工程科学学报,第 41 卷,第 10 期:1229-1239,2019 年 10 月
Chinese Journal of Engineering, Vol. 41, No. 10: 1229-1239, October 2019
DOI: 10.13374/j.issn2095-9389.2019.03.27.002; http://journals.ustb.edu.cn

深度神经网络模型压缩综述

李江昀[1,2)], 赵义凯[1,2)], 薛卓尔[1)], 蔡　铮[1)], 李　擎[1,2)]✉

1)北京科技大学自动化学院, 北京 100083　2) 工业过程知识自动化教育部重点实验室, 北京 100083
✉通信作者, E-mail: Liqing@ies.ustb.edu.cn

摘　要　深度神经网络近年在计算机视觉以及自然语言处理等任务上不断刷新已有最好性能,已经成为最受关注的研究方向. 深度网络模型虽然性能显著,但由于参数量巨大、存储成本与计算成本过高,仍然难以部署到硬件受限的嵌入式或移动设备上. 相关研究发现,基于卷积神经网络的深度模型本身存在参数冗余,模型中存在对最终结果无用的参数,这为深度网络模型压缩提供了理论支持. 因此,如何在保证模型精度条件下降低模型大小已经成为热点问题. 本文对国内外学者近几年在模型压缩方面所取得的成果与进展进行了分类归纳并对其优缺点进行评价,并探讨了模型压缩目前存在的问题以及未来的发展方向.

关键词　深度神经网络; 模型压缩; 深度学习; 网络剪枝; 网络蒸馏
分类号　TP183

A survey of model compression for deep neural networks

LI Jiang-yun[1,2)], *ZHAO Yi-kai*[1,2)], *XUE Zhuo-er*[1)], *CAI Zheng*[1)], *LI Qing*[1,2)]✉

1) School of Automation & Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China
2) Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing 100083, China
✉Corresponding author, E-mail: Liqing@ies.ustb.edu.cn

ABSTRACT　In recent years, deep neural networks (DNN) have attracted increasing attention because of their excellent performance in computer vision and natural language processing. The success of deep learning is due to the fact that the models have more layers and more parameters, which gives them stronger nonlinear fitting ability. Furthermore, the continuous updating of hardware equipment makes it possible to quickly train deep learning models. The development of deep learning is driven by the greater amounts of available annotated or unannotated data. Specifically, large-scale data provide models with greater learning space and stronger generalization ability. Although the performance of deep neural networks is significant, they are difficult to deploy in embedded or mobile devices with limited hardware due to their large number of parameters and high storage and computing costs. Recent studies have found that deep models based on a convolutional neural network are characterized by parameter redundancy as well as parameters that are irrelevant to the final model results, which provides theoretical support for the compression of deep network models. Therefore, determining ways to reduce model size while retaining model precision has become a hot research issue. Model compression refers to the reduction of a trained model through some operation to obtain a lightweight network with equivalent performance. After model compression, there are fewer network parameters and usually a reduction in the computation required, which greatly reduces the computational and storage costs and enables the deployment of the model in restricted hardware conditions. In this paper, the achievements and progress made in recent years by domestic and foreign scholars with respect to model compressionwere classified and summarized and their advantages and disadvantages were evaluated, including network pruning, parameter sharing, quantization, network decomposition, and network distillation. Then, existing problems and the future development of model compression were discussed.

KEY WORDS　deep neural networks; model compression; deep learning; network pruning; network distilling

收稿日期: 2019-03-27
基金项目: 国家自然科学基金资助项目(61671054); 北京市自然科学基金资助项目(4182038)

近几年,深度学习[1]受到极大关注,在图像处理、自然语言处理、语音处理等方面都取得了成功应用,已经成为机器学习中的热点领域. 深度学习的热潮源于多伦多大学的 Krizhevsky 等[2]在 2012 年的图像视觉识别比赛(imagenet large scale visual recognition competition,ILSVRC)上,通过搭建深度神经网络(convolutional neural networks,CNN),即 AlexNet 最终在图像分类以及图像定位上取得最优精度,远超过其他基于传统机器学习的方法. 自此,深度学习受到学术界和工业界前所未有的追捧. 后来的图像视觉识别比赛中,VGG[3]、GoogLeNet[4]、ResNet[5]、DenseNet[6]等网络陆续出现,并不断刷新比赛最佳性能,尤其是在图像识别任务上 ResNet 的准确率首次超越人类. 深度学习之所以会取得成功,一方面是因为模型具有更深层数和更多参数,这使得模型具有更强大的非线性拟合能力;另一方面是因为硬件设备不断更新为深度学习模型快速训练提供了可能;此外,更多可获得的标注或未标注数据推动了深度学习的发展,具体来说,大规模数据意味着模型将有更大的学习空间以及更强大的泛化能力.

