

文章编号: 1000-5641(2020)05-0068-15

深度神经网络模型压缩方法与进展

赖叶静, 郝珊锋, 黄定江

(华东师范大学 数据科学与工程学院, 上海 200062)

摘要: 深度神经网络 (Deep Neural Network, DNN) 模型通过巨大的内存消耗和高计算量来实现强大的性能, 难以部署在有限资源的硬件平台上. 通过模型压缩来降低内存成本和加速计算已成为热点问题, 近年来已有大量的这方面的研究工作. 主要介绍了 4 种具有代表性的深度神经网络压缩方法, 即网络剪枝、量化、知识蒸馏和紧凑神经网络设计; 着重介绍了近年来具有代表性的压缩模型方法及其特点; 最后, 总结了模型压缩的相关评价标准和研究前景.

关键词: 深度神经网络压缩; 网络剪枝; 量化; 知识蒸馏; 紧凑神经网络

中图分类号: TP391 **文献标志码:** A **DOI:** [10.3969/j.issn.1000-5641.202091001](https://doi.org/10.3969/j.issn.1000-5641.202091001)

Methods and progress in deep neural network model compression

LAI Yejing, HAO Shanfeng, HUANG Dingjiang

(School of Data Science and Engineering, East China Normal University, Shanghai 200062, China)

Abstract: The deep neural network (DNN) model achieves strong performance using substantial memory consumption and high computational power, which can be difficult to deploy on hardware platforms with limited resources. To meet these challenges, researchers have made great strides in this field and have formed a wealth of relevant literature and methods. This paper introduces four representative compression methods for deep neural networks used in recent years: network pruning, quantization, knowledge distillation, and compact network design; in particular, the article focuses on the characteristics of these representative models. Finally, evaluation criteria and research prospects of model compression are summarized.

Keywords: deep neural network compression; network pruning; quantification; knowledge distillation; compact neural network

0 引 言

近年来, 深度神经网络 (DNN) 在许多领域取得了巨大的成功, 包括但不限于目标识别和检测^[1]、语音识别^[2]、自然语言处理^[3]. 这些成功依赖于更多的参数、更大更深的模型. 例如, 在 VGG-16 (Visual Geometry Group-16) 网络上训练 ImageNet 数据集得到的模型大小超过 500 MB, 参数数量高达 138 357 544 个. 自 2012 年 AlexNet 出现并拿下当年 ImageNet 竞赛的冠军后, 深度神经网络在计算机视觉领域大放异彩, 随后出现的卷积神经网络模型数量大幅度增加, 比如 VGG^[4]、ResNet^[5]、DenseNet^[6] 等. 这些深度网络模型在许多领域的实验中表现良好, 但在实际应用中仍然受到时间和空间的限制.

收稿日期: 2020-08-02

基金项目: 国家自然科学基金 (11501204, U1711262)

通信作者: 黄定江, 男, 教授, 博士生导师, 研究方向为机器学习与人工智能. E-mail: djhuang@dase.ecnu.edu.cn

即使使用图形处理单元 (Graphics Processing Unit, GPU) 或张量处理单元 (Tensor Processing Unit, TPU) 进行加速, 这些宽而深的网络模型仍然不能满足在许多应用场景中的实时需求. 与此同时, 手机和边缘设备等资源受限设备的数量每年都在增加, 体积大、计算成本高的模型会消耗大量的计算资源, 不适用于手机等移动设备. 因此, 在不影响深度网络模型准确度的前提下, 模型压缩是一个重要的研究问题.

实现模型压缩的方法有很多. 本文将这些方法分为 4 类: 网络剪枝、量化、知识蒸馏和紧凑神经网络设计. 网络剪枝主要通过设计一个标准去判断参数的重要程度, 再根据重要性去除冗余参数. 量化减少表示每个权值和激活值所需的比特数, 如二值化 (1-bit)、int8 量化 (8-bit) 等. 知识蒸馏主要利用大型网络的知识, 并将其知识转化到紧凑小型的学生模型中. 紧凑神经网络通过设计一个特殊结构的卷积核或紧凑卷积的计算单元, 来降低模型的存储和计算复杂度. 表 1 简要总结了 4 类神经网络压缩方法的优缺点和适用场景.

表 1 神经网络压缩方法概要
Tab. 1 Summary of neural network compression methods

压缩方法	描述	优缺点	适用场景
剪枝	移除已训练好模型中冗余的、信息量较少的权重	降低网络复杂度, 解决过拟合问题; 但需要设计专用的计算库, 计算复杂度高	已知预训练模型, 微调时需要原始数据集. 适合计算内存和存储容量低的设备
量化	减少表示一个权值所需比特数	与硬件相结合, 大大提高推理速度; 但精度下降明显	适合实时推理速度较高且计算内存低的场景
知识蒸馏	学生模型学习大型教师模型知识	大大降低计算量和存储量; 主要用于分类任务, 适用范围窄, 且知识定义困难	已知教师预训练模型, 适用于数据集较小或者没有数据集的情况
紧凑神经网络	设计更紧凑的卷积核或卷积方式	通用卷积网络, 网络参数量减少; 特殊卷积核计算较慢	端到端训练压缩模型, 有完整的训练、测试数据集

在模型压缩和加速领域, 涌现出了许多优秀的综述论文: 文献 [7-8] 对模型压缩研究的进展做了详尽的调查分析, 主要介绍了前几年传统的优秀算法, 但错过了近年来许多重要并且有代表性的工作; 文献 [9-10] 将模型压缩分为 6 个方向, 并介绍了近年来提出的算法. 与之前的综述文献不同, 本文将模型压缩分为 4 大类别, 对过去几年相关的研究进展与具有代表性的方法进行了调查, 并给出该领域内评价准则的具体计算方法, 在先进压缩算法上综合比较了图像分类和目标检测两个任务的性能.

本文后续结构: 第 1 章分别介绍 4 种模型压缩方法, 并着重介绍近年来涌现出的模型压缩方法及其特点; 第 2 章介绍模型压缩的相关评价标准, 并对当下具有代表性的压缩方法进行性能评估; 第 3 章对全文进行总结并展望未来的研究热点.

1 模型压缩算法

模型压缩有许多优秀的算法, 并在各个领域有着广泛的应用^[11-14]. 如 Wang 等^[11]提出的 PeleeNet, 通过设计高效的卷积方式并与移动设备硬件的运行库结合, 能够对移动设备上的目标检测、图像分类等任务进行实时预测. 使用 PeleeNet 在 iphone8 上实现目标检测任务, 可达到 23.6 FPS (Frames Per Second, 每秒传输帧数), 且准确率较高. 下面介绍 4 种具有代表性的深度网络压缩方法, 并比较分析它们的优点与不足.

1.1 网络剪枝

深度网络模型中存在许多冗余和信息量较少的权值, 网络剪枝通过去除训练好的模型中冗余的参数, 从而减小模型的体积, 并加快模型的计算速度, 压缩模型的存储空间. 另外, 剪枝也可以降低网络的复杂度, 解决过拟合的问题.

