值化,以达到压缩网络的目的.Courbariaux 等人[56]提出了 Binaryconnect,将二值化策略用于前向计算和反向传播,但在使用随机梯度更新法(SGD)更新参数时,仍需使用较高位宽.Hou 等人[57]提出一种直接考虑二值化权重对 loss 产生影响的二值化算法,采用对角海森近似的近似牛顿算法得到二值化权重.Xu 等人[58]提出局部二值卷积(LBC)来替代传统卷积,LBC 由一个不可学习的预定义 filter、一个非线性激活函数和一部分可以学习的权重组成,其组合达到与激活的传统卷积 filter 相同的效果.Guo 等人[59]提出了 Network sketching 方法,使用二值权重共享的卷积,即:对于同层的卷积运算(即拥有相同输入),保留前一次卷积的结果,卷积核的相同部分直接复用结果.McDonnell 等人[60]将符号函数作为实现二值化的方法.Hu 等人[61]通过哈希将数据投影到汉明空间,将学习二值参数的问题转化为一个在内积相似性下的哈希问题.

(2) 二值化权重和激活值

在二值化网络权重的基础上,研究人员提出可以同时二值化权重和激活值,以加快推理速度.Courbariaux 等人[62]首先提出了 Binarized neural network(BNN),将权重和激活值量化到±1.Rastegari 等人[63]在文献[62]的基础上提出了 Xnor-net,将卷积通过 xnor 和位操作实现,从头训练一个二值化网络.Li 等人[64]在 Xnor-net[63]的基础上改进其激活值量化,提出了 High-order residual quantization(HORQ)方法.Liu 等人[65]提出了 Bi-real net,针对 Xnor-net[63]进行网络结构改进和训练优化.Lin 等人[66]提出了 ABC-Net,用多个二值操作加权来拟合卷积操作.

2.2.2 三值化

三值化是指在二值化的基础上引入 0 作为第 3 阈值,减少量化误差.Li 等人[67]提出了三元权重网络 TWN,将权重量化为$\{-w,0,+w\}$.不同于传统的 1 或者权重均值,Zhu 等人[68]提出了 Trained ternary quantization(TTQ),使用两个可训练的全精度放缩系数,将权重量化到$\{-w_n,0,w_p\}$,权重不对称使网络更灵活.Achterhold 等人[69]提出了 Variational network quantization(VNQ),将量化问题形式化为一个变分推理问题.引入量化先验,最后可以用

确定性量化值代替权值.

2.2.3 聚类量化

当参数数量庞大时,可利用聚类方式进行权重量化.Gong 等人[70]最早提出将 *k*-means 聚类用于量化全连接层参数,如图 5 所示,对原始权重聚类形成码本,为权值分配码本中的索引,所以只需存储码本和索引,无需存储原始权重信息.Wu 等人[71]将 *k*-means 聚类拓展到卷积层,将权值矩阵划分成很多块,再通过聚类获得码本,并提出一种有效的训练方案抑制量化后的多层累积误差.Choi 等人[72]分析了量化误差与 loss 的定量关系,确定海森加权失真测度是量化优化的局部正确目标函数,提出了基于海森加权 *k*-means 聚类的量化方法.Xu 等人[73]提出了分别针对不同位宽的 Single-level network quantization(SLQ)和 Multi-level network quantization(MLQ)两种方法,SLQ 方法针对高位宽,利用 *k*-means 聚类将权重分为几簇,依据量化 loss,将簇分为待量化组和再训练组,待量化组的每个簇用簇内中心作为共享权重,剩下的参数再训练.而 MLQ 方法针对低位宽,不同于 SLQ 方法一次量化所有层,MLQ 方法采用逐层量化的方式.

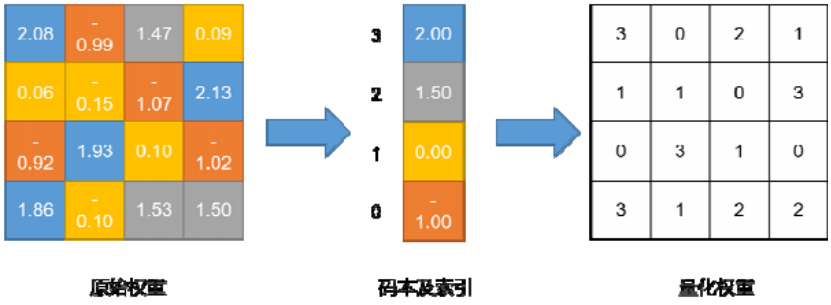

Fig.5 Flow chart of clustering quantization

图 5 聚类量化流程图

2.2.4 混合位宽

(1) 手工固定

由于二值网络会降低模型的表达能力,研究人员提出,可以根据经验手工选定最优的网络参数位宽组合.Lin 等人[74]在 BNN[62]的基础上,提出把 32-bit 权重概率性地转换为二元和三元值的组合.Zhou 等人[75]提出了 DoReFa-Net,将权重和激活值分别量化到 1-bit 和 2-bit.Mishra 等人[76]提出了 WRPN,将权重和激活值分别量化到 2-bit 和 4-bit.Köster 等人[77]提出的 Flexpoint 向量有一个可动态调整的共享指数,证明 16 位尾数和 5 位共享指数的 Flexpoint 向量表示在不修改模型及其超参数的情况下,性能更优.Wang 等人[78]使用 8 位浮点数进行网络训练,部分乘积累加和权重更新向量的精度从 32-bit 降低到 16-bit,达到与 32-bit 浮点数基线相同的精度水平.除了权重和激活值,研究者们将梯度和误差也作为可优化的因素.这些同时考虑了权重、激活值、梯度和误差的方法的量化位数和特点对比可见表 4.表中的 *W*、*A*、*G* 和 *E* 分别代表权重、激活值、梯度和误差.

Table 4 Comparison of several mixed bit-width quantization methods

表 4 几种混合位宽量化法对比

方法	*W*	*A*	*G*	*E*	特点
Ref.[79]	16	32	32	32	引入随机舍入技术
Ref.[80]	32	8	8	32	加速数据传输,提高并行训练的性能
Ref.[81]	8	8	32	32	仅测试时使用整数运算
Ref.[82]	8	8	8	32	计算梯度的最后一个步骤保留更高精度
Ref.[83]	2	8	8	8	离散训练和推理

(2) 自主确定

由于手工确定网络参数位宽存在一定的局限性,可以设计一定的策略,以帮助网络选择合适的位宽组合.Khoram 等人[18]迭代地使用 loss 的梯度来确定每个参数的最优位宽,使得只有对预测精度重要的参数才有高精

度表示.Wang 等人[84]提出两步量化方法:先量化激活值再量化权重.针对激活值量化,提出了稀疏量化方法.对于权重量化,将其看成非线性最小二乘回归问题.Faraone 等人[85]提出了基于梯度的对称量化方法 SYQ,设计权值二值化或三值化,并在 pixel 级别、row 级别和 layer 级别定义不同粒度的缩放因子以估计网络权重;至于激活值,则量化为 2-bit 到 8-bit 的定点数.Zhang 等人[86]提出了 Learned quantization(LQ-nets),使量化器可以与网络联合训练,自适应地学习最佳量化位宽.

2.2.5 训练技巧

由于量化网络的网络参数不是连续的数值,所以不能像普通的卷积神经网络那样直接使用梯度下降方法进行训练,而需要特殊的方法对这些离散的参数值进行处理,使其不断优化,最终实现训练目标.Zhou 等人[87]提出了一种增量式网络量化方法 INQ,先对权重进行划分,将对预测精度贡献小的权重划入量化组;然后通过再训练恢复性能.Cai 等人[88]提出了 Halfwave Gaussian quantizer(HWGQ)方法,设计了两个 ReLU 非线性逼近器(前馈计算中的半波高斯量化器和反向传播的分段连续函数),以训练低精度的深度学习网络.Leng 等人[89]提出,利用 ADMM[90]解决低位宽网络训练问题.Zhuang 等人[91]针对低位宽卷积神经网络提出 3 种训练技巧,以期得到较高精度.Zhou 等人[92]提出一种显式的 loss-error-aware 量化方法,综合考虑优化过程中的 loss 扰动和权值近似误差,采用增量量化策略.Park 等人[93]提出了价值感知量化方法来降低训练中的内存成本和推理中的计算/内存成本,并且提出一种仅在训练过程中使用量化激活值的量化反向传播方法.Shayer 等人[94]展示了如何通过对局部再参数化技巧的简单修改,来实现离散权值的训练.该技巧以前用于训练高斯分布权值.Louizos 等人[95]引入一种可微的量化方法,将网络的权值和激活值的连续分布转化为量化网格上的分类分布,随后被放宽到连续代理,可以允许有效的基于梯度的优化.