尽管深度网络模型在许多任务上都取得了不俗表现,但这仅仅是在科研层面上进行的实验与尝试,如果考虑到实际应用将其移植到嵌入式或者移动设备上,它将会受到多方面约束:1)模型参数量巨大:比如 AlexNet 模型大小约 200 M,VGG-16[3]模型大小超过 500 M,并且随着网络的不断加深,许多模型可以达到 1 G 或 2 G 甚至更大,如此大的模型对于实际应用是巨大挑战;2)模型计算量大:由于深度神经网络模型中存在着大量卷积运算,仅仅运行一次前向传播计算量都很巨大,比如 ResNet50[5]计算量达到 3.8 亿次,ResNet152[5]计算量甚至达到 11.3 亿次. 而这仅仅是基础网络计算量,如果针对具体任务设计的最终模型计算量还要更大. 即使使用图像处理单元[7](graphic processing unit, GPU)或者神经网络处理单元(neural network processing unit, NPU)等进行加速,也依然无法满足商用需求;3)电能消耗大:网络运行时持续的内访问存和计算资源的大量使用都导致耗电量巨大. 对于硬件资源有限的嵌入式或者移动设备来说,深度神经网络直接应用在时间与空间上都受到巨大约束[8]. 因此,如何在保证模型效果的前提下对模型进行压缩已经成为了一个热点问题,这使得深度网络模型压缩快速地发展起来.

模型压缩是指对在保证模型精度情况下对模型进行精简的方法. 经过模型压缩后的网络参数量会更少,计算量通常也会减少,这使得计算开销与存储开销大大降低,为模型部署在受限的硬件条件下提供了可能. 当前存在的模型压缩方法大致可以分为六种方式:网络剪枝(network pruning)、参数共享(parameters sharing)、量化(quantization)、网络分解(network decomposition)、网络蒸馏(network distilling)和紧凑网络设计(compact network design). 本文尽可能详尽地整理了近几年在深度网络模型压缩领域的工作,并对每种方法给出具体评价,同时对深度网络模型压缩当前存在的问题以及未来发展方向进行探讨,最后对全文进行总结.

1　网络剪枝

网络剪枝[9]是深度神经网络压缩中一种广泛使用的方法. 网络剪枝早期是用来删除网络中冗余参数,降低网络复杂度,从而提高网络泛化能力,并防止过拟合. 网络剪枝从剪枝粒度可以分为:单个权重(weight)剪枝、核内权重(intra kernel weight)剪枝、卷积核(kernel)剪枝、通道(channel/feature map/filter)剪枝以及隐层(Layer)剪枝. 不同剪枝方法虽然从思想上类似,但是在实际操作过程中区别很大,最终效果也不尽相同. 以下子章节将对这几种剪枝方法进行详细叙述.

1.1　单个权重剪枝

在 1989 年,LeCun 等[10]提出将网络中所有权重参数都看作单个参数的最优脑损伤(optimal brain damage,OBD)方法. 优化时基于对角假设、极值假设和二次假设利用二阶导数近似参数显著性移除网络中不重要的权重来提高网络精度和泛化能力. Hassibi 等[11-12]基于最优脑损伤方法,提出增加基于手术恢复权重更新步骤的最优脑手术(optimal brain surgeon,OBS)方法. Han 等[13]提出仅仅学习网络中重要连接可以在不影响网络最终精度情况下降低模型参数量与计算量. 基于此方法,Han 等[14]进一步提出三阶段流水线式深度压缩方法,包括剪枝、量化和霍夫曼编码. 为了加速网络计算,Srinivas 等[15]提出使用稀疏计算训练神经网络,通过引入额外门变量进行参数选择,最终网络精度与正常训练结果相当.

1.2　核内权重剪枝

上一节中的单个权重剪枝为非结构化剪枝,这使得所有层变为稀疏分布. 这将导致三个问题:仅仅对部分参数进行剪枝并不能显著降低计算量与参数量;剪枝部分参数将会得到稀疏网络,而目前对稀疏操作加速支持的库非常有限;得到的稀疏数据结构将会需要额外存储开销尤其对于低精度权重.