根据剪枝粒度级别的不同,网络剪枝可分为4种剪枝粒度^[15],如图1所示.在最粗粒度的级别中,可以移去一个层(layer),被修剪的层如图1a)中的阴影所示.第二个修剪粒度是特征图/滤波器(feature map/filter),其中,特征图是网络输出,滤波器是网络中的参数.在剪枝的过程中,剪去一个特征图等价于上一层的一个滤波器可以得到一个更薄的网络,图1b)中阴影部分表示被移除的滤波器.下一个修剪粒度是核(kernel),即修剪滤波器中的一个通道,如图1c)所示.最细粒度的是修剪核内的一个权重(weight),如图1d)所示,核中阴影部分的零即是被修剪的权重参数,此剪枝粒度可以产生更稀疏的权重矩阵.这4种剪枝粒度可进一步分为结构化剪枝和非结构化剪枝两类.层间剪枝、特征图剪枝和核剪枝是结构化剪枝,核内权重修剪(核内剪枝)是非结构化修剪.

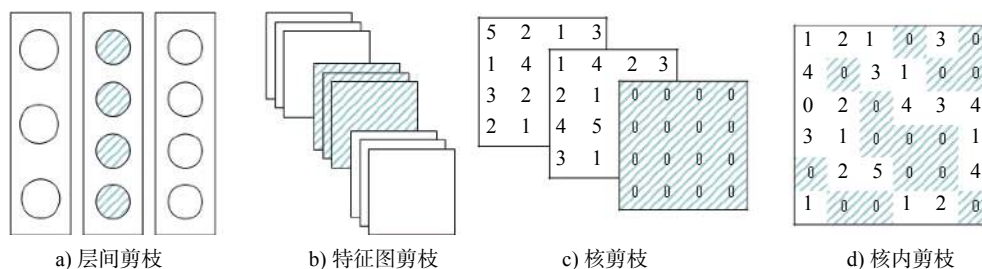


图1 4种剪枝粒度

Fig. 1 Four pruning granularities

早期的网络剪枝方法多是非结构化剪枝. LeCun 等^[16]和 Hassibi 等^[17]分别在1990年和1993年提出了最优化脑损失 (Optimal Brain Damage, OBD) 和最优化脑手术 (Optimal Brain Surgeon, OBS) 方法, 后者是前者的改进, 其基本思想是使用损失函数相对于权重的 Hessian 矩阵来度量网络中核内权重的重要性, 从而删除不重要权重. 这两个算法能在一定范围内提升准确度, 但时间代价较高, 因此, 不能在大型网络上应用. Zhang 等^[18]提出了一种基于交替方向乘子法 (Alternating Direction Method of Multipliers, ADMM) 的权值剪枝系统: 首先将深度神经网络的权值剪枝问题转化为具有指定稀疏性要求的组合约束的非凸优化问题; 然后利用 ADMM 将非凸优化问题分解为两个迭代求解的子问题, 一个用随机梯度下降法求解, 另一个用解析法求解. 该系统使得 AlexNet 和 LeNet 在 ImageNet 和 MNIST 数据集上的权重参数获得了大幅度的压缩. 以上非结构化剪枝往往使得层内的权重矩阵变成稀疏矩阵, 但并没有减少计算量, 而且对稀疏矩阵的加速计算需要特定的软件库或者硬件来支持. 针对此问题, Ma 等^[19]提出了 PCONV 方法, 一种粗粒度结构中的细粒度剪枝模式, 包含模式化剪枝和连通性剪枝. 模式化剪枝可以获得 filter 不同的稀疏性, 连通性剪枝进一步对 filter 做核剪枝, 获得 filter 之间的稀疏性. 此外, 利用 PCONV 剪枝特性设计专门的编译器, 在具有代表性的大规模 DNN 上实现了高压缩和较高的推理速度. 除了设计特别的剪枝结构、设计专有的编译器, 结构化剪枝也可以克服非结构化剪枝的缺点.

结构化剪枝的主要的思想是将不重要的信道或者滤波器去除, 并最小化重构误差. He 等^[20]基于 LASSO (Least Absolute Shrinkage and Selection Operator) 回归的方法进行信道选择, 再通过线性最小二乘法重构网络输出. Chin 等^[21]提出了学习卷积网络不同层 filter 的全局排序, 该排序用于获得一组具有不同精度/延迟的卷积网络架构; 对于跨层的 filter 排序, 通过学习分层的参数仿射变换替代以往的范数准则来评判 filter 的重要性. Molchanov 等^[22]使用一阶和二阶泰勒展开式来近似 filter 的贡献, 并逐层去除贡献低的 filter, 这个方法可以应用于任何类型的层, 如残差网络的 shortcut 层.

Luo 等^[23]提出的 ThiNet 框架, filter 是否被剪去取决于下一层, 而不是当前层; 在对第 i 层的 filter 进行剪枝时, 学习如何选择 $i+1$ 层通道数的某个子集输入原网络, 并且逼近原来的输出结果, 此

时第 i 层的 filter 也可以去除, 在对某一层进行剪枝后, 通过最小化重建误差进行微调, 但这个方法忽略了信道的鉴别能力. 因此 Zhuang 等^[24] 提出了一种用于深度神经网络压缩的鉴别力感知通道修剪策略 (Discrimination-aware Channel Pruning, DCP), 将信道剪枝问题作为稀疏性优化问题, 同时考虑重构误差和信道鉴别能力. 对于深层模型, 由于传播路径长, 其浅层往往具有很小的鉴别能力. 为了提高中间层的鉴别能力, DCP 把深层网络分为 $p+1$ 个阶段, 并在这 $p+1$ 阶段中引入额外的鉴别力感知损失函数 $\mathcal{L}_{S,p}(W)$. 考虑到鉴别力损失和重建损失, 该策略关于信道选择的联合损失函数为

$$\mathcal{L}(W) = \mathcal{L}_M(W) + \mathcal{L}_{S,p}(W),$$

其中, $\mathcal{L}_M(W)$ 为重建损失误差. 整个策略分为两个阶段: 第一阶段把预训练模型分为 $p+1$ 个阶段后计算其鉴别力感知损失, 再根据 \mathcal{L}_f 模型损失和 $\mathcal{L}_{S,p}(W)$ 微调权重参数 W 以降低模型的损失. 第二阶段, 对每个阶段 p 使用贪婪算法选择最重要的通道, 并最小化联合损失函数 $\mathcal{L}(W)$. DCP 架构如图 2 所示. 图 2 中 X 、 W 、 O 分别代表剪枝网络的输入、权重、输出, X_b 、 W_b 、 O_b 分别代表基线网络的输入、权重、输出, O_p 代表第 p 层的输出特征, F_p 代表 O_p 经过 BatchNorm-ReLU-AvgPooling 操作后的输出特征.

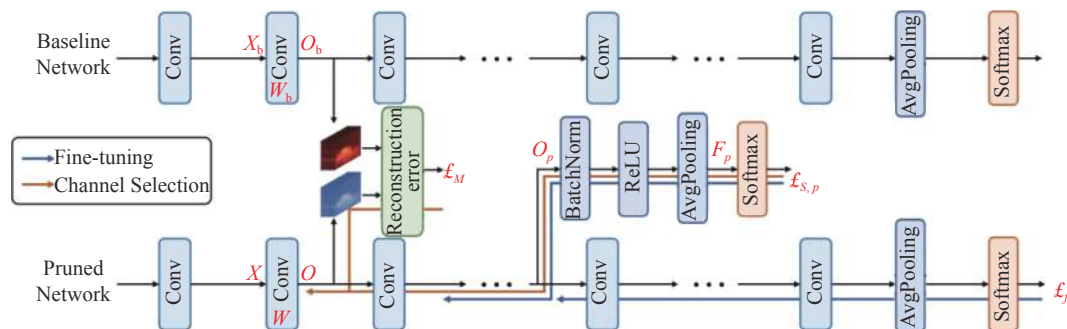


图 2 DCP 框架^[24]