2.3 低秩分解

神经网络的 filter 可以看作是四维张量:宽度 w×高度 h×通道数 c×卷积核数 n,由于 c 和 n 对网络结构的整体影响较大,所以基于卷积核(w×h)矩阵信息冗余的特点及其低秩特性,可以利用低秩分解方法进行网络压缩.低秩分解是指通过合并维数和施加低秩约束的方式稀疏化卷积核矩阵,由于权值向量大多分布在低秩子空间,所以可以用少数的基向量来重构卷积核矩阵,达到缩小存储空间的目的.低秩分解方法在大卷积核和中小型网络上有不错的压缩和加速效果,过去的研究已经比较成熟,但近两年已不再流行.原因在于:除了矩阵分解操作成本高、逐层分解不利于全局参数压缩,需要大量的重新训练才能达到收敛等问题之外,近两年提出的新网络越来越多地采用 1×1 卷积,这种小卷积核不利于低秩分解方法的使用,很难实现网络压缩与加速.

2.3.1 二元分解

Jaderberg 等人[96]将 w×h 的卷积核分解为 w×1 和 1×h 的卷积核,学习到的字典权重线性组合重构,得到输出 feature map.Liu 等人[97]使用两阶段分解法研究 filter 的通道间和通道内冗余.Tai 等人[98]提出一种计算低秩张量分解的新算法,利用 BN 层转换内部隐藏单元的激活.Masana 等人[99]主要解决在大数据集上训练的网络在小目标域的使用问题,证明在压缩权重时考虑激活统计量,会导致一个具有闭型解的秩约束回归问题.Wen 等人[100]提出了 Force regularization,将更多权重信息协调到低秩空间中.Wang 等人[101]提出了定点分解,再通过伪全精度权重复原,权重平衡和微调恢复性能.与其他基于 filter 空间或信道数的低秩分解算法不同,Peng 等人[102]的工作基于 filter 组近似,达到降低参数冗余的目的.Qiu 等人[103]提出将 filter 分解为带预固定基的截断展开,展开系数从数据中学习.Novikov 等人[104]提出 Tensor train 分解来压缩全连接层的稠密权值矩阵,而 Garipov 等人[105]将其推广到卷积层.Wang 等人[106]提出了 Tensor ring 分解,用于压缩卷积层和全连接层.

2.3.2 多元分解

对 filter 的二元分解会引入 w×h×c×d 张量和 d×n 张量,由于第 1 个张量 w×h×c×d 很大并且耗时,三元分解提出对其进行分解.Kim 等人[107]提出了 Tucker 分解,对第 1 个张量沿输入通道维进行二元分解,得到 w×1、1×h 和 1×1 的卷积.由于第 2 个分量 d×n 也需要大量计算,但其在输入和输出通道维数上的秩已经很低,Wang 等人[108]提出了基于低秩和群稀疏分解的块项分解(BTD),用一些较小的子张量之和近似原始权重张量.在三元分解的基础上,Lebedev 等人[109]提出了 CP 分解,即位 tensor 分解,将四维卷积核分解成 4 个:1×1、w×1、1×h 和 1×1

的卷积,即,将 1 层网络分解为 5 层低复杂度的网络层.

2.4 参数共享

参数共享是指利用结构化矩阵或聚类等方法映射网络参数,减少参数数量.参数共享方法的原理与参数剪枝类似,都是利用参数存在大量冗余的特点,目的都是为了减少参数数量.但与参数剪枝直接裁剪不重要的参数不同,参数共享设计一种映射形式,将全部参数映射到少量数据上,减少对存储空间的需求.由于全连接层参数数量较多,参数存储占据整个网络模型的大部分,所以参数共享对于去除全连接层冗余性能够发挥较好的效果;也由于其操作简便,适合与其他方法组合使用.但其缺点在于不易泛化,如何应用于去除卷积层的冗余性仍是一个挑战.同时,对于结构化矩阵这一常用映射形式,很难为权值矩阵找到合适的结构化矩阵,并且其理论依据不够充足.

2.4.1 循环矩阵

如果一个大小为 $m\times n$ 的矩阵能够用少于 $m\times n$ 个参数来描述,这个矩阵就是一个结构化矩阵.循环矩阵作为结构化矩阵的一种,是参数共享法常用的一种映射形式.令向量:

$$r=(r_0,r_1,\ldots,r_{d-1}) \tag{5}$$

循环矩阵的每一行都是由上一行的各元素依次右移一个位置得到,即:

$$R=cir(r)=\begin{bmatrix} r_0 & r_1 & \cdots & r_{d-2} & r_{d-1} \\ r_{d-1} & r_0 & \cdots & r_{d-3} & r_{d-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_2 & r_3 & \cdots & r_0 & r_1 \\ r_1 & r_2 & \cdots & r_{d-1} & r_0 \end{bmatrix} \tag{6}$$

Cheng 等人[110]提出用循环投影代替传统的线性投影.对于具有 d 个输入节点和 d 个输出节点的神经网络层,将时间复杂度从 $O(d^2)$降低到 $O(d\times\log d)$,空间复杂度从 $O(d^2)$降低到 $O(d)$.Wang 等人[111]利用循环矩阵来构造特征图,对 filter 进行重新配置,建立从原始输入到新的压缩特征图的映射关系.Sindhwani 等人[112]提出一个统一的框架来学习以低位移秩(LDR)为特征的结构参数矩阵.Zhao 等人[113]证明:具有 LDR 权值矩阵的神经网络,在保持较高精度的同时,可以显著降低空间和计算复杂度.Le 等人[114]提出 Fastfood 变换,通过一系列简单矩阵的乘法来代替稠密矩阵与向量的乘积,这些简单矩阵通过规则一次生成,后面无需调整.Yang 等人[115]在文献[114]的基础上提出自适应 Fastfood 变换,对全连接层的矩阵-向量乘法进行重新参数化,替换成 Fastfood 层.

2.4.2 聚类共享

Chen 等人[116,117]使用哈希函数将网络参数随机分组到哈希桶中,同一个桶的参数共享一个通过标准反向传播学习到的值.Wu 等人[118]提出对权值进行 k-means 聚类,并引入一种新的频谱松弛的 k-means 正则化方法.Son 等人[119]将 k-means 聚类应用于 3×3 卷积核,一个 filter 用放缩因子×聚类中心来表示.

2.4.3 其他方法

Reagen 等人[120]提出了有损权值编码方案 Bloomier filter,以引入随机误差为代价来节省空间,利用神经网络的容错能力进行再训练.Havasi 等人[121]提出了放松权重决定论,使用权重上的全变分分布,实现更加有效的编码方案,以提高压缩率.Jin 等人[122]提出了 Weight Sampling Network (WSNet),沿着空间维度和通道维度进行加权采样.Kossaifi 等人[123]提出了 Tensorized-network(T-net),使用单个高阶张量来参数化地表示整个网络.

2.5 紧凑网络

以上 4 种利用参数冗余性减少参数数量或者降低参数精度的方法虽然能够精简网络结构,但往往需要庞大的预训练模型,在此基础上进行参数压缩,并且这些方法大都存在精确度下降的问题,需要微调来提升网络性能.设计更紧凑的新型网络结构,是一种新兴的网络压缩与加速理念,构造特殊结构的 filter、网络层甚至网络,从头训练,获得适宜部署到移动平台等资源有限设备的网络性能,不再需要像参数压缩类方法那样专门存储预训练模型,也不需要通过微调来提升性能,降低了时间成本,具有存储量小、计算量低和网络性能好的特点.但其

缺点在于:由于其特殊结构很难与其他的压缩与加速方法组合使用,并且泛化性较差,不适合作为预训练模型帮助其他模型训练.

2.5.1 卷积核级别

(1) 新型卷积核

Iandola 等人[124]提出了 Squeezenet,使用 1×1 卷积代替 3×3 卷积,为了减少 feature map 的数量,将卷积层转变成两层:squeeze 层和 expand 层,减少了池化层.Howard 等人[125]提出了 MobileNet,将普通卷积拆分成 depth-wise 卷积和 point-wise 卷积,减少了乘法次数.Sandler 等人[126]提出的 MobileNetV2 相比 MobileNet[125],在 depth-wise 卷积之前多加了一个 1×1 expand 层以提升通道数,获得了更多的特征.Zhang 等人[127]提出了 ShuffleNet,为克服 point-wise 卷积的昂贵成本和通道约束,采用了逐点组卷积(point-wise group convolution)和通道混洗(channel shuffle)的方式.Ma 等人[128]提出的 ShuffleNetV2 相比 ShuffleNet[127],为了减少内存访问成本,提出了通道分割(channel split)这一概念.Zhang 等人[129]提出了交错组卷积(IGC),引入第 2 次组卷积,其每组输入通道来自于第 1 次组卷积中不同的组,从而与第 1 次组卷积交替互补.Xie 等人[130]在文献[129]的基础上进行泛化,提出交错的稀疏化组卷积,将两个结构化稀疏卷积核组成的构建块扩展到多个.Wan 等人[131]提出了完全可学习的组卷积模块(FLGC),可以嵌入任何深度神经网络进行加速.Park 等人[132]提出了直接稀疏卷积,用于稠密的 feature map 与稀疏的卷积核之间的卷积操作.Zhang 等人[133]证明:高性能的直接卷积在增加线程数时性能更好,消除了所有内存开销.