针对以上非结构化剪枝的问题，Anwar 等[16]率先提出了结构化剪枝的概念. 方法的主要思想是定义显著性变量并进行贪婪剪枝，提出核内定步长粒度将细粒度剪枝转化为粗粒度剪枝如通道剪枝或卷积核剪枝. 核内定步长粒度核心思想为作用在同一输入特征图上的卷积核必须采用相同的步长和偏置，作用在不同特征图上的卷积核步长与偏置可以不同. 进行贪婪剪枝时，提出使用进化粒子滤波器决定网络连接重要性，使用此方法可以保证网络准确率下降更低. Wen 等[17]提出了结构稀疏化的学习方法去对深度神经网络的结构如卷积核、通道、层深度等进行正则化. 此方法可以从一个更大的深度神经网络学习到一个更紧凑的模型来减少参数量和计算量，而且可以获得对深度神经网络硬件友好的结构化稀疏方式来加速网络验证并可以提高最终分类精度.

Lin 等[18]提出结构化稀疏正则化. 此方法结合两种不同的结构化稀疏方式，充分协调全局输出与局部剪枝操作去自适应的剪枝滤波器. 此外，提出基于拉格朗日乘法器的交替更新机制在提升网络结构化稀疏性与优化识别损失之间进行更替. Guo 等[19]提出一种新的模型压缩方法称为“动态网络手术”(dynamic network surgery)，它可以通过网络训练动态地进行网络剪枝. 与之前使用贪婪方式进行剪枝的方法比，此方法将连接拼接的思想结合到整个网络剪枝过程中可以有效避免错误剪枝且使其成为连续过程. Jia 等[20]提出 Drop Prunning 剪枝技术，此方法也采用标准的迭代剪枝-重训练的策略，提出在每一个剪枝阶段都使用策略，随机删除一部分不重要权重和随机恢复一些被剪枝权重，此方法可以解决传统剪枝方法的局部重要性判断以及无法弥补的剪枝过程，原理类似于随机神经元失活.

1.3 卷积核剪枝与通道剪枝

卷积核剪枝与通道剪枝都属于粗粒度剪枝，剪枝后模型使用现有硬件与计算库即可进行训练，并且最终可以获得更高压缩率与更短计算时间. 虽然两种剪枝方式出发点并不一样，但其本质上是一致的，特征图与卷积核具有对应关系，减去某一个特征图的同时与其相连的卷积核也将被一起移除.

Li 等[21]提出基于卷积核剪枝的压缩技术. 主要思想是对网络输出精度影响较小的卷积核进行剪枝，通过移除网络中那些卷积核以及其所连接的特征图，网络计算量大幅度降低. 假设网络第 i 个卷积层的输入通道数为 n_i，h_i/w_i 分别表示输入特征图的宽和高，通过卷积层得到的输出特征图通道数为 n_{i+1}，h_{i+1}/w_{i+1} 分别表示输出特征图的宽和高，卷积核的宽和高均为 k，那么卷积核矩阵 $\boldsymbol{F}_i \in \mathbf{R}^{n_i \times n_{i+1} \times k \times k}$，进行卷积运算的计算量为 $n_{i+1} n_i k^2 h_{i+1} w_{i+1}$. 根据图 1 所示，当一个滤波器 $\boldsymbol{F}_{i,j}$ 被剪枝掉，它所对应的特征图 $\boldsymbol{x}_{i+1,j}$ 也将被移除，这减少了 $n_i k^2 h_{i+1} w_{i+1}$ 次计算量. 此外，下一层中与被剪枝掉特征图所对应的滤波器也将被移除，这又减少 $n_{i+2} k^2 h_{i+2} w_{i+2}$ 的计算量. 为了衡量每层滤波器重要性，提出通过计算每一层滤波器权重绝对值之和，然后移除那些求和值较小的滤波器以及所对应特征图的方法. 为了在多个层同时进行剪枝，又提出独立剪枝和贪婪剪枝两种方式. 对于剪枝后网络训练也提出两种训练方式，分别为多次剪枝操作后再训练与剪枝和重训练交替进行，针对不同难度网络采用不同训练方式.