Fig. 2 The DCP framework^[24]

结构化剪枝常常使用范数评价指标来评价某个 filter 是否重要, 用范数评价 filter 时隐含了两个条件: 一是范数的方差要大; 二是最小范数接近零. 使用符合这两个条件的范数作为评价 filter 的准则, 最后的重建误差往往会比较低. 但实际中, filter 范数分布无法同时满足这两个条件, 所以可以通过判断两个 filter 的相似性来判断 filter 是否冗余. 但判断其相似性是比较困难的. 所以, He 等^[25] 利用几何中值 (Geometric Median, GM) 来判断两个 filter 间的相似性. 几何中值是在欧氏空间中点的中心估计, 靠近该中心点的 filter 可以用远离该中心点的 filter 来近似表示, 即可以剪去接近几何中值的 filter. 而 Lin 等人^[26] 发现即使输入图像的批大小 (Batch Size) 不同, 但单个 filter 生成的多个特征图的平均秩总是相等. 高秩的特征图往往包含更多的信息, 所以可以剪去产生低秩特征图的 filter. 这两个方法与范数无关, 不需满足范数评价指标的条件, 但在 CIFAR-10 和 ImageNet 数据集上使用不同的网络进行剪枝, 都可以在几乎不损失准确度的情况下获得更高的压缩率.

剪枝就是去除原网络中不重要的权重, 从另一个角度来看, 被剪枝后的网络结构就是原网络的一个子结构, 这跟神经网络搜索 (Neural Architecture Search, NAS) 很相似, 所以剪枝也可以说是神经网络搜索的一个特例, 它可以使得搜索空间变得更小. 目前许多剪枝方法越来越偏向于神经网络搜索. 剪枝方法和量化方法通常可以结合使用.

1.2 量 化

网络量化通过减少表示每个权值所需的比特数来压缩原始网络^[7]. 通过这种方式, 网络中权值和

激活值都被量化, 并且浮点乘法累加操作 (Multiply Accumulate, MAC) 可以被低比特的乘法累加操作代替, 在二值化网络^[27-29]和三值化网络^[30-31]的情况下甚至不需要乘法. 因此, 使用低比特量化神经网络可以降低存储和计算复杂度. 同时, 低比特量化也有利于面向神经网络芯片硬件的加速, 每降低一比特通常更容易简化硬件的设计, 也可以设计出更精细的芯片^[32]. 量化通过最小化量化误差找到最优量化器, 相应公式为

$$\min J(q_x(x)) = \|x - q_x(x)\|_2^2,$$

其中, x 表示全精度参数, $q_x(x)$ 表示量化后的低比特参数, $J(q_x(x))$ 表示全精度参数与二进制参数之间的量化误差.

量化方法可分为权重共享和低比特表示. 具有代表性的权重共享方法通过聚类来现. Han 等^[33]对每一层的权重矩阵使用 K -Means 聚类算法, 并使用每个簇的质心来表示权重矩阵中的值. 因为相同簇的权值共享一个质心, 所以只需要存储质心的索引, 并通过查找表获得该索引对应的值, 最后通过梯度进行微调就可减少精度损失, 如图 3 所示. 另一种经典的权重共享方法是使用散列函数进行量化. Chen 等^[34]使用哈希技巧将网络连接权值随机分组到散列桶中, 每个散列桶中的网络连接共享相同的权值参数. 这些方法都属于传统的标量量化方法. Stock 等^[35]基于乘积量化 (Product Quantization, PQ) 提出了比特下降的向量量化方法, 区别于以前的向量量化方法^[36], 该方法注重激活值的重要性, 而不是权重的重要性. 其原理是通过最小化域内输入的重建误差, 然后把未压缩的神经网络作为教师网络, 把压缩后的网络作为学生网络, 在量化时只需要一组未标记的数据, 再使用字节对齐的码书来存储压缩后的权重. 通过这种方法可以在 CPU 上进行有效的推理.

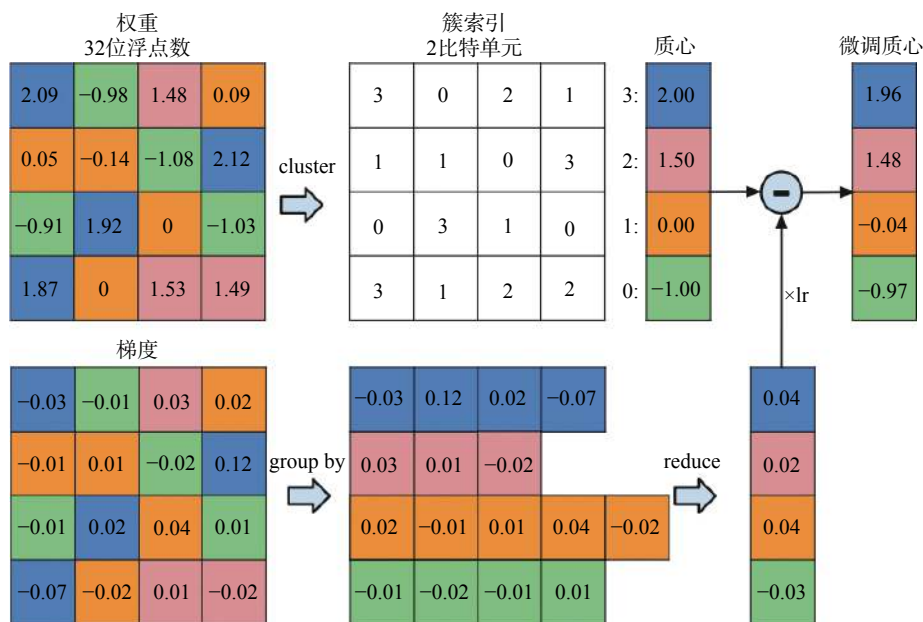


图 3 通过标量量化 (顶部) 和中心体微调 (底部) 共享权重^[33]

Fig. 3 Weight sharing by scalar quantization (top) and fine-tuning of centroids (bottom)^[33]

大量基于低比特量化的方法是基于二值化网络^[27]的改进工作^[37-39]. 二值化方法使用 1 bit 对数据进行量化, 量化后的数据取值只有两个可能的值: 0(-1) 或 +1. 2016 年, Courbariaux 等^[40]提出 BinaryConnect 后, 权重值和激活值分别二值化就成为一种有效的深度神经网络压缩方法. 通过二值化, 可以将繁重的矩阵乘法运算替换为轻量级的异或运算和位计数运算. 二值化的离散性和有限的表示能力, 导致正向和逆向传播中都存在严重的信息损失. 在正向传播中, 当权重值和激活值限制在两

个值时, 模型的多样性会大大下降, 导致了精度大幅度下降. Qin 等^[41] 提出信息保留网络 (Information Retention Network, IR-Net) 来保留前向激活和后向梯度的信息. IR-Net 分为两个过程: 首先通过均衡和标准化正向传播的权重; 同时最小化参数的量化误差和最大化量化参数的信息熵, 在没有对激活值添加额外操作的同时, 减少了权重和激活的信息损失. 二值化常常使用符号函数作为量化函数, 对符号函数进行反向传播会造成巨大的信息损失, 所以在反向传播中需要进行梯度近似. 为了保留反向传播中损失函数的信息, IR-Net 通过逐步逼近反向传播中的符号函数来最小化梯度的信息损失. 由该方法在 CIFAR-10 和 ImageNet 数据集上使用多个深度神经网络进行的实验可知, 其比普通二值化方法获得的精度更高.