(2) 简单 filter 组合

Ioannou 等人[134]提出了从零开始学习一组小的不同空间维度的基 filter,在训练过程中,将这些基 filter 组合成更复杂的 filter.Bagherinezhad 等人[135]提出对每层构建一个字典,每个 filter 由字典中的某些向量线性组合得到.将输入向量和整个字典里的向量进行卷积,查表得到该输入向量和 filter 的卷积结果.Wang 等人[136]提出了构建高效 CNN 的通用 filter,二级 filter 从主 filter 中继承,通过整合从不同感受域提取的信息来增强性能.

2.5.2 层级别

Huang 等人[137]提出了随机深度用于类似 ResNet 含残差连接的网络的训练,对于每个 mini-batch,随机删除 block 子集,并用恒等函数绕过它们.Dong 等人[138]为每个卷积层配备一个低成本协同层(LCCL),预测哪些位置的点经过 ReLU 后会变成 0,测试时忽略这些位置的计算.Li 等人[139]将网络层分为权重层(如卷积层和全连接层)和非权重层(如池化层、ReLU 层等),提出了将非权重层与权重层进行合并的方法,去除独立的非权重层后,运行时间显著减少.Prabhu 等人[140]使用同时稀疏且连接良好的图来建模卷积神经网络 filter 之间的连接.Wu 等人[141]通过平移 feature map 的形式取代传统的卷积,从而减小了计算量.Chen 等人[142]引入稀疏移位层(SSL)来构造高效的卷积神经网络.在该体系结构中,基本块仅由 1×1 卷积层组成,对中间的 feature map 只进行少量的移位操作.

2.5.3 网络结构级别

Kim 等人[143]提出了 SplitNet,自动学会将网络层分成多组,获得一个树形结构的网络,每个子网共享底层权重.Gordon 等人[144]提出了 Morphnet,通过收缩和扩展阶段循环优化网络:在收缩阶段,通过稀疏正则化项识别效率低的神经元从网络中去除;在扩展阶段,使用宽度乘数来统一扩展所有层的大小,所以含重要神经元更多的层拥有更多计算资源.Kim 等人[145]提出了嵌套稀疏网络 NestedNet,每一层由多层次的网络组成,高层次网络与低层次网络以 Network in network (NIN)的方式共享参数:低层次网络学习公共知识,高层次网络学习特定任务的知识.

2.6 知识蒸馏

知识蒸馏最早由 Buciluǎ 等人[146]提出,用以训练带有伪数据标记的强分类器的压缩模型和复制原始分类器的输出.与其他压缩与加速方法只使用需要被压缩的目标网络不同,知识蒸馏法需要两种类型的网络:教师模型和学生模型.预先训练好的教师模型通常是一个大型的神经网络模型,具有很好的性能.如图 6 所示,将教师模型的 softmax 层输出作为 soft target 与学生模型的 softmax 层输出作为 hard target 一同送入 total loss 计算,指导

学生模型训练,将教师模型的知识迁移到学生模型中,使学生模型达到与教师模型相当的性能.学生模型更加紧凑高效,起到模型压缩的目的.知识蒸馏法可使深层网络变浅,极大地降低了计算成本,但也存在其局限性.由于使用 softmax 层输出作为知识,所以一般多用于具有 softmax 损失函数的分类任务,在其他任务的泛化性不好;并且就目前来看,其压缩比与蒸馏后的模型性能还存在较大的进步空间.

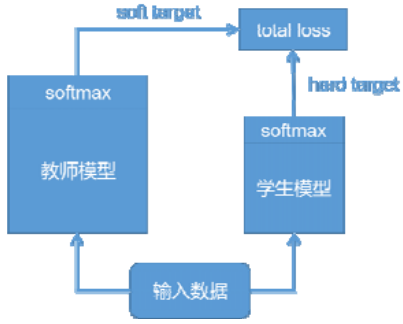

Fig.6 Flow chart of knowledge distillation

图 6 知识蒸馏流程图

2.6.1 学生模型的网络结构

知识蒸馏法的研究方向之一就是如何为学生模型选择合适的网络结构,帮助学生模型更好地学习教师模型的知识.Ba 等人[147]提出:在保证教师模型和学生模型网络参数数量相同的情况下,设计更浅的学生模型,每一层变得更宽.Romero 等人[148]与文献[147]的观点不同,他们认为更深的学生模型分类效果更好,提出 Fitnets 使用教师网络的中间层输出 Hints,作为监督信息训练学生网络的前半部分.Chen 等人[149]提出使用生长式网络结构,以复制的方式重用预训练的网络参数,在此基础上进行结构拓展.Li 等人[150]与文献[149]观点一致,提出分别从宽度和深度上进行网络生长.Crowley 等人[151]提出将知识蒸馏与设计更紧凑的网络结构相结合,将原网络作为教师模型,将使用简化卷积的网络作为学生模型.Zhu 等人[152]提出基于原始网络构造多分支结构,将每个分支作为学生网络,融合生成推理性能更强的教师网络.

2.6.2 教师模型的学习信息

除了使用 softmax 层输出作为教师模型的学习信息以外,有研究者认为,可以使用教师模型中的其他信息帮助知识迁移.Hinton 等人[153]首先提出使用教师模型的类别概率输出计算 soft target,为了方便计算,还引入温度参数.Yim 等人[154]将教师模型网络层之间的数据流信息作为学习信息,定义为两层特征的内积.Chen 等人[155]将教师模型在某一类的不同样本间的排序关系作为学习信息传递给学生模型.

2.6.3 训练技巧

Czarnecki 等人[156]提出了 Sobolev 训练方法,将目标函数的导数融入到神经网络函数逼近器的训练中.当训练数据由于隐私等问题对于学生模型不可用时,Lopes 等人[157]提出了如何通过 extra metadata 来加以解决的方法.Zhou 等人[158]的工作主要有两个创新点:第一,不用预训练教师模型,而是教师模型和学生模型同时训练;第二,教师模型和学生模型共享网络参数.

2.6.4 其他场景

由于 softmax 层的限制,知识蒸馏法被局限于分类任务的使用场景.但近年来,研究人员提出多种策略使其能够应用于其他深度学习场景.在目标检测任务中,Li 等人[159]提出了匹配 proposal 的方法,Chen 等人[160]结合使用文献[148,153]提出的方法,提升多分类目标检测网络的性能.在解决人脸检测任务时,Luo 等人[161]提出将更高隐层的神经元作为学习知识,其与类别输出概率信息量相同,但更为紧凑.Gupta 等人[162]提出了跨模态迁移知识的做法,将在 RGB 数据集学习到的知识迁移到深度学习的场景中.Xu 等人[163]提出一种多任务指导预测和蒸馏网络(PAD-net)结构,产生一组中间辅助任务,为学习目标任务提供丰富的多模态数据.

2.7 混合方式

以上这些压缩与加速方法单独使用时能够获得很好的效果,但也都存在各自的局限性,组合使用可使它们互为补充.研究人员通过组合使用不同的压缩与加速方法或者针对不同网络层选取不同的压缩与加速方法,设计了一体化的压缩与加速框架,能够获得更好的压缩比与加速效果.参数剪枝、参数量化、低秩分解和参数共享经常组合使用,极大地降低了模型的内存需求和存储需求,方便模型部署到计算资源有限的移动平台[164].知识蒸馏可以与紧凑网络组合使用,为学生模型选择紧凑的网络结构,在保证压缩比的同时,可提升学生模型的性能.混合方式能够综合各类压缩与加速方法的优势,进一步加强了压缩与加速效果,将会是未来在深度学习模型压缩与加速领域的重要研究方向.

2.7.1 组合参数剪枝和参数量化

Ullrich 等人[165]基于 Soft weight sharing 的正则化项,在模型再训练过程中实现了参数量化和参数剪枝.Tung 等人[166]提出了参数剪枝和参数量化的一体化压缩与加速框架 Compression learning by in parallel pruning-quantization(CLIP-Q).如图 7 所示,Han 等人[167]提出了 Deep compression,将参数剪枝、参数量化和哈夫曼编码相结合,达到了很好的压缩效果;并在其基础上考虑到软/硬件的协同压缩设计,提出了 Efficient inference engine(Eie)框架[168].Dubey 等人[169]同样利用这 3 种方法的组合进行网络压缩.