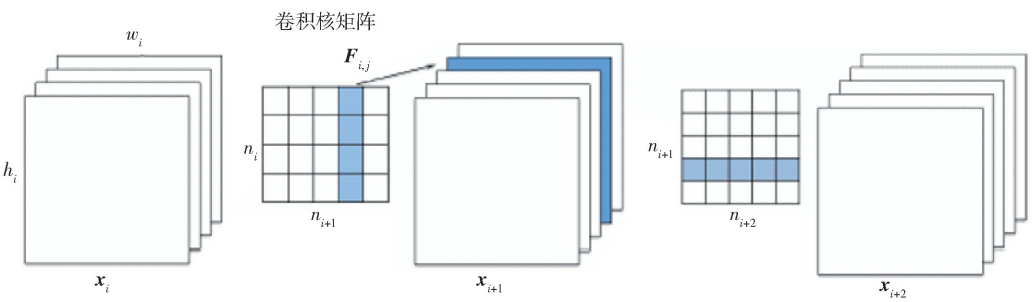

图 1 减去一个滤波器及其对应的特征图[21]

Fig. 1 Pruning a filter and its corresponding feature map[21]

Hu 等[22]提出网络修剪(network trimming)，通过对在大型数据集上神经元的输出分析来减掉不重要部分来迭代地优化网络. Tian 等[23]发现，在最后一个卷积层中，经过线性判别分析(linear discriminant analysis, LDA)分析发现对于每一个类别，有很多滤波器之间的激活是高度不相关的，因此利用这点来剔除大量只具有少量信息的滤波器而不影响模型性能. Luo 等[24]提出将滤波器剪枝当作一个优化问题，并且揭示对滤波器进行剪枝需要基于从下一层计算得到统计信息而非当前层，这是此方法与其

他方法的重要区别. He 等[25]提出软滤波器剪枝加速深度神经网络的推断. 此方法允许被剪掉滤波器在训练中继续更新,这使得网络有更大模型容量与优化空间,并且可以减少对预训练模型依赖,使得模型能从头训练.

He 等[26]提出一种递归两阶段通道剪枝算法去有效地对每层进行剪枝. 首先利用 lasso 回归方法对卷积通道进行选择性删除,然后利用最小二乘重构删除之前网络响应. 进一步将此方法推广到多分支情况,在保证性能前提下降低模型参数量与计算量. Hu 等[27]认为存在的剪枝方法直接重构特征图误差来进行通道选择,这忽略了特征图特征和语义分布以及通道对整体表现的真正贡献,因此提出同时利用基础模型输出以及被剪枝的分类损失函数去监督每层通道选择,尤其引入附加损失去编码基础模型和被剪枝模型特征图内特征与语义分布差异,通过同时考虑重构误差、附加损失和分类损失,被剪枝后模型精度获得极大提高. Zhuang 等[28]提出一个简单有效判别力驱动的通道剪枝去选择真正有助于判别力的通道. 引入判别力驱动损失函数去增加网络中间层判别力并通过考虑附加损失和重构误差去选择出每层最有判别力通道,最后提出一种贪婪算法去进行通道选择以及参数优化. He 与 Han[29]提出一种自动深度压缩方法,利用强化学习去采样设计空间实现了更高的模型压缩率.

评价:网络剪枝是为了移除网络中冗余参数,从而降低网络复杂度,提高网络泛化能力. 因此最重要的问题就是如何确定网络中哪些参数是冗余的,不论是哪种剪枝方式都必须要面对此问题. 为此,许多方法已经被提出来,但是目前并没有明确准则来解决此问题. 大多数方法都需要反复实验来确定,这造成人力与物力的极大浪费. 此外,剪枝后的网络有可能需要额外的存储开销以及计算库,这对于模型在嵌入式或者移动设备上部署造成影响.

2　参数共享

参数共享或者权值共享主要思想就是让网络中多个参数共享同一值,但是具体实现方式不尽相同. Appuswamy 等[30]推导了高效能神经形态结构中网络表示与块托普利兹卷积矩阵之间关系,并在此基础上利用一组结构化卷积矩阵来实现深度卷积网络并取得显著效果. Sindhwani 等[31]提出统一框架来学习一大类以低秩为特征的结构参数矩阵. 结构化转换允许快速函数和梯度评估,并跨越丰富参数共享配置范围,其统计建模能力可以沿着从结构化到非结构化的连续体显式调整. Cheng 等[32]提出使用循环推测方法去代替全连接层中卷积线性预测. 此循环结构可以持续减少内存占用并可以使用快速傅里叶变换来加速计算,大大降低模型空间复杂度. 以上三种方法均是基于结构化矩阵方式来减少内存消耗并加速网络的推理以及训练速度.