除了二值化网络, 三值化和 int8 量化方法也是常见的低比特量化方法. Wang 等^[42] 提出了有效的两步量化法 (Two-Step Quantization, TSQ) 框架, 将网络量化问题分解为两个步骤: 第一, 使用稀疏法对激活值进行量化, 只对重要的正值进行量化, 将其他不重要的值设为零; 第二, 假设经过第一步量化激活值获得的编码是最优的, 则可以将最优化问题表述为具有低位约束的非线性最小二乘回归问题, 并用迭代求解量化权重. Mellempud 等^[43] 提出了利用参数动态范围内的局部相关性来最小化量化对整体精度的影响的细粒度三元化方法. Zhu 等^[44] 基于误差敏感的学习率调节和方向自适应的梯度截断方法解决了量化后的精度损失. 表 2 给出了近年来在 CIFAR10 和 ImageNet 数据集上量化方法的性能对比, 其中, W 代表权重使用的比特位数, A 代表激活值使用的比特位数, $\text{acc}_{\text{Top-1}}$ 代表预测概率最高的类别与真实类别相符的准确率, $\text{acc}_{\text{Top-5}}$ 代表预测概率排名前五的类别中包含真实类别的准确率.

表 2 CIFAR10 和 ImageNet 数据集上不同量化方法的性能对比

Tab. 2 Performance comparison of different quantization methods on the CIFAR10 and ImageNet datasets

网络(数据集)	压缩方法	W/bit	A/bit	$\text{acc}_{\text{Top-1}}/\%$	$\text{acc}_{\text{Top-5}}/\%$
VGG-Small(CIFAR10)		32	32	93.8	
	BNN ^[22]	1	1	89.9	
	XNOR-Net ^[38]	1	1	89.8	
	IR-Net ^[34]	1	1	90.4	
	TSQ ^[35]	3	2	93.5	
	BWN ^[38]	1	32	90.1	
ResNet-18(ImageNet)		32	32	69.6	89.20
	BWN ^[38]	1	32	60.8	83.00
	Bi-Real ^[21]	1	1	56.4	79.50
	TWN ^[23]	2	32	61.8	84.20
	IR-Net ^[34]	1	32	62.9	84.10
	IR-Net ^[34]	1	1	58.1	80.00
	BENN ^[30]	1	1	61.0	
	CI-BCNN ^[31]	1	1	59.9	84.18
	CBCN ^[32]	1	1	61.4	82.80

当使用低比特表示参数时, 量化神经网络的精度与全精度神经网络相比下降了很多. 这是由于网络量化阶段引入的噪声使得梯度下降法难以收敛. 当使用非常低的位表示来量化权值和激活时, 问题

可能会更加严重. 在对深度网络模型(如 ResNet^[5])进行较大的压缩和加速时, 若使用同时量化权值和激活值的网络(如 XNOR-Net^[45]), 分类精度损失严重. 此外, 结构化矩阵的限制可能会导致模型的偏差和精度的损失. 因此, 量化方法一般是与其他方法结合使用.

1.3 知识蒸馏

在知识蒸馏中, 使用宽而深的网络训练得到的模型一般称为教师模型(Teacher Model), 比如 VGG-16 等; 使用轻量化的网络训练得到的模型一般称为学生模型(Student Model), 比如 MobileNet 等. 知识蒸馏的基本思想是将大型教师模型的软知识提炼到较小的学生模型中. 2015 年, Hitton 等^[46]首次提出了知识蒸馏的压缩框架, 该框架主要使用教师网络的软输出对学生网络进行指导和惩罚, 其中软输出可以提供更大的信息熵, 提供更多原网络的信息, 为此, 教师网络的软输出被用作一个标签来训练和压缩学生网络. 知识蒸馏过程如图 4 所示. 使用带温度 T 的 softmax 函数 q_i 来生成每个类别的预测概率, 公式为

$$q_i = \frac{\exp(\frac{z_i}{T})}{\sum_j \exp(\frac{z_j}{T})},$$

其中, z_i 是输出向量中第 i 个类别的概率, $j \in \{1, 2, \dots, k\}$, k 为总类别数. T 设为 1 时即为原始的 softmax 函数. 对于同一个输入 x , 教师网络和学生网络分别生成一个软目标, 学生网络联合利用真实标签(hard target)和两个软目标(soft target)作为交叉熵损失(cross entropy loss)函数的输入来学习权重.

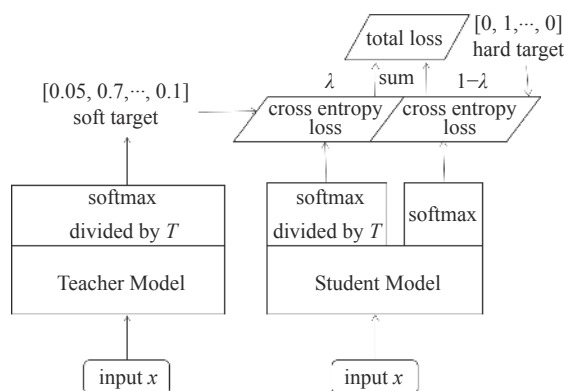


图 4 知识蒸馏.

Fig. 4 Knowledge distillation

知识蒸馏的目标是将原网络中具有代表性的知识转移到另一个更小的神经网络中. 通常是以最小化概率输出之间的 KL(Kullback-Leibler) 散度为目标建立教师网络和学生网络. 但 Tian 等^[47]提出这一 KL 目标忽略了教师网络的重要知识结构. 并提出了对比学习方法. 对比学习的关键思想是学习“正”对在某些度量空间中接近的表示, 并将“负”对之间的表示分离开. 实验表明, 该方法在各种知识转移任务(包括单模型压缩、集成蒸馏和跨模态转移)上的性能都优于知识蒸馏和其他最优的蒸馏方法, 有时甚至超过教师网络与知识蒸馏的结合.

近年来, 许多学者在知识蒸馏领域展开工作, 以设计出更好的学生模型. 学生模型不仅模仿老师的行为, 还可以学习教师网络以外的知识, 从而超越教师模型. Furlanello 等^[48]提出的重生神经网络(Born-Again Neural Networks, BAN), 使得最后获得的学生模型可以超越教师模型. 与原来的方法不同, 这种方法的目的不再是压缩模型, 而是将知识从教师模型转移到具有相同能力的学生模型. 如 DenseNet 作为教师模型、ResNet 作为学生模型. 在经过 BAN 蒸馏后的 ResNet, 其测试错误率比原来的教师模型 DenseNet 更低. 在教师模型收敛之后, 初始化一个新的学生模型. 然后, 在满足设置正

确的预测标签和匹配教师模型输出分布的双重要求的同时,对学生模型进行训练. Gao 等^[49]提出的残差知识蒸馏方法,引进了助理 (Assistant) 的概念,即助理通过学习教师和学生之间的残差来帮助学生获取更多的知识.