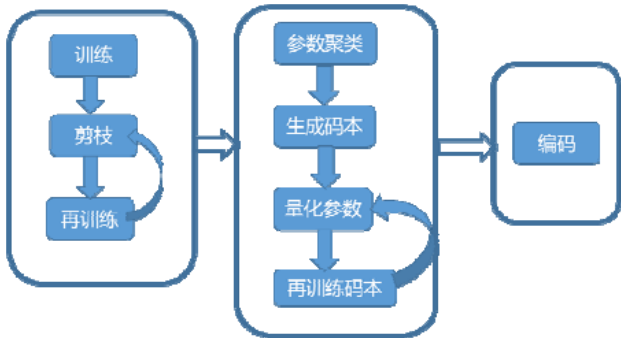

Fig.7 Flow chart of Deep Compression[167]

图 7 Deep Compression[167]流程图

2.7.2 组合参数剪枝和参数共享

Louizos 等人[170]采用贝叶斯原理,通过先验分布引入稀疏性对网络进行剪枝,使用后验不确定性确定最优的定点精度来编码权重.Ji 等人[171]通过重新排序输入/输出维度进行剪枝,并将具有小值的不规则分布权重聚类到结构化组中,实现更好的硬件利用率和更高的稀疏性.Zhang 等人[172]不仅采用正则化器鼓励稀疏性,同时也学习哪些参数组应共享一个公共值以显式地识别出高度相关的神经元.

2.7.3 组合参数量化和知识蒸馏

Polino 等人[173]提出了加入知识蒸馏 loss 的量化训练方法,有浮点模型和量化模型,用量化模型计算前向 loss,并对其计算梯度,用以更新浮点模型.每次前向计算之前,用更新的浮点模型更新量化模型.Mishra 等人[174]提出用高精度教师模型指导低精度学生模型的训练,有 3 种思路:教师模型和量化后的学生模型联合训练;预训练的教师模型指导量化的学生模型从头开始训练;教师模型和学生模型都进行了预训练,但学生模型已经量化,之后在教师模型的指导下再进行微调.

3 压缩效果比较

我们从以上介绍的 7 种主流网络压缩技术中选出其中一些具有代表性的方法,按照文献中声明的压缩与加速效果进行对比.通过对相关文献中使用较多的数据集和模型的统计,我们使用 MNIST[175]、CIFAR-10[176]和 ImageNet[177]这三大常用数据集,在 LeNet、AlexNet、VGG-16、ResNet 等公开深度模型上进行压缩方法测试,

比较其压缩效果.图表中的Δ*accuracy*=压缩后模型 *accuracy*–原始模型 *accuracy*,*#Params*↓=原始模型参数量/压缩后模型参数量,*#FLOPs*↓=原始模型浮点计算次数/加速后模型浮点计算次数.Weight bits 和 activation bits 分别代表权值和激活值被量化后的表示位数.*T-accuracy* 代表教师模型的 *accuracy*,*S-accuracy* 代表学生模型的 *accuracy*.*T-#Params* 代表教师模型的参数数量,*S-#Params* 代表学生模型的参数数量.

表 5 展示了参数剪枝、紧凑网络、参数共享、知识蒸馏和混合方式这 5 类压缩技术的一些代表性方法使用 MNIST 数据集在 LeNet-5 上的压缩效果,可以看出,除了文献[157]带来较大的 *accuracy* 损失以外,其他方法的压缩效果都不错.从 *accuracy* 的角度来看,自适应 fastfood 变换[115]的效果更好,在达到压缩效果的同时,还提升了 *accuracy*;从参数压缩量的角度来看,混合方式在 *accuracy* 轻微下降的情况下,都实现了较大的压缩比,其中,文献[169]的效果最好.

Table 5 Compression effects of different compression methods on LeNet-5 on MNIST

表 5 不同压缩方法在 LeNet-5 on MNIST 上的压缩效果

分类	方法	Δ*accuracy*	*#Params*↓
参数剪枝	Ref.[23]	+0.03	12x
	DNS[24]	0	108x
	Ref.[30]	–0.1	63x
	FDNP[34]	0	130x
紧凑网络	Versatile[136]	+0.02	1.97x
参数共享	Circulant projections[110]	–0.03	5.8x
	Adaptive fastfood[115]	+0.16	11.1x
知识蒸馏	Ref.[157]	–6.18	2x
混合方式	Soft weight sharing[165]	–0.09	162x
	Deep Compression[167]	+0.06	39x
	Ref.[169]	–0.01	193x
	Ref.[170]	–0.1	156x
	Ref.[171]	0	10x

表 6 展示了参数剪枝、紧凑网络、参数共享和混合方式这 4 类压缩技术的一些代表性方法使用 CIFAR-10 数据集在 VGG-16 上的压缩效果,可以看出,这 4 类方法的压缩效果差别比较大.整体来看,结构化剪枝[40]效果更好,同时起到了网络压缩和加速的效果,*accuracy* 甚至有些提升.权值随机编码方法[121]能够实现高达 159x 的参数压缩比,*accuracy* 略有下降.

Table 6 Compression effects of different compression methods on VGG-16 on CIFAR-10

表 6 不同压缩方法在 VGG-16 on CIFAR-10 上的压缩效果

分类	方法	Δ*accuracy*	*#Params*↓	*#FLOPs*↓
参数剪枝	Ref.[40]	+0.15	2.78x	1.52x
紧凑网络	Ref.[140]	–1.0	13x	–
参数共享	Ref.[119]	–0.8	13.7x	7.63x
	MIRACLE[121]	–0.07	159x	–
混合方式	Ref.[170]	–0.2	14x	–
	Ref.[172]	–0.8	14.5x	–

表 7 展示了参数剪枝、紧凑网络、低秩分解、参数共享和混合方式这 5 类压缩技术的一些代表性方法使用 ImageNet 数据集在 AlexNet 上的压缩效果.整体来看,5 类方法达到的压缩效果和加速效果比较均衡,*accuracy* 都略有下降.其中,参数剪枝和混合方式能够实现更大的压缩比,但低秩分解的加速效果更好;另两类方法都有不同程度的 *accuracy* 下降.

表 8 展示了参数剪枝、低秩分解、参数共享和混合方式这 4 类压缩技术的一些代表性方法在使用 ImageNet 数据集在 VGG-16 上的压缩效果.整体的压缩与加速效果都很明显,其中,剪枝方法的 *accuracy* 略微有所提升;混合方式达到的压缩比最高.另外两类方法虽然 *accuracy* 有所下降,但加速效果更优秀.

表 9 展示了参数剪枝、紧凑网络、参数共享和混合方式这 4 类压缩技术的一些代表性方法在使用 ImageNet 数据集在 ResNet-50 上的压缩效果.整体来看,*accuracy* 的下降趋势比较明显,压缩与加速效果不如其他网络在

ImageNet 上的好.其中,混合方式压缩效果最好,文献[169]达到 15.8x 的压缩比;而在参数共享方法中,循环矩阵[111]达到了最高加速比 5.82x.

Table 7 Compression effects of different compression methods on AlexNet on ImageNet

表 7 不同压缩方法在 AlexNet on ImageNet 上的压缩效果

分类	方法	Top-1 Δ*accuracy*	Top-5 Δ*accuracy*	*#Params*↓	*#FLOPs*↓
参数剪枝	Ref.[23]	+0.01	+0.06	9x	–
	Ref.[24]	–0.31	–	17.7x	–
	FDNP[34]	+0.26	+0.14	20.9x	–
	PerforatedCNNs[37]	–	–3.2	2.0x	2.6x
	Ref.[54]	–1.78	–1.15	–	2.77x
紧凑网络	LCNN[135]	–1.5	–2.1	–	3.2x
	Versatile[136]	–1.2	–0.9	3.15x	–
	Ref.[140]	–2.0	–	8.03x	–
	SplitNet[143]	–1.3	–	2.28x	1.03x
低秩分解	Ref.[98]	–	-0.37	5x	1.82x
	Ref.[100]	–	–1.71	–	4.05x
	Tucker[107]	–	–1.70	5.46x	2.67x
参数共享	Ref.[110]	–0.4	–0.7	11.27x	–
	RedCNN[111]	–0.3	–0.1	–	4.31x
	Adaptive fastfood[115]	–0.12	–	3.7x	–
混合方式	CLIP-Q [166]	+0.7	–	51x	–
	Deep compression[167]	0	+0.03	35x	–
	Ref.[169]	–0.84	–	19.1x	–

Table 8 Compression effects of different compression methods on VGG-16 on ImageNet

表 8 不同压缩方法在 VGG-16 on ImageNet 上的压缩效果

分类	方法	Top-1 Δ*accuracy*	Top-5 Δ*accuracy*	*#Params*↓	*#FLOPs*↓
参数剪枝	Ref.[23]	+0.16	+0.44	13x	–
	PerforatedCNNs[37]	–	–6.8	2.4x	2.8x
	Trimming[17]	+2.08	+1.35	2.59x	–
	Thinet[49]	–1.0	+1.35	16.6x	3.3x
	GDP[54]	–2.81	–1.47	–	4.08x
低秩分解	Ref.[98]	–	–0.29	2.75x	2.05x
	Tucker[107]	–	–0.5	1.09x	4.93x
参数共享	RedCNN[111]	–0.3	–0.1	–	9.63x
混合方式	Deep compression[167]	+0.43	+0.41	49x	–
	Ref.[169]	–0.98	–	17x	–
	Ref.[171]	–2.356	–	2.35x	–