Chen 等[33]提出利用网络本身冗余去实现模型压缩的哈希网络. 此方法利用低成本哈希函数随机地将有连接权重分组到哈希桶中,同一个哈希桶内参数将共享同一值,这些参数可以使用反向传播进行调整. 哈希程序不需要额外内存消耗并可以大大提高网络的速度. Shi 等[34]在哈希网络基础上进一步提出函数哈希网络. 对于网络每一层,使用多个低成本哈希函数在压缩空间取值,然后使用小重构网络去恢复那一层输出. Han 等[14]使用 k 均值聚类的方法进行参数共享. 首先修剪不重要连接,重新训练稀疏连接网络,然后使用权重共享连接权重,再对量化后的权重和码本使用霍夫曼编码,以进一步降低压缩率. 如图 2 所示,对训练好的模型,使用 k 均值聚类算法确定每一层共享权重数值,所有属于同一簇权重共享相同权值,但是对不同层权值不共享. 将 n 个原始权重 $\boldsymbol{W}=\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_n\}$ 划分到 k 个聚类中 $\boldsymbol{C}=\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_n\}$,$n\gg k$,最小化簇内平方和(within-cluster sum of squares, WCSS):

$$\underset{C}{\operatorname{argmin}}\sum_{i=1}^{k}\sum_{\boldsymbol{w}\in\boldsymbol{c}_i}|\boldsymbol{w}-\boldsymbol{c}_i|^2 \tag{1}$$

Wu 等[35]提出对权重使用 k 均值算法,通过仅记录 k 个聚类中心以及权重分配指数进行权值共享压缩算法. Lu 等[36]提出对循环神经网络(recurrent neural networks,RNN)使用低秩分解与参数共享机制减少循环结构冗余. Jin 等[37]提出权重采样网络,从一个紧密可学习参数集中采样来学习模型参数. 这种方式天然迫使网络参数共享,且可以同时提升权重与计算共享,最终结合权值量化可以在模型大小以及计算量上取得巨大下降. Yang 等[38]提出卷积滤波器表示的权重共享方法进行模型压缩. 该方法通过学习三维张量滤波器(filter summary, FS),减少卷积层参数个数. 将卷积滤波器作为重叠三维块从张量滤波器中提取,并在它们重叠区域将附近滤波器加入张量滤波器共享权重.

评价:参数共享主要是让网络中多个参数共享一值. 此方法从原理上可以极大的降低模型参数量,但是在实际操作过程中,通常会需要进行 k 均值聚类,模型最终结果与聚类效果息息相关. 而且,如果过分追求模型压缩比,使用参数共享后模型精度