基于知识蒸馏的方法可以压缩更深更大的模型,并有助于显著降低计算成本. 而知识蒸馏大多数都是应用于具有 softmax 损失函数的分类任务. 近年来,许多研究使用知识蒸馏的方法对目标检测、语义分割任务进行压缩. He 等^[50]提出了一种有效的语义分割知识蒸馏方法,在不引入额外参数或计算的情况下,大幅度提高了学生模型的能力,并以更少的计算开销获得了更好的结果. 此方法可以分为两个阶段: 第一阶段通过自编码器将知识压缩成紧凑的形式,目的是将教师网络中的知识转为更具代表性信息的压缩空间; 第二阶段为了从教师网络中获取长期依赖关系,提出了亲和蒸馏模型 (Affinity Distillation Module), 主要添加一个卷积层 (称为特征适配器) 来解决教师模型和学生模型各自 feature map 不匹配的问题. 知识蒸馏可以用于许多领域,比如自然语言处理、半监督学习等. 教师网络的软输出可以对学生网络进行微调以获得更好的精度.

1.4 紧凑神经网络设计

随着人们对神经网络原理的理解和在实践中验证,神经网络模型的发展正朝更小卷积核、覆盖更多特征信息、减少冗余的方向发展. 与压缩已经训练好的模型相比,设计紧凑的神经网络是另一种方法. 紧凑网络设计主要针对卷积网络设计一种更高效、计算复杂度更低的方法,在不损失网络性能的情况下减少每秒浮点运算次数,并降低模型体积. 基于此思想,设计高效的 CNN 结构可以从不同的卷积核和卷积方式进行设计. 比如,在设计卷积核时,将全连接层换成全局平均池化层^[51],用小卷积核 (1×1 , 3×3) 替换大卷积核 (5×5). 在设计卷积方式时,使用分组卷积、深度可分离卷积、反卷积等方式替换原来的标准卷积方式,从而加快了网络计算速度,减少了计算量.

Iandola 等^[52]提出的 SqueezeNet、Google 团队提出的 MobileNetV1-V3 系列^[53-55]、Face++ 团队提出的 ShuffleNetV1-V2 系列^[56-57]都是近年来出现的并具有代表性的紧凑神经网络,其共同点是大量地使用了 1×1 小滤波器.

SqueezeNet 由多个 Fire Module、卷积层 (Conv)、采样层 (Pooling) 和全连接层 (Fully Connected layers, FC) 组成. Fire Module 的 squeeze 层用 1×1 的小滤波器替换原来的 3×3 滤波器, expand 层组合使用 1×1 滤波器和 3×3 滤波器,减少了原始网络的大滤波器 (3×3),从而减少了网络的参数量. 该模型在可以达到 AlexNet 分类精度的同时,可将模型参数大小降至原来的 1/50 倍.

MobileNetV1 提出深度分离卷积来代替原来的标准卷积计算. MobileNetV1 将卷积运算分为 depth-wise 和 point-wise,如图 5 所示. 在 depth-wise 中,每个滤波器只考虑一个通道. 滤波器的数目等于输入通道的数目,其中滤波器是一个 3×3 矩阵. point-wise 运算与常规卷积运算非常相似,但使用的是 1×1 的小滤波器. 经过这两个运算后,滤波器的参数将大大降低. MobileNetV2 同样采用了 MV1 的 depth-wise 和 point-wise,并设计了 Inverted Residuals 来获取更多的特征信息,以减少推理时间. 由于在低维空间中增加非线性会破坏其特征, MV2 在每一个块 (block) 中去除了 Relu 层,引入 Linear Bottlenecks. MobileNetV3 则通过神经结构搜索获得子网络,并在 MV2 的 block 中添加了 SENet^[58] 增强模块,使得网络提取特征的能力大大增强,从而获得了更高的准确性.

ShuffleNetV1 使用组卷积 (Group Convolution) 来降低模型参数大小,并使用通道混排 (Channel Shuffle) 增强各个特征图的连接. ShuffleNetV2 提出了 4 条指导性原则: ①输入输出通道数相等会最小化内存访问成本; ②组卷积数目过多将增加内存访问的次数; ③在多路结构中,网络碎片化降低了并行化程度; ④元素级操作 (Element-Wise Operators) 不可忽略. 基于以上 4 条原则对 ShuffleNetV1 进行改进,提升了准确度和模型运行速度.

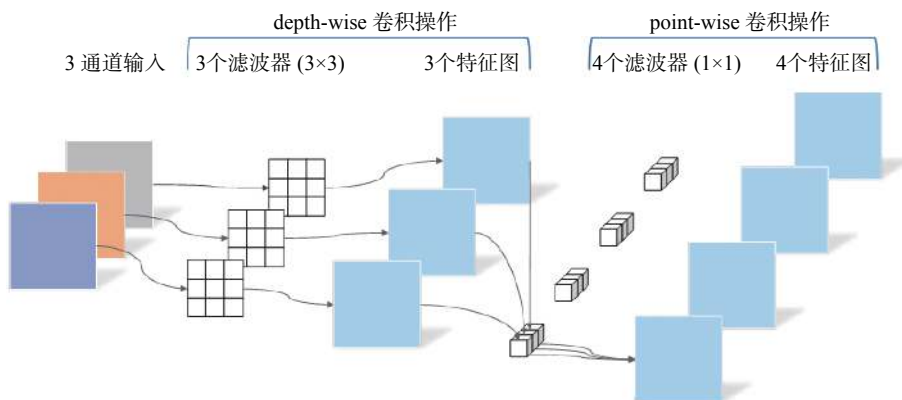


图 5 MobileNetV1 结构

Fig. 5 Structure of MobileNetV1

华为诺亚方舟团队的 Han^[59] 等提出了 GhostNet 模块, 对于每一个输入, 首先生成一组内部特征图, 然后基于这组内部的特征图, 使用一系列成本低廉的线性变换来生成许多 Ghost 特征图, 最后该 Ghost 特征图和内部特征图的个数之和等于原始特征图个数之和, 并利用这些模块, 构建了 Ghost bottleneck(G-bneck) 和 GhostNet. 其中 Ghost bottleneck 类似于 ResNet 中的残差块, GhostNet 则使用 MobileNetV3 作为基本体系架构, 并将 MobileNetV3 中的 bottleneck 换成 G-bneck. 在 ImageNet 分类中, GhostNet 可以获得比 MNV3 更高的精确度. Chen 等人^[60] 提出的 Octave Convolution, 基于自然图像可以分为高频和低频两部分的思想^[61](经过傅里叶变换后对应的高低频), 将卷积层输出的特征图分为高低频两部分, 将低频的特征图存储在更小的张量中, 减少空间冗余. 表 3 给出了近年来紧凑神经网络方法在 ImageNet 数据集上实验的性能对比, 其中, Param 代表网络中的参数量大小, FLOPs(Floating Point Of Operations) 为浮点运算数.

表 3 ImageNet 数据集上不同紧凑神经网络方法的性能对比

Tab. 3 Performance comparison of different compact neural network methods on the ImageNet dataset

模型	Param/ $(\times 10^6)$	acc _{Top-1} /%	FLOPs/ $(\times 10^9)$	推理延迟/ms
SqueezeNet ^[52]	1.25	57.5	1.70	
MobileNetV2 ^[54]	3.40	70.6	0.30	75
MobileNetV3 ^[55]	5.40	75.2	0.22	
ShuffleNetV1 ^[56]	3.40	71.5	0.53	108
ShuffleNetV2 ^[57]	5.30	73.7	0.30	
GhostNet ^[59]	5.20	73.9	0.14	
Oct-MobileNetV2 ^[60]	3.50	72.0	0.27	53
CondenseNet ^[62]	4.80	73.2	0.53	1 890

紧凑神经网络采用紧凑的卷积核或改变卷积方式, 大大减少了模型中的参数, 降低了模型的大小. 在轻量的语义分割、目标检测和分类模型上都有应用. 但由于其特殊的卷积核, 难以和其他模型压缩方法一起使用.