Table 9 Compression effects of different compression methods on ResNet-50 on ImageNet

表 9 不同压缩方法在 ResNet-50 on ImageNet 上的压缩效果

分类	方法	Top-1 Δ*accuracy*	Top-5 Δ*accuracy*	*#Params*↓	*#FLOPs*↓
参数剪枝	Ref.[23]	–0.01	–	6.2x	–
	Ref.[25]	–0.62	+0.6	4.34x	–
	Thinet[49]	–4.46	–2.84	2.95x	3.51x
	DCP[52]	–1.06	–0.61	2.06x	2.25x
	GDP[54]	–4.21	–2.16	–	2.46x
紧凑网络	Versatile[136]	-0.8	–0.4	2.33x	–
	Ref.[140]	–1.61	–	2x	–
参数共享	RedCNN[111]	–1.1	–0.5	–	5.82x
混合方式	CLIP-Q[166]	+0.6	–	15x	–
	Ref.[169]	–0.01	–	15.8x	–

表 10 展示了一些主流量化技术使用 ImageNet 数据集在 AlexNet 上的压缩效果,其中,weight bits 为 1 表示二值化网络,weight bits 为 2 表示三值化网络.除此之外还有一些特殊位宽,其中,文献[89]中的 3{±2}表示权值从{0,±1,±2}中选择,3{±2 ±4}表示权值从{0,±1,±2,±4}中选择.XNOR-Net[63]虽然能够达到比较好的压缩性能,但

accuracy 损失太大.SYQ[85]在实现权重二值化、三值化的同时,将激活值也量化到 8 位,*accuracy* 几乎没有损失,还略有提升.

Table 10 Compression effects of different quantization methods on AlexNet on ImageNet

表 10 不同量化方法在 AlexNet on ImageNet 上的压缩效果

方法	Top-1 Δ*accuracy*	Top-5 Δ*accuracy*	Weight bits	Activation bits
XNOR-net[63]	−12.4	−11.0	1	1
SYQ[85]	0	−0.8	1	8
	+1.5	+0.6	2	8
Ref.[89]	−3.0	−2.7	1	32
	−1.8	−1.8	2	32
	−0.8	−0.6	3{±2}	32
	0	−0.2	3{±2 ±4}	32
TTQ[68]	+0.3	−0.6	2	32
SLQ[73]	+0.46	+0.3	5	32
INQ[87]	+0.15	+0.23	5	32

表 11 展示了一些主流量化技术使用 ImageNet 数据集在 ResNet-18 上的压缩效果.整体来看,*accuracy* 的下降程度更大,对权值和激活值的大尺度量化带来不同程度的精度损失,SLQ[73]和 INQ[87]将权值量化到 5 位,*accuracy* 略有提升.

Table 11 Compression effects of different quantization methods on ResNet-18 on ImageNet

表 11 不同量化方法在 ResNet-18 on ImageNet 上的压缩效果

方法	Top-1 Δ*accuracy*	Top-5 Δ*accuracy*	Weight bits	Activation bits
XNOR-net[63]	−18.1	−16.0	1	1
Bi-real net[65]	−12.9	−9.7	2	2
ABC-net[66]	−4.3	−3.3	5	5
SYQ[85]	−6.2	−4.4	1	8
	−1.4	−1.2	2	8
TTQ[68]	−3.0	−2.0	2	32
SLQ[73]	+0.82	+0.46	5	32
INQ[87]	+0.71	+0.41	5	32
Ref.[89]	−4.3	−2.8	1	32
	−2.1	−1.5	2	32
	−1.6	−1.1	3{±2}	32
	−1.1	−0.7	3{±2 ±4}	32

表 12 展示了一些有代表性的知识蒸馏方法在 MNIST、CIFAR-10、CIFAR-100 和 ImageNet 数据集上的压缩效果.由于使用的教师模型和学生模型的网络结构不同,所以我们将两个模型的 *accuracy* 和参数数量都展示出来,以方便读者对比.可以看出,相比其他方法,知识蒸馏的模型 *accuracy* 下降更多,压缩比更小.目前来看,未来知识蒸馏在模型压缩与加速领域还有很大的发展空间.

Table 12 Compression effects of different knowledge distillation methods

表 12 不同知识蒸馏方法的压缩效果

数据集	方法	*T-accuracy*	*S-accuracy*	Δ*accuracy*	*T-#Params*	*S-#Params*	*#Params*↓
MNIST	Fitnets[148]	99.45	99.49	+0.04	361k	30k	12x
CIFAR-10	Ref.[147]	88	85.8	−2.2	35k	70M	−/
	Fitnets[148]	90.18	91.61	+1.43	9M	2.5M	3.6x
	Ref.[151]	95.21	94.08	−1.13	2243.5k	542.5k	4.13x
CIFAR-100	Fitnets[148]	63.54	64.96	+1.42	9M	2.5M	3.6x
	Ref.[151]	76.15	72.92	−3.23	2243.5k	641.3k	3.5x
ImageNet	Ref.[151]	73.27/91.43	73.39/91.38	+0.12/-0.05	21.8M	8.1M	2.7x

结论:我们总结的 7 类压缩与加速方法各有利弊,由于实验使用的硬件平台不同,并不能量化地确定孰优孰劣.依据不同的应用场景和现实需要,可以进行方法的选取.例如:对于存储有限的嵌入式设备,可以使用非结构

化剪枝或者二值、三值量化,以大幅度地减少模型占用的内存大小.对于没有预训练模型的情况,可以考虑紧凑网络法,直接训练网络.对于期望较高压缩比与加速比的应用场景,可以使用混合方式,组合使用几种压缩与加速方法.

4 未来研究方向

截止到目前,深度学习模型压缩与加速技术尚未发展成熟,在实际部署和产品化水平上还有很大的进步空间.下面介绍几个值得关注与讨论的研究方向.

(1) 知识蒸馏作为一种迁移学习的形式,可使小模型尽可能多地学习到大模型的知识,具有方法灵活、不依赖硬件平台的特点,但目前,其压缩比和蒸馏后性能都有待提高.未来知识蒸馏可从以下几个方向展开研究:打破 softmax 函数的限制,结合中间特征层,使用不同形式的知识;在选择学生模型的结构时,可以与其他方法集成;打破任务的限制,例如将图片分类领域的知识迁移到其他领域;

(2) 将模型压缩技术与硬件架构设计相结合.目前的压缩与加速方法大多仅从软件层面对模型进行优化,并且不同方法由于使用的硬件平台不同,也很难比较其加速效果的好坏.未来可针对主流的压缩与加速方法专门设计硬件架构,既能在现有基础上加速模型,又方便不同方法的比较;

(3) 制定更智能的模型结构选择策略.目前,无论是参数剪枝方法还是设计更紧凑的网络结构,都是基于现有模型作为主干网络,手动选择或使用启发式策略进行结构缩减,缩小了模型搜索空间.未来可以利用强化学习等策略进行自动网络结构搜索,得到更优的网络结构;

(4) 将模型压缩技术推广到更多任务和更多平台.目前的压缩与加速方法多是为图片分类任务的卷积神经网络模型设计,然而实际应用中,还有大量其他模型应用于人工智能领域,例如语音识别和机器翻译领域常使用的递归神经网络(RNN)、知识图谱领域的图神经网络(GNN).为卷积神经网络模型设计的压缩与加速方法能否直接用于RNN与GNN,还需要探索.同时,小型移动平台(如智能手机、机器人、无人驾驶汽车等)的硬件限制及其有限的计算资源阻碍了深度学习模型的直接部署,如何为这些平台设计独有的压缩方法,仍是一个巨大的挑战;

(5) 模型压缩后的安全问题.由于当前压缩与加速方法更注重压缩完成后的任务完成度(例如分类任务 *accuracy* 是否有降低),而忽略了模型压缩可能带来的安全隐患,例如:相比原模型,是否更容易被对抗样本攻击.所以,未来在提升模型性能的同时,也应注意模型是否安全.

5 总 结

本文首先介绍深度学习模型压缩与加速技术的研究背景;其次,对目前主流的方法进行分析,从参数剪枝、参数量化、紧凑网络、知识蒸馏、低秩分解、参数共享、混合方式这 7 个方面进行分类总结;然后,对各类压缩技术中的一些代表性方法进行压缩效果比较;最后,探讨了模型压缩领域未来的发展方向.希望本文能够给研究者带来关于模型压缩与加速技术的更多了解,促进和推动模型压缩与加速技术的未来发展.

References:

[1] LeCun Y. Generalization and network design strategies. Connectionism in Perspective, 1989,19:143−155.