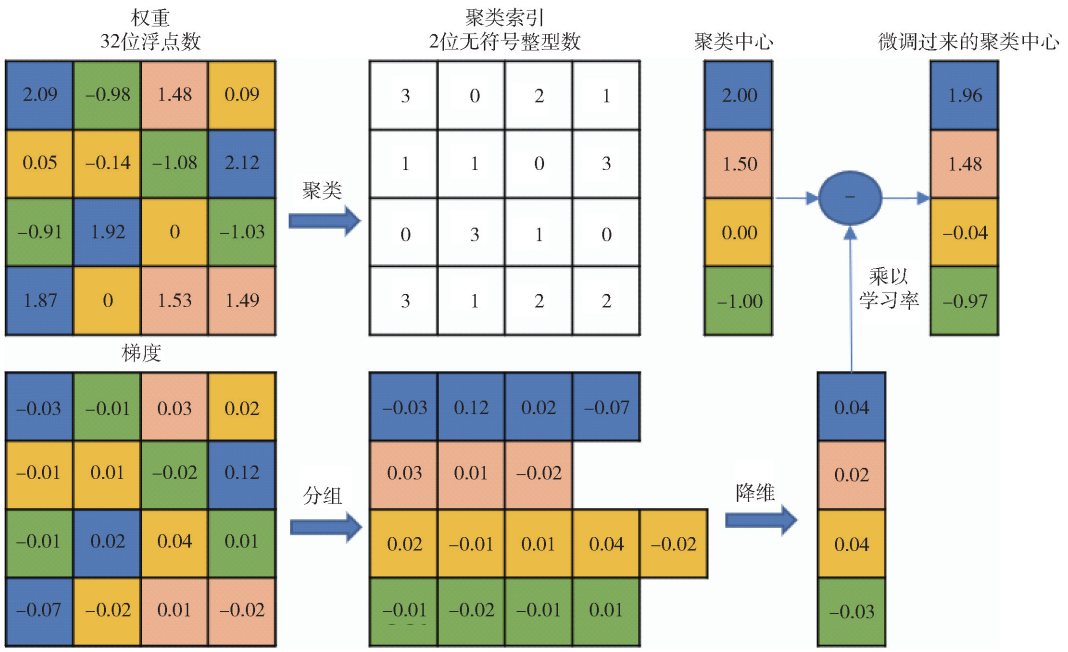

图 2 通过标量量化与中心体微调实现权重共享[14]

Fig. 2 Weight sharing by scalar quantization (top) and centroid fine-tuning (bottom)[14]

会明显下降.

3 量化

量化主要思想是降低权重所需要的比特数来压缩原始网络,主要包括低精度和重编码两类方法.对于卷积神经网络来说,网络模型权重都是单精度浮点型 32 位. 低精度方法使用更低位数的浮点数或整型数对其进行训练、测试或存储;重编码方法对原有数据进行重新编码,采用更少的位数对原有数据进行表示,实现了模型的压缩.

低精度:Gupta 等[39]基于随机修约的卷积神经网络训练中使用 16 比特定点表示法,显著降低内存和浮点运算,同时分类准确率几乎没有受到损失. Han 等人[14]提出对数据冗余精度进行裁剪. 原本的 32 位精度浮点数由“1,8,23”结构构成,根据预训练得到的全精度神经网络模型中的数据分布分别对阶码和位数的长度进行适当的减少. 实验表明,对于大部分人物来说,6 比特或者 8 比特的数据已经足够保证测试的准确率. 目前工业界对于 8 位整型或者 16 位半精度浮点数量化技术已经成熟,比如 TensorFlow-lite[40]和 Nvidia 的 TensorRT 均已经支持 8 位整型数据的网络训练以及存储.

重编码:Courbariaux 等[41-42]提出在运行时使用二值化权重和激活二值化神经网络. 网络训练时,二值化权重与激活用来计算参数梯度,并在前传过程中用位运算代替算术运算大大降低内存大小. Rastegari 等[43]提出两个标准卷积神经网络近似网络,分别为二值权重网络(binary-weight-networks)和同或网络(XNOR-networks). 二值权重网络通过对滤波器权重进行二值量化节省 32 倍内存. 同或网络对输入以及滤波器权重均进行二值量化,结果可以提速 58 倍,减少内存 32 倍. Li 等[44]对同或网络改进,将卷积神经网络网络层输入进行高精度二值量化,从而实现高精度二值网络计算,同或网络也是对每个卷积神经网络网络层权值和输入进行二值化,这样整个卷积神经网络计算都是二值化,计算速度快,占内存小. Hwang 与 Sung[45]提出基于重训练的反向传播优化定点设计. 定点网络使用 3 值权重和 3 位表示信号,结果与浮点型网络相比精度下降很微弱. Hou 等[46]提出具有对角近似牛顿算法直接最小化二值化权值损失的海森近似. 底层近端步骤有封闭形式解决方案,二阶信息可以从 Adam 优化器已经计算的二阶矩有效获得. Lee 等[47]提出基于迭代解决 L1 正则化最小二乘和 L2 约束最小二乘问题的稀疏编码算法. 此算法明显加快稀疏编码速度,并获得比之前更大的稀疏码本. Gudovskiy 与 Rigazio[48]提出用于无乘法卷积神经网络推理的普遍低精度结构 ShiftCNN. ShiftCNN 基于 2 的幂加权表示,因此只执行移位和加法操作. Meller 等[49]提出适当因子分解可以显著降低量化引起的退化. 一个给定网络有许多因子分解,它们在不改变网络功能情况下改变网络权重. Xu 等[50]提出基于强化学习的动态网络量化框架,由位宽控制器与量化器组成. 利用策略梯度训练一个中间体通过位宽控制器

来学习每一层的位宽. 该控制器可以在精度和压缩比之间进行权衡. 在给定量化位宽序列情况下,量化器采用量化距离作为量化过程中权重重要性判据.