1.5 其他深度神经网络压缩算法

训练好的模型卷积核存在低秩特性, 所以低秩分解也可用在模型压缩中, 主要通过矩阵或张量分

解模型中的参数, 如 SVD(Singular Value Decomposition) 分解、Tucker 分解和 CP(Canonical Polyadic) 分解等. Jaderberg 等^[63] 使用低秩扩展来加速卷积神经网络. 主要观点是利用存在于不同通道和滤波器之间的冗余, 对已训练好的网络进行加速, 并提出了两种优化方案, 可以很容易地应用于现有的 CPU 和 GPU 卷积框架: 第一, 将某一个通道进行秩 1 分解, 并学习 M 个滤波器基组后通过线性组合来近似原来的滤波器; 第二, 每个卷积层被分解为两个常规卷积层的序列, 但在空间域内带有矩形滤波器. 所得近似结果需要的计算操作要少得多, 在场景文本字符识别 CNN 训练中可以观察到, 分类精度仅下降 1%, 但加速率却可以达到 4.5×1 .

另外, 各种压缩方法也可以结合起来一起用. Han 等^[33] 提出的三阶段方法, 结合了剪枝和量化两个方法: 首先修剪值较小的连接, 获得一个稀疏化网络; 再对剪枝后的网络使用聚类的方法进行量化, 实现权值共享; 经过前两个步骤后重新训练网络以调整剩余的连接和量化质心, 最后使用哈弗曼编码对网络再进一步压缩. 该压缩方法在 VGG-16 上压缩率高达 $49 \times$, 而准确率却没有降低. Polino^[64] 等结合量化和知识蒸馏对模型进行压缩. 在量化的过程中加入知识蒸馏损失进行训练, 并使用量化后的权重计算梯度更新模型. 但在量化神经网络中, 对原始权重进行量化的过程是离散的, 因此梯度几乎处处为 0, 这意味着不能使用量化函数进行反向传播. 基于此问题, Polino 等还提出了可微量化方法, 通过随机梯度下降优化量化点的位置. 该方法使得量化的浅层次学生模型可以达到与全精度教师模型相似的水平, 同时可获得高压缩率并加速推理过程. Cai^[65] 等提出了面向边缘应用的压缩方法, 根据网络中各层权重的分布对权重进行裁剪, 并针对嵌入式设备的特性对权重和激活值采用定点量化的方法进行量化. 在降低计算量和存储成本的同时几乎达到了无损压缩.

2 评价准则

衡量模型压缩和加速质量的标准是压缩率、加速率和浮点运算数 (FLOPs). 假设 μ 是原始模型 M 中参数的内存成本, μ^* 是压缩模型 M^* 的内存成本, 则模型的压缩率 φ 为

$$\varphi(M, M^*) = \frac{\mu}{\mu^*}.$$

类似的, 假设 v 是原始模型 M 的推理时间, v^* 是压缩模型 M^* 的推理时间, 则加速率 ϕ 可以被定义为

$$\phi(M, M^*) = \frac{v}{v^*}.$$

假设卷积是以滑动窗口实现的, 并且非线性函数是可以计算的, 卷积核 FLOPs 的计算定义为

$$F_{\text{FLOPs}} = 2HW(C_{\text{in}}K^2 + 1)C_{\text{out}},$$

其中, H 、 W 和 C_{in} 分别是输入特征图的高度、宽度和通道数, K 是卷积核宽度 (假定卷积核长宽相等), C_{out} 是输出通道数, 全连接层 FLOPs 的计算 (本文用 F_{FLOPs} 表示) 定义为

$$F_{\text{FLOPs}} = (2I - 1)O,$$

其中, I 是输入维数, O 是输出维数.

另外模型压缩方法用在分类、目标检测和语义分割任务上时, 根据数据集的不同会定义不同的评价准则. 如在分类任务中, 小数据集 CIFAR 和 MINST 使用 Top-1 分类错误率, 而大数据集 ImageNet

1 “ \times ”代表与原始速度 (模型大小) 相比, 压缩后的加速 (压缩) 倍率.

常使用 Top-1 和 Top-5 分类错误率. 在目标检测或语义分割任务中, 常使用平均精度 (Average Precision, AP) 来做评估. 一般来说, 压缩后的方法与原模型的分类错误率相似, 但参数减少、加速率增加、浮点运算操作数降低.

表 4 给出了近年来在数据集 ImageNet 上进行图像分类的压缩方法性能对比, 其中参数量 (Param) 为 float32 类型 (4 字节). 从表 4 可以看出, 对 AlexNet 和 VGG-16 这些大型神经网络进行压缩, 几乎没有精度损失, 甚至还会超过原来的精度. 对 ResNet-50 这类原始参数本来就较小的网络进行压缩, 精度损失也不大.

表 4 ImageNet 数据集上不同压缩算法的性能对比

Tab. 4 Performance comparison of different compression methods on the ImageNet dataset

网络	压缩方法	Param/ $(\times 10^6)$	ϕ	acc _{Top-1} /%	acc _{Top-5} /%	FLOPs/ $(\times 10^9)$	ϕ
AlexNet		61	1 \times	57.22	80.27	0.72	1.0 \times
	Han等 ^[33]	1.70	35 \times	57.22	80.30		3.0 \times
	Zhang等 ^[18]	2.90	21 \times		80.20		
VGG-16		138.00	1 \times	68.50	88.68	15.50	1.0 \times
	Luo等 ^[23]	8.32	16.63 \times	67.34	87.92	9.34	2.3 \times
	Han等 ^[33]	11.30	49 \times	68.83	89.09		(3.0 \sim 4.0) \times
	Yu等 ^[66]	9.70	15 \times	68.75	89.06		
	Cheng等 ^[67]	28.00	19.6 \times	67.37	88.23		4.9 \times
	Hu等 ^[68]	9.20	15 \times	64.78	86.03	4.40	2.5 \times
ResNet-50		25.56	1 \times	72.88	91.14	7.72	1.0 \times
	Luo等 ^[23]	8.66	2.60 \times	68.42	88.30	2.20	
	Zhuang等 ^[24]	12.38	2.06 \times	71.82	90.53	3.41	
	Pierre等 ^[35]	5.09	19 \times	73.79			

注: “ \times ”代表与原始速度 (模型大小) 相比, 压缩后的加速 (压缩) 倍率

表 5 给出了近年来在 Microsoft COCO 数据集上进行目标检测任务的压缩方法对比, 其中 AP (Average Precision) 代表平均精度, $AP_{0.5}$ 代表交并比 (Intersection over Union, IoU) 阈值为 0.5 时的 AP 值, $AP_{0.75}$ 代表在 IoU 阈值为 0.75 时的 AP 值. 在对目标检测模型进行压缩时, 可以在骨干网络 (Backbone) 使用以上介绍的 4 种主流方法进行压缩或者设计更小型的网络结构. 从表 5 可以看出, EfficientDet-B0 在 Microsoft COCO 2017 数据集上表现出了优异的性能, 并且参数量和 FLOPs 都降低了.

3 总结和展望

本文主要对主流的基于深度神经网络压缩的方法进行了介绍: 首先分析了需要模型压缩的原因; 其次, 介绍了 4 种具有代表性的深度神经网络压缩方法, 即网络剪枝、量化、知识蒸馏和紧凑网络设计, 并着重对近年来压缩模型方法的性能进行了分析; 最后给出了模型压缩的评估准则. 接下来对模型压缩未来的发展和面临的挑战进行展望.