[2] Deng J, Dong W, Socher R, *et al*. Imagenet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2009. 248−255.

[3] Russakovsky O, Deng J, Su H, *et al*. Imagenet large scale visual recognition challenge. Int'l Journal of Computer Vision, 2015, 115(3):211−252.

[4] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. 2012. 1097−1105.

[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv Preprint arXiv: 1409.1556, 2014.

[6] He K, Zhang X, Ren S, *et al*. Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2016. 770−778.

[7] Kim YD, Park E, Yoo S, *et al*. Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv Preprint arXiv: 1511.06530, 2015.

[8] Lei J, Gao X, Song J, Wang XL, Song ML. Survey of deep neural network model compression. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):251−266 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5428.htm [doi: 10.13328/j.cnki.jos.005428]

[9] Ji RZ, Lin SH, Chao F, Wu YJ, Huang FY. Deep neural network compression and acceleration. Computer Research and Development, 2018,55(9):1871−1888 (in Chinese with English abstract). http://crad.ict.ac.cn/CN/10.7544/issn1000-1239.2018.20180129 [doi: 10.7544/issn1000-1239.2018.20180129]

[10] Cao WL, Rui JW, Li M. Survey of neural network model compression methods. Computer Application Research, 2019,36(3):649−656 (in Chinese with English abstract). http://www.arocmag.com/article/01-2019-03-002.html [doi: 10.19734/j.issn.1001-3695.2018.01.0061]

[11] Cheng Y, Wang D, Zhou P, *et al*. A survey of model compression and acceleration for deep neural networks. arXiv Preprint arXiv: 1710.09282, 2017.

[12] Cheng J, Wang P, Li G, *et al*. Recent advances in efficient computation of deep convolutional neural networks. Frontiers of Information Technology & Electronic Engineering, 2018,19(1):64−77.

[13] Chen C, Tung F, Vedula N, *et al*. Constraint-aware deep neural network compression. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 400−415.

[14] Yang H, Zhu Y, Liu J. Energy-constrained compression for deep neural networks via weighted sparse projection and layer input masking. arXiv Preprint arXiv: 1806.04321, 2018.

[15] Yang TJ, Chen YH, Sze V. Designing energy-efficient convolutional neural networks using energy-aware pruning. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 5687−5695.

[16] Yang H, Zhu Y, Liu J. Ecc: Platform-independent energy-constrained deep neural network compression via a bilinear regression model. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2019. 11206−11215.

[17] Hu H, Peng R, Tai YW, *et al*. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. arXiv Preprint arXiv: 1607.03250, 2016.

[18] Khoram S, Li J. Adaptive quantization of neural networks. In: Proc. of the ICLR 2018. 2018.

[19] LeCun Y, Denker JS, Solla SA. Optimal brain damage. In: Advances in Neural Information Processing Systems. 1990. 598−605.

[20] Hassibi B, Stork DG. Second order derivatives for network pruning: Optimal brain surgeon. In: Advances in Neural Information Processing Systems. 1993. 164−171.

[21] Srinivas S, Babu RV. Data-free parameter pruning for deep neural networks. arXiv Preprint arXiv: 1507.06149, 2015.

[22] Dong X, Chen S, Pan S. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In: Advances in Neural Information Processing Systems. 2017. 4857−4867.

[23] Han S, Pool J, Tran J, *et al*. Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems. 2015. 1135−1143.

[24] Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient DNNs. In: Advances in Neural Information Processing Systems. 2016. 1379−1387.

[25] Lin C, Zhong Z, Wei W, *et al*. Synaptic strength for convolutional neural network. In: Advances in Neural Information Processing Systems. 2018. 10149−10158.

[26] Lee N, Ajanthan T, Torr PHS. Snip: Single-shot network pruning based on connection sensitivity. arXiv Preprint arXiv: 1810.02340, 2018.

[27] Macchi O. The coincidence approach to stochastic point processes. Advances in Applied Probability, 1975,7(1):83−122.

[28] Mariet Z, Sra S. Diversity networks: Neural network compression using determinantal point processes. arXiv Preprint arXiv: 1511.05077, 2015.

[29] Kingma DP, Salimans T, Welling M. Variational dropout and the local reparameterization trick. In: Advances in Neural Information Processing Systems. 2015. 2575−2583.

[30] Molchanov D, Ashukha A, Vetrov D. Variational dropout sparsifies deep neural networks. In: Proc. of the 34th Int'l Conf. on Machine Learning, Vol.70. JMLR.org, 2017. 2498−2507.

[31] Louizos C, Welling M, Kingma DP. Learning sparse neural networks through $ L_0 $ regularization. arXiv Preprint arXiv: 1712.01312, 2017.

[32] Tartaglione E, Lepsøy S, Fiandrotti A, *et al*. Learning sparse neural networks via sensitivity-driven regularization. In: Advances in Neural Information Processing Systems. 2018. 3878−3888.

[33] Carreira-Perpinán MA, Idelbayev Y. "Learning-Compression" algorithms for neural net pruning. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 8532−8541.

[34] Liu Z, Xu J, Peng X, *et al*. Frequency-domain dynamic pruning for convolutional neural networks. In: Advances in Neural Information Processing Systems. 2018. 1043−1053.

[35] Wen W, Wu C, Wang Y, *et al*. Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems. 2016. 2074−2082.

[36] Alvarez JM, Salzmann M. Learning the number of neurons in deep networks. In: Advances in Neural Information Processing Systems. 2016. 2270−2278.

[37] Figurnov M, Ibraimova A, Vetrov DP, *et al*. Perforatedcnns: Acceleration through elimination of redundant convolutions. In: Advances in Neural Information Processing Systems. 2016. 947−955.

[38] Lebedev V, Lempitsky V. Fast convnets using group-wise brain damage. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2016. 2554−2564.

[39] Zhou H, Alvarez JM, Porikli F. Less is more: Towards compact cnns. In: Proc. of the European Conf. on Computer Vision. Cham: Springer-Verlag, 2016. 662−677.

[40] Li H, Kadav A, Durdanovic I, *et al*. Pruning filters for efficient convnets. arXiv Preprint arXiv: 1608.08710, 2016.

[41] Chen YH, Emer J, Sze V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. ACM SIGARCH Computer Architecture News, 2016,44(3):367−379.

[42] Yang TJ, Howard A, Chen B, *et al*. Netadapt: Platform-aware neural network adaptation for mobile applications. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 285−300.

[43] He Y, Kang G, Dong X, *et al*. Soft filter pruning for accelerating deep convolutional neural networks. arXiv Preprint arXiv: 1808.06866, 2018.

[44] Liu Z, Li J, Shen Z, *et al*. Learning efficient convolutional networks through network slimming. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 2736−2744.

[45] Huang Z, Wang N. Data-driven sparse structure selection for deep neural networks. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 304−320.

[46] Ye J, Lu X, Lin Z, *et al*. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. arXiv Preprint arXiv: 1802.00124, 2018.

[47] Dai B, Zhu C, Wipf D. Compressing neural networks using the variational information bottleneck. arXiv Preprint arXiv: 1802.10399, 2018.

[48] He Y, Lin J, Liu Z, *et al*. AMC: Automl for model compression and acceleration on mobile devices. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 784−800.

[49] Luo JH, Wu J, Lin W. Thinet: A filter level pruning method for deep neural network compression. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 5058−5066.

[50] He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 1389−1397.

[51] Yu R, Li A, Chen CF, *et al*. Nisp: Pruning networks using neuron importance score propagation. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 9194−9203.

[52] Zhuang Z, Tan M, Zhuang B, *et al*. Discrimination-aware channel pruning for deep neural networks. In: Advances in Neural Information Processing Systems. 2018. 875−886.

[53] Molchanov P, Tyree S, Karras T, *et al*. Pruning convolutional neural networks for resource efficient transfer learning. arXiv Preprint arXiv: 1611.06440, 2016.

[54] Lin S, Ji R, Li Y, *et al*. Accelerating convolutional networks via global & dynamic filter pruning. In: Proc. of the IJCAI. 2018. 2425−2432.

[55] Zhang T, Ye S, Zhang K, *et al*. A systematic DNN weight pruning framework using alternating direction method of multipliers. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 184−199.

[56] Courbariaux M, Bengio Y, David JP. Binaryconnect: Training deep neural networks with binary weights during propagations. In: Advances in Neural Information Processing Systems. 2015. 3123−3131.

[57] Hou L, Yao Q, Kwok JT. Loss-aware binarization of deep networks. arXiv Preprint arXiv: 1611.01600, 2016.

[58] Juefei-Xu F, Naresh Boddeti V, Savvides M. Local binary convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 19−28.

[59] Guo Y, Yao A, Zhao H, *et al*. Network sketching: Exploiting binary structure in deep CNNs. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 5955−5963.

[60] McDonnell MD. Training wide residual networks for deployment using a single bit for each weight. arXiv Preprint arXiv: 1802.08530, 2018.