评价:目前对于图像分类任务,许多量化技术都可以达到无损压缩,但是当任务变得复杂时,例如图像分割等等,使用量化通常会对模型精度带来巨大影响. 尤其对于模型较大的卷积神经网络,比如 GoogLeNet,二值网络会使网络精度大幅下降. 此外,目前存在的二值化策略都是基于简单的矩阵相乘,这忽略了二值化对精度损失的影响.

4　网络分解

卷积神经网络主要包括卷积层以及全连接层,卷积层计算量较大而参数量较少,而全连接层与之相反. 因此,网络分解主要是将矩阵二维张量奇异值分解(singular value decomposition, SVD)[51]推广到三维卷积核或者使用多个一维张量外积求和逼近来减少网络推断的时间.

Jaderberg 等[52]提出通过利用跨通道或滤波器冗余来构造一个在空间域中秩为 1 的低秩滤波器. 使用秩为 1 的卷积核作用在输入图上产生相互独立的 M 个基本特征图. 此方法与架构无关,可以很容易应用到现有的 CPU 和 GPU 卷积框架上,从而提高性能. Kim 等[53]首先进行变分贝叶斯矩阵分解的秩选择,然后再进行核张量 Tucker 分解,最后再次对模型进行调整. Denil 等[54]提出给定每个特征一部分权重值就可能准确预测剩下的值,并进一步说明不仅参数值可以被预测,而且许多参数甚至不需要学习. Calvi 等[55]提出塔克张量层去代替全连接层的稠密矩阵,将这些权值矩阵视为高阶权值张量展开. 利用张量分解压缩特性,提出利用权张量多途径性质来有效减少参数个数的框架. 塔克分解将权张量分解为核张量和因子矩阵. 通过将矩阵导数的概念扩展到张量,在这个框架内重新推导反向传播,通过计算梯度,利用的塔克分解物理可解释性来洞察训练.

评价:网络分解方法对于模型压缩与加速非常直接,但是由于涉及到矩阵分解操作,实现起来并不容易. 此外,由于不同层含有不同的信息,因此全局压缩对于网络至关重要,但目前方法是按层来进行矩阵分解,这不能进行全局参数压缩. 最后,矩阵分解需要大量模型重训练已达到与原模型一致的收敛效果.

5　网络蒸馏

网络蒸馏或知识蒸馏(knowledge distillation, KD)的概念最早由 Buciluǎ 等[56]提出. 他们训练一个带有伪数据标记强分类器的压缩/集成模型,并恢复原始较大网络输出. 但这项工作仅限于浅层模型. 后来 Ba 与 Caruana[57]将深度较广的网络压缩成较浅的网络,其中压缩模型模仿复杂模型学习函数. 知识蒸馏方法主要思想是通过 softmax 学习类分布输出,将知识从一个大的教师模型转化到一个小的学生模型.

Hinton 等[58]引入知识蒸馏压缩框架,使用教师模型最终 softmax 的输出对学生模型的输出进行监督,使得教师模型的信息可以传递到学生模型中. 知识蒸馏算法虽然简单,但在各种图像分类任务中显示出良好效果. Romero 等[59]提出当模型层数较多时,直接使用教师模型的输出对学生模型进行监督会比较困难,因此提出 Fitnets 模型,将教师模型的中间层输出作为对学生模型的中级监督信息,使得两者中间层的响应尽量一致. 如图 3 所示,首先使用中级监督训练学生模型前半部分参数,然后再利用教师模型最后输出监督整个学生模型参数训练. 其中,$\boldsymbol{W}_{\mathrm{T}}$ 表示教师网络的训练参数,$\boldsymbol{W}_{\mathrm{S}}$ 表示 FitNet 的训练参数,$\boldsymbol{W}_{\mathrm{Hint}}$ 表示教师网络从第一层到第 h 层的训练参数,$\boldsymbol{W}_{\mathrm{Guided}}$ 表示 FitNet 从第一层到指导层的训练参数,$\boldsymbol{W}_{\mathrm{r}}$ 是回归参数. Chen 等[60]提出使用网络生长方法来获得学生模型网络结构. Li 与 Hoiem[61]提出分别从横向和纵向上进行网络生长,再利用知识蒸馏方法训练学生模型.