深度网络压缩的基本目的是从网络中提取有用的信息, 并降低模型大小和参数量. 从目前的研究

结果来看, 深度神经网络压缩仍处于发展阶段, 压缩方法本身的性能还有待提高. 以下是一些值得研究和探索的方向.

表 5 Microsoft COCO 数据集上不同压缩方法的性能对比

Tab. 5 Performance comparison of different compression methods on the Microsoft COCO dataset

模型	骨干网络	输入维度	Param/ $(\times 10^6)$	FLOPs/ $(\times 10^9)$	AP	AP _{0.5}	AP _{0.75}
Yolov3-Tiny ^[69]	Tiny-Darknet	416 \times 416	12.30	3.49		33.1	
Pelee ^[11]	PeleeNet	304 \times 304	5.98	1.39	22.4	38.3	22.9
SSD ^[53]	MobileNetV1	300 \times 300	6.80	1.20	19.3		
SSD-lite ^[54]	MobileNetV2	320 \times 320	4.30	0.80	22.1		
Tiny-DSOD ^[70]	DDB-Net+D-FPN	300 \times 300		1.12	23.2	40.4	22.8
ThunderNet ^[14]	SNet146	320 \times 320		0.47	23.7	40.3	24.6
EfficientDet ^[71]	EfficientNet-B0	512 \times 512	3.90	2.50	33.8	52.2	35.8
FQN-INT4 ^[72]	RetinaNet18	800 \times 800			28.6	46.9	29.9

注: 带*代表数据集使用Microsoft COCO 2017, 不带*代表数据集使用Microsoft COCO 2015

(1) 高效评估方法和剪枝率的选择: 剪枝技术评估卷积核或核内权重参数的重要性仍然是比较简单的方法, 尽管近年来提出了许多评估方法, 但普遍计算复杂、难度大. 因此, 如何定义更高效的方法来确定卷积核或其他参数的重要性, 是未来值得探索的方向. 另外, 每一层的卷积参数分布是不同的, 对每层的剪枝率的选择也是未来可探索的方向.

(2) 设计新的网络结构: 模型量化后会改变数据的原始分布, 在大型数据集和深度神经网络中模型性能损失大大增加. 主要原因在于现在的网络结构不一定全都适合于量化操作, 量化后保留的信息与全精度所保留的信息不一定相同, 所以如何针对量化设计特定的网络结构也是值得研究的方向.

(3) 多场景模型压缩: 目前的模型压缩方法主要针对深度神经网络, 而深度神经网络压缩模型大多采用 CNN, 主要针对图像分类场景. 常见的神经网络场景还有递归神经网络 (Recurrent Neural Network, RNN)、长期短期记忆网络 (Long Short-Term Memory, LSTM)、强化学习、目标检测和生成式对抗网络 (Generative Adversarial Networks, GAN) 等. 这些应用场景需要的精度往往很高, 现有的压缩方法难以进行高效的压缩. 所以在未来一段时间内, 有必要研究多场景下的模型如何进行压缩.

(4) 端设备部署: 边缘设备和各种小型平台 (如自动驾驶汽车) 的硬件和资源限制仍然是阻碍深度神经网络模型落地的主要原因. 随着模型压缩方法的流行, 对于分类任务的模型压缩方法已经得到了很大的发展, 但对于常见的目标检测、语义分割等任务的研究还很少. 所以如何在资源受限的边缘设备中部署各种大模型, 使得应用真正落地, 仍然是一个大挑战.

(5) 与自动机器学习 (Automated Machine Learning, AutoML) 结合: 如在剪枝中, 使用 AutoML 自动化地选择修剪率, 在量化中, AutoML 根据量化前参数重要性大小来动态选择量化的比特数.

[参 考 文 献]

- [1] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge [J]. International journal of computer vision, 2015, 115(3): 211-252.
- [2] HE Y, SAINATH T N, PRABHAVALKAR R, et al. Streaming end-to-end speech recognition for mobile devices [C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6381-6385.
- [3] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [EB/OL].

- (2019-05-24)[2020-07-02]. <https://arxiv.org/pdf/1810.04805.pdf>.
- [4] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [EB/OL]. (2015-04-10)[2020-07-02]. <https://arxiv.org/pdf/1409.1556.pdf>.
- [5] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016: 770-778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [6] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 2261-2269. DOI: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [7] CHENG Y, WANG D, ZHOU P, et al. A survey of model compression and acceleration for deep neural networks [EB/OL]. (2020-06-14)[2020-07-02]. <https://arxiv.org/pdf/1710.09282.pdf>.
- [8] 雷杰, 高鑫, 宋杰, 等. 深度网络模型压缩综述 [J]. 软件学报, 2018, 29(2): 251-266.
- [9] CHOUDHARY T, MISHRA V, GOSWAMI A, et al. A comprehensive survey on model compression and acceleration [J/OL]. Artificial Intelligence Review, 2020. (2020-02-08)[2020-07-02]. <https://doi.org/10.1007/s10462-020-09816-7>.
- [10] 李江昀, 赵义凯, 薛卓尔, 等. 深度神经网络模型压缩综述 [J]. 工程科学学报, 2019, 41(10): 1229-1239.
- [11] WANG R J, LI X, LING C X. Pelee: A real-time object detection system on mobile devices [C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2018: 1967-1976.
- [12] CHEN X L, GIRSHICK R, HE K M, et al. TensorMask: A foundation for dense object segmentation [C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 2061-2069.
- [13] SANH V, DEBUT L, CHAUMOND J, et al. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter [EB/OL]. (2020-01-24)[2020-07-01]. <https://arxiv.org/pdf/1910.01108v3.pdf>.
- [14] QIN Z, LI Z, ZHANG Z, et al. ThunderNet: Towards real-time generic object detection on mobile devices [C]//Proceedings of the IEEE International Conference on Computer Vision. IEEE, 2019: 6718-6727.
- [15] ANWAR S, SUNG W. Coarse pruning of convolutional neural networks with random masks[EB/OL]. [2020-07-02]. <https://openreview.net/pdf?id=HkvS3Mqxe>.
- [16] LECUN Y, DENKER J S, SOLLIA S A. Optimal brain damage [C]//Advances in Neural Information Processing Systems. 1989: 598-605.
- [17] HASSIBI B, STORK D G. Second order derivatives for network pruning: Optimal brain surgeon [C]//Advances in Neural Information Processing Systems. 1993: 164-171.
- [18] ZHANG T, YE S, ZHANG K, et al. A systematic dnn weight pruning framework using alternating direction method of multipliers [C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 184-199.
- [19] MA X L, GUO F M, NIU W, et al. PCONV: The missing but desirable sparsity in DNN weight pruning for real-time execution on mobile devices [C]//Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20). 2020: 5117-5124.
- [20] HE Y, ZHANG X, SUN J. Channel pruning for accelerating very deep neural networks [C]// Proceedings of the IEEE International Conference on Computer Vision. 2017: 1389-1397.
- [21] CHIN T W, DING R, ZHANG C, et al. Towards efficient model compression via learned global ranking [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1518-1528.
- [22] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 11264-11272.
- [23] LUO J H, WU J, LIN W. Thinet: A filter level pruning method for deep neural network compression [C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 5058-5066.
- [24] ZHUANG Z W, TAN M K, ZHUANG B, et al. Discrimination-aware channel pruning for deep neural networks [C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems(NIPS'18). New York: Curran Associates Inc., 2018: 883-894.
- [25] HE Y, LIU P, WANG Z, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4340-4349.
- [26] LIN M, JI R, WANG Y, et al. HRank: Filter pruning using high-rank feature map [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1529-1538.
- [27] LIN X, ZHAO C, PAN W. Towards accurate binary convolutional neural network [C]//Advances in Neural Information Processing Systems. 2017: 345-353.
- [28] 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 [C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 722-737.
- [29] HUBARA I, COURBARIAUX M, SOUDRY D, et al. Binarized neural networks [C]//Advances in Neural Information Processing Systems. 2016: 4107-4115.
- [30] LI F F, ZHANG B, LIU B. Ternary weight networks [EB/OL]. (2016-11-19)[2020-07-03]. <https://arxiv.org/pdf/1605.04711.pdf>.
- [31] WANG P, CHENG J. Fixed-point factorized networks [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 4012-4020.
- [32] BOROUMAND A, GHOSE S, KIM Y, et al. Google workloads for consumer devices: Mitigating data movement bottlenecks [C]//Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems. 2018: 316-331.
- [33] HAN S, MAO H Z, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and