[61] Hu Q, Wang P, Cheng J. From hashing to CNNs: Training binary weight networks via hashing. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[62] Courbariaux M, Hubara I, Soudry D, *et al*. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or −1. arXiv Preprint arXiv: 1602.02830, 2016.

[63] Rastegari M, Ordonez V, Redmon J, *et al*. Xnor-net: Imagenet classification using binary convolutional neural networks. In: Proc. of the European Conf. on Computer Vision. Cham: Springer-Verlag, 2016. 525−542.

[64] Li Z, Ni B, Zhang W, *et al*. Performance guaranteed network acceleration via high-order residual quantization. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 2584−2592.

[65] Liu Z, Wu B, Luo W, *et al*. Bi-real net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 722−737.

[66] Lin X, Zhao C, Pan W. Towards accurate binary convolutional neural network. In: Advances in Neural Information Processing Systems. 2017. 345−353.

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

[68] Zhu C, Han S, Mao H, *et al*. Trained ternary quantization. arXiv Preprint arXiv: 1612.01064, 2016.

[69] Achterhold J, Koehler J M, Schmeink A, *et al*. Variational network quantization. In: Proc. of the ICLR 2017. 2017.

[70] Gong Y, Liu L, Yang M, *et al*. Compressing deep convolutional networks using vector quantization. arXiv Preprint arXiv: 1412.6115, 2014.

[71] Wu J, Leng C, Wang Y, *et al*. Quantized convolutional neural networks for mobile devices. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2016. 4820−4828.

[72] Choi Y, El-Khamy M, Lee J. Towards the limit of network quantization. arXiv Preprint arXiv: 1612.01543, 2016.

[73] Xu Y, Wang Y, Zhou A, *et al*. Deep neural network compression with single and multiple level quantization. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[74] Lin Z, Courbariaux M, Memisevic R, *et al*. Neural networks with few multiplications. arXiv Preprint arXiv: 1510.03009, 2015.

[75] Zhou S, Wu Y, Ni Z, *et al*. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv Preprint arXiv: 1606.06160, 2016.

[76] Mishra A, Nurvitadhi E, Cook JJ, *et al*. WRPN: Wide reduced-precision networks. arXiv Preprint arXiv: 1709.01134, 2017.

[77] Köster U, Webb T, Wang X, *et al*. Flexpoint: An adaptive numerical format for efficient training of deep neural networks. In: Advances in Neural Information Processing Systems. 2017. 1742−1752.

[78] Wang N, Choi J, Brand D, *et al*. Training deep neural networks with 8-bit floating point numbers. In: Advances in Neural Information Processing Systems. 2018. 7675−7684.

[79] Gupta S, Agrawal A, Gopalakrishnan K, *et al*. Deep learning with limited numerical precision. In: Proc. of the Int'l Conf. on Machine Learning. 2015. 1737−1746.

[80] Dettmers T. 8-bit approximations for parallelism in deep learning. arXiv Preprint arXiv: 1511.04561, 2015.

[81] Jacob B, Kligys S, Chen B, *et al*. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 2704−2713.

[82] Banner R, Hubara I, Hoffer E, *et al*. Scalable methods for 8-bit training of neural networks. In: Advances in Neural Information Processing Systems. 2018. 5145−5153.

[83] Wu S, Li G, Chen F, *et al*. Training and inference with integers in deep neural networks. arXiv Preprint arXiv: 1802.04680, 2018.

[84] Wang P, Hu Q, Zhang Y, *et al*. Two-step quantization for low-bit neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 4376−4384.

[85] Faraone J, Fraser N, Blott M, *et al*. SYQ: Learning symmetric quantization for efficient deep neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 4300−4309.

[86] Zhang D, Yang J, Ye D, *et al*. LQ-nets: Learned quantization for highly accurate and compact deep neural networks. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 365−382.

[87] Zhou A, Yao A, Guo Y, *et al*. Incremental network quantization: Towards lossless CNNs with low-precision weights. arXiv Preprint arXiv: 1702.03044, 2017.

[88] Cai Z, He X, Sun J, *et al*. Deep learning with low precision by half-wave Gaussian quantization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 5918−5926.

[89] Leng C, Dou Z, Li H, *et al*. Extremely low bit neural network: Squeeze the last bit out with ADMM. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[90] Boyd S, Parikh N, Chu E, *et al*. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, 2011,3(1):1−122.

[91] Zhuang B, Shen C, Tan M, *et al*. Towards effective low-bitwidth convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 7920−7928.

[92] Zhou A, Yao A, Wang K, *et al*. Explicit loss-error-aware quantization for low-bit deep neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 9426−9435.

[93] Park E, Yoo S, Vajda P. Value-Aware quantization for training and inference of neural networks. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 580−595.

[94] Shayer O, Levi D, Fetaya E. Learning discrete weights using the local reparameterization trick. arXiv Preprint arXiv: 1710.07739, 2017.

[95] Louizos C, Reisser M, Blankevoort T, *et al*. Relaxed quantization for discretized neural networks. arXiv Preprint arXiv: 1810.01875, 2018.

[96] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions. arXiv Preprint arXiv: 1405.3866, 2014.

[97] Liu B, Wang M, Foroosh H, *et al*. Sparse convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2015. 806−814.

[98] Tai C, Xiao T, Zhang Y, *et al*. Convolutional neural networks with low-rank regularization. arXiv Preprint arXiv: 1511.06067, 2015.

[99] Masana M, van de Weijer J, Herranz L, *et al*. Domain-adaptive deep network compression. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 4289−4297.

[100] Wen W, Xu C, Wu C, *et al*. Coordinating filters for faster deep neural networks. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 658−666.

[101] Wang P, Cheng J. Fixed-Point factorized networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 4012−4020.

[102] Peng B, Tan W, Li Z, *et al*. Extreme network compression via filter group approximation. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 300–316.

[103] Qiu Q, Cheng X, Calderbank R, *et al*. DCFnet: Deep neural network with decomposed convolutional filters. arXiv Preprint arXiv: 1802.04145, 2018.

[104] Novikov A, Podoprikhin D, Osokin A, *et al*. Tensorizing neural networks. In: Advances in Neural Information Processing Systems. 2015. 442–450.

[105] Garipov T, Podoprikhin D, Novikov A, *et al*. Ultimate tensorization: compressing convolutional and fc layers alike. arXiv Preprint arXiv: 1611.03214, 2016.

[106] Wang W, Sun Y, Eriksson B, *et al*. Wide compression: Tensor ring nets. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 9329–9338.

[107] Kim YD, Park E, Yoo S, *et al*. Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv Preprint arXiv: 1511.06530, 2015.

[108] Wang P, Cheng J. Accelerating convolutional neural networks for mobile applications. In: Proc. of the 24th ACM Int'l Conf. on Multimedia. 2016. 541–545.

[109] Lebedev V, Ganin Y, Rakhuba M, *et al*. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv Preprint arXiv: 1412.6553, 2014.

[110] Cheng Y, Yu FX, Feris RS, *et al*. An exploration of parameter redundancy in deep networks with circulant projections. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2015. 2857–2865.

[111] Wang Y, Xu C, Xu C, *et al*. Beyond filters: Compact feature map for portable deep model. In: Proc. of the 34th Int'l Conf. on Machine Learning, Vol.70. JMLR.org, 2017. 3703–3711.

[112] Sindhwani V, Sainath T, Kumar S. Structured transforms for small-footprint deep learning. In: Advances in Neural Information Processing Systems. 2015. 3088–3096.

[113] Zhao L, Liao S, Wang Y, *et al*. Theoretical properties for neural networks with weight matrices of low displacement rank. In: Proc. of the 34th Int'l Conf. on Machine Learning, Vol.70. JMLR.org, 2017. 4082–4090.

[114] Le Q, Sarlós T, Smola A. Fastfood-approximating kernel expansions in loglinear time. In: Proc. of the Int'l Conf. on Machine Learning. 2013. 85.

[115] Yang Z, Moczulski M, Denil M, *et al*. Deep fried convnets. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2015. 1476–1483.

[116] Chen W, Wilson J, Tyree S, *et al*. Compressing neural networks with the hashing trick. In: Proc. of the Int'l Conf. on Machine Learning. 2015. 2285–2294.

[117] Chen W, Wilson J, Tyree S, *et al*. Compressing convolutional neural networks in the frequency domain. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2016. 1475–1484.

[118] Wu J, Wang Y, Wu Z, *et al*. Deep $ *k* $-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions. arXiv Preprint arXiv: 1806.09228, 2018.

[119] Son S, Nah S, Mu Lee K. Clustering convolutional kernels to compress deep neural networks. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 216–232.

[120] Reagen B, Gupta U, Adolf R, *et al*. Weightless: Lossy weight encoding for deep neural network compression. arXiv Preprint arXiv: 1711.04686, 2017.