- huffman coding [EB/OL]. (2015-11-20)[2020-07-03]. <https://arxiv.org/pdf/1510.00149v3.pdf>.
- [34] CHEN W, WILSON J, TYREE S, et al. Compressing neural networks with the hashing trick [C]// International Conference on Machine Learning. 2015: 2285-2294.
- [35] STOCK P, JOULIN A, GRIBONVAL R, et al. And the bit goes down: Revisiting the quantization of neural networks [EB/OL]. (2019-12-20)[2020-07-02]. <https://arxiv.org/pdf/1907.05686.pdf>.
- [36] CARREIRA-PERPINÁN M A, IDELBAYEV Y. Model compression as constrained optimization, with application to neural nets. Part II: Quantization [EB/OL]. (2017-07-13)[2020-07-03]. <https://arxiv.org/pdf/1707.04319.pdf>.
- [37] ZHU S, DONG X, SU H. Binary ensemble neural network: More bits per network or more networks per bit? [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4923-4932.
- [38] WANG Z, LU J, TAO C, et al. Learning channel-wise interactions for binary convolutional neural networks [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 568-577.
- [39] LIU C, DING W, XIA X, et al. Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 2691-2699.
- [40] COURBARIAUX M, BENGIO Y, DAVID J P. BinaryConnect: Training deep neural networks with binary weights during propagations [C]//Advances in Neural Information Processing Systems. 2015: 3123-3131.
- [41] QIN H T, GONG R H, LIU X L, et al. Forward and backward information retention for accurate binary neural networks [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 2247-2256.
- [42] WANG P, HU Q, ZHANG Y, et al. Two-step quantization for low-bit neural networks [C]// Proceedings of the IEEE Conference on computer vision and pattern recognition. 2018: 4376-4384.
- [43] MELLEMPUDI N, KUNDU A, MUDIGERE D, et al. Ternary neural networks with fine-grained quantization [EB/OL]. (2017-05-30)[2020-07-03]. <https://arxiv.org/pdf/1705.01462.pdf>.
- [44] ZHU F, GONG R, YU F, et al. Towards unified int8 training for convolutional neural network [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1969-1979.
- [45] RASTEGARI M, ORDONEZ V, REDMON J, et al. Xnor-net: Imagenet classification using binary convolutional neural networks [C]//European Conference on Computer Vision. Cham: Springer, 2016: 525-542.
- [46] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network [EB/OL]. (2015-03-09)[2020-07-04]. <https://arxiv.org/pdf/1503.02531.pdf>.
- [47] TIAN Y L, KRISHNAN D, ISOLA P. Contrastive representation distillation [EB/OL]. (2020-01-18)[2020-07-04]. <https://arxiv.org/pdf/1910.10699.pdf>.
- [48] FURLANELLO T, LIPTON Z C, TSCHANEN M, et al. Born again neural networks [C]//Proceedings of the 35th International Conference on Machine Learning. 2020: 1602-1611.
- [49] GAO M Y, SHEN Y J, LI Q Q, et al. Residual knowledge distillation [EB/OL]. (2020-02-21)[2020-07-04]. <https://arxiv.org/pdf/2002.09168.pdf>.
- [50] HE T, SHEN C, TIAN Z, et al. Knowledge adaptation for efficient semantic segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 578-587.
- [51] LIN M, CHEN Q, YAN S C. Network in network [EB/OL]. (2014-03-04)[2020-07-04]. <https://arxiv.org/pdf/1312.4400/>.
- [52] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size [EB/OL]. (2016-11-04)[2020-07-04]. <https://arxiv.org/pdf/1602.07360.pdf>.
- [53] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications [EB/OL]. (2017-04-17)[2020-07-04]. <https://arxiv.org/pdf/1704.04861.pdf>.
- [54] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks [C] //Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.
- [55] HOWARD A, SANDLER M, CHU G, et al. Searching for mobilenetv3 [C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 1314-1324.
- [56] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6848-6856.
- [57] MA N, ZHANG X, ZHENG H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design [C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 116-131.
- [58] HU J, SHEN L, SUN G. Squeeze-and-excitation networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
- [59] HAN K, WANG Y, TIAN Q, et al. GhostNet: More features from cheap operations [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1580-1589.
- [60] CHEN Y, FAN H, XU B, et al. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution [C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 3435-3444.
- [61] CAMPBELL F W, ROBSON J G. Application of Fourier analysis to the visibility of gratings [J]. The Journal of Physiology, 1968, 197(3): 551-566.
- [62] HUANG G, LIU S, VAN DER MAATEN L, et al. Condensenet: An efficient densenet using learned group convolutions [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2752-2761.
- [63] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions [EB/OL].

- (2014-05-15)[2020-07-04]. <https://arxiv.org/pdf/1405.3866.pdf>.
- [64] POLINO A, PASCANU R, ALISTARH D. Model compression via distillation and quantization [EB/OL]. (2018-02-15)[2020-07-04]. <https://arxiv.org/pdf/1802.05668.pdf>.
- [65] 蔡瑞初, 钟椿荣, 余洋, 等. 面向“边缘”应用的卷积神经网络量化与压缩方法 [J]. 计算机应用, 2018, 38(9): 2449-2454.
- [66] YU X Y, LIU T L, WANG X C, et al. On compressing deep models by low rank and sparse decomposition [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 7370-7379.
- [67] CHENG J, WU J X, LENG C, et al. Quantized CNN: A unified approach to accelerate and compress convolutional networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 29(10): 4730-4743.
- [68] HU H Y, PENG R, TAI Y W, et al. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures [EB/OL]. (2016-07-12)[2020-7-04]. <https://arxiv.org/pdf/1607.03250.pdf>.
- [69] WANG R J, LI X, LING C X. Pelee: A real-time object detection system on mobile devices [C]//Advances in Neural Information Processing Systems. 2018: 1963-1972.
- [70] LI Y, LI J, LIN W, et al. Tiny-DSOD: Lightweight object detection for resource-restricted usages[EB/OL]. (2018-07-29)[2020-07-04]. <https://arxiv.org/pdf/1807.11013.pdf>.
- [71] TAN M, PANG R, LE Q V. Efficientdet: Scalable and efficient object detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10781-10790.
- [72] LI R, WANG Y, LIANG F, et al. Fully quantized network for object detection [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 2810-2819.

(责任编辑: 李 艺)