[121] Havasi M, Peharz R, Hernández-Lobato JM. Minimal random code learning: Getting bits back from compressed model parameters. arXiv Preprint arXiv: 1810.00440, 2018.

[122] Jin X, Yang Y, Xu N, *et al*. Wsnet: Compact and efficient networks through weight sampling. arXiv Preprint arXiv: 1711.10067, 2017.

[123] Kossaifi J, Bulat A, Tzimiropoulos G, *et al*. T-net: Parametrizing fully convolutional nets with a single high-order tensor. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2019. 7822–7831.

[124] Iandola FN, Han S, Moskewicz MW, *et al*. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv Preprint arXiv: 1602.07360, 2016.

[125] Howard AG, Zhu M, Chen B, *et al*. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv Preprint arXiv: 1704.04861, 2017.

[126] Sandler M, Howard A, Zhu M, *et al*. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 4510−4520.

[127] Zhang X, Zhou X, Lin M, *et al*. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 6848−6856.

[128] Ma N, Zhang X, Zheng HT, *et al*. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 116−131.

[129] Zhang T, Qi GJ, Xiao B, *et al*. Interleaved group convolutions. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2017. 4373−4382.

[130] Xie G, Wang J, Zhang T, *et al*. Interleaved structured sparse convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 8847−8856.

[131] Wang X, Kan M, Shan S, *et al*. Fully learnable group convolution for acceleration of deep neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2019. 9049−9058.

[132] Park J, Li S, Wen W, *et al*. Faster CNNs with direct sparse convolutions and guided pruning. arXiv Preprint arXiv: 1608.01409, 2016.

[133] Zhang J, Franchetti F, Low TM. High performance zero-memory overhead direct convolutions. arXiv Preprint arXiv: 1809.10170, 2018.

[134] Ioannou Y, Robertson D, Shotton J, *et al*. Training cnns with low-rank filters for efficient image classification. arXiv Preprint arXiv: 1511.06744, 2015.

[135] Bagherinezhad H, Rastegari M, Farhadi A. LCNN: Lookup-based convolutional neural network. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 7120−7129.

[136] Wang Y, Xu C, Chunjing XU, *et al*. Learning versatile filters for efficient convolutional neural networks. In: Advances in Neural Information Processing Systems. 2018. 1608−1618.

[137] Huang G, Sun Y, Liu Z, *et al*. Deep networks with stochastic depth. In: Proc. of the European Conf. on Computer Vision. Cham: Springer-Verlag, 2016. 646−661.

[138] Dong X, Huang J, Yang Y, *et al*. More is less: A more complicated network with less inference complexity. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 5840−5848.

[139] Li D, Wang X, Kong D. Deeprebirth: Accelerating deep neural network execution on mobile devices. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[140] Prabhu A, Varma G, Namboodiri A. Deep expander networks: Efficient deep networks from graph theory. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 20−35.

[141] Wu B, Wan A, Yue X, *et al*. Shift: A zero flop, zero parameter alternative to spatial convolutions. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 9127−9135.

[142] Chen W, Xie D, Zhang Y, *et al*. All you need is a few shifts: Designing efficient convolutional neural networks for image classification. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2019. 7241−7250.

[143] Kim J, Park Y, Kim G, *et al*. SplitNet: Learning to semantically split deep networks for parameter reduction and model parallelization. In: Proc. of the 34th Int'l Conf. on Machine Learning, Vol.70. JMLR.org, 2017. 1866−1874.

[144] Gordon A, Eban E, Nachum O, *et al*. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 1586−1595.

[145] Kim E, Ahn C, Oh S. Nestednet: Learning nested sparse structures in deep neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 8669−8678.

[146] Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2006. 535−541.

[147] Ba J, Caruana R. Do deep nets really need to be deep? In: Advances in Neural Information Processing Systems. 2014. 2654−2662.

[148] Romero A, Ballas N, Kahou SE, *et al*. Fitnets: Hints for thin deep nets. arXiv Preprint arXiv: 1412.6550, 2014.

[149] Chen T, Goodfellow I, Shlens J. Net2net: Accelerating learning via knowledge transfer. arXiv Preprint arXiv: 1511.05641, 2015.

[150] Li Z, Hoiem D. Learning without forgetting. IEEE Trans. on Pattern Analysis And Machine Intelligence, 2017,40(12):2935−2947.

[151] Crowley EJ, Gray G, Storkey AJ. Moonshine: Distilling with cheap convolutions. In: Advances in Neural Information Processing Systems. 2018. 2888−2898.

[152] Zhu X, Gong S. Knowledge distillation by on-the-fly native ensemble. In: Advances in Neural Information Processing Systems. 2018. 7517−7527.

[153] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv Preprint arXiv: 1503.02531, 2015.

[154] Yim J, Joo D, Bae J, *et al*. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 4133−4141.

[155] Chen Y, Wang N, Zhang Z. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[156] Czarnecki WM, Osindero S, Jaderberg M, *et al*. Sobolev training for neural networks. In: Advances in Neural Information Processing Systems. 2017. 4278−4287.

[157] Lopes RG, Fenu S, Starner T. Data-free knowledge distillation for deep neural networks. arXiv Preprint arXiv: 1710.07535, 2017.

[158] Zhou G, Fan Y, Cui R, *et al*. Rocket launching: A universal and efficient framework for training well-performing light net. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[159] Li Q, Jin S, Yan J. Mimicking very efficient network for object detection. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 6356−6364.

[160] Chen G, Choi W, Yu X, *et al*. Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. 2017. 742−751.

[161] Luo P, Zhu Z, Liu Z, *et al*. Face model compression by distilling knowledge from neurons. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence. 2016.

[162] Gupta S, Hoffman J, Malik J. Cross modal distillation for supervision transfer. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2016. 2827−2836.

[163] Xu D, Ouyang W, Wang X, *et al*. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 675−684.

[164] Hu Q, Li G, Wang P, *et al*. Training binary weight networks via semi-binary decomposition. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 637−653.

[165] Ullrich K, Meeds E, Welling M. Soft weight-sharing for neural network compression. arXiv Preprint arXiv: 1702.04008, 2017.

[166] Tung F, Mori G. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2018. 7873−7882.

[167] Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv Preprint arXiv: 1510.00149, 2015.

[168] Han S, Liu X, Mao H, *et al*. EIE: Efficient inference engine on compressed deep neural network. ACM SIGARCH Computer Architecture News, 2016,44(3):243−254.

[169] Dubey A, Chatterjee M, Ahuja N. Coreset-based neural network compression. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 454−470.

[170] Louizos C, Ullrich K, Welling M. Bayesian compression for deep learning. In: Advances in Neural Information Processing Systems. 2017. 3288−3298.

[171] Ji Y, Liang L, Deng L, *et al*. TETRIS: Tile-matching the tremendous irregular sparsity. In: Advances in Neural Information Processing Systems. 2018. 4115−4125.

[172] Zhang D, Wang H, Figueiredo M, *et al*. Learning to share: Simultaneous parameter tying and sparsification in deep learning. In: Proc. of the 6th Int'l Conf. on Learning Representations. 2018.

[173] Polino A, Pascanu R, Alistarh D. Model compression via distillation and quantization. arXiv Preprint arXiv: 1802.05668, 2018.

[174] Mishra A, Marr D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. arXiv Preprint arXiv: 1711.05852, 2017.

[175] LeCun Y, Bottou L, Bengio Y, *et al*. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998,86(11): 2278−2324.

[176] Krizhevsky A. Learning multiple layers of features from tiny images [MS. Thesis]. University of Toronto, 2009.

[177] Deng J, Dong W, Socher R, *et al*. Imagenet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2009. 248−255.

附中文参考文献:

[8] 雷杰,高鑫,宋杰,王兴路,宋明黎.深度网络模型压缩综述.软件学报,2018,29(2):251−266. http://www.jos.org.cn/1000-9825/5428.htm [doi: 10.13328/j.cnki.jos.005428]

[9] 纪荣嵘,林绍辉,晁飞,吴永坚,黄飞跃.深度神经网络压缩与加速综述.计算机研究与发展,2018,55(9):1871−1888. http://crad.ict.ac.cn/CN/10.7544/issn1000-1239.2018.20180129 [doi: 10.7544/issn1000-1239.2018.20180129]

[10] 曹文龙,芮建武,李敏.神经网络模型压缩方法综述.计算机应用研究,2019,36(3):649−656. http://www.arocmag.com/article/01-2019-03-002.html [doi: 10.19734/j.issn.1001-3695.2018.01.0061]

高晗(1997－),女,学士,主要研究领域为深度学习及其安全挑战.

田育龙(1995－),男,学士,主要研究领域为深度学习安全,系统安全.

许封元(1983－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为深度学习安全,边缘计算系统,用于受信任执行环境的软件.

仲盛(1974－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为密码学,博弈论及其在计算机网络,分布式系统中的应用.