Zagoruyko 与 Komodakis[62]将注意力图(attention map)作为知识从教师模型迁移到学生模型. 注意力机制很早就被用在自然语言处理中,后来被证明将注意力应用在卷积神经网络中也可以取得不错效果. Mirzadeh 等[63]引入多级知识蒸馏,利用一个中等规模网络来弥合学生和教师之间鸿沟. 研究教师助理粒度影响,并将该框架扩展到多级蒸馏. 并对教师助理知识提炼框架进行实证和理论分析. Liu 等[64]为了同时达到良好性能和可解释性,将深度神经网络引入决策树. 将此问题表示为一个多输出回归问题,实验表明,在相同树深水平上,学生模型比普通决策树具有更好精度性能.

Yang 等[65]引入快照蒸馏(snapshot distillation, SD),这是第一个支持师生一代优化的框架. 快照蒸馏的概念非常简单:不借用前辈监督信号,而是从同一代人早期时代中提取信息,同时确保教师和学

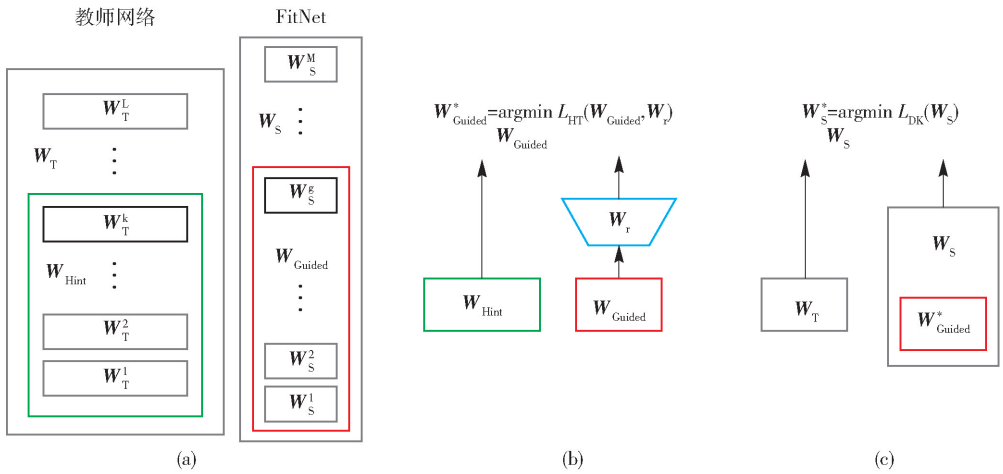

图 3 使用知识提示方式训练学生网络[59]．(a)教师学生网络；(b)暗示学习；(c)知识蒸馏

Fig. 3 Training a student network using hints[59]: (a) teacher and student networks; (b) hints training; (c) knowledge distillation

生之间差异足够大．Wang 等[66]设计了专用模型压缩框架．基于知识蒸馏范式，联合使用暗示学习、蒸馏学习和自学习来训练一个紧凑、快速神经网络．从繁琐模型中提取知识是自适应有界的，并小心地加以扰动，以加强差异隐私．进一步提出一种查询样本选择方法，以减少查询数量和控制隐私．Lee 等[67]提出基于奇异值分解的知识提取方法．此外，将知识转移定义为一个自我监督任务，并提出连续接收教师模型信息的方法．Lan 等[68]提出用于在线蒸馏的实时本地集成学习策略．Liu 等[69]将经过不同初始化训练的多个模型集合提取为一个模型．除了学习如何匹配集成对参考状态的概率输出外，还使用集成来探索搜索空间，并从探索中遇到的状态中学习．

目前网络蒸馏方法大多用于分类任务，但对于更加复杂的检测、分割任务也存在一定应用．目标检测：Li 等[70]将网络蒸馏的方法引入到目标检测中，提出利用区域候选框采样后的特征图进行蒸馏，使用 L2 损失减小大小网络上局部特征采样结果的差别，在检测任务上取得了一定的效果．Chen 等[71]将 hint 学习、软目标和硬目标的方法同时引入到目标检测蒸馏里面，通过对不同的损失函数赋予不同权重实现整个网络的训练．Wang 等[72]通过实验说明直接使用分类的蒸馏方式在检测中表现不足的原因，并在此基础上引入区域周边的锚点作为监督信息，避免了完全模拟时的背景噪声问题．语义分割：Liu 等[73]将相对熵损失，注意力图损失以及生成对抗网络（generative adversarial network，GAN）损失一起引入到分割任务中，在分割表现上取得了一定提升．He 等[74]提出使用自动编码器对教师模型特征进行迁移使其更容易被学生模型所学习，同时也引入了注意力的损失．

评价：基于网络蒸馏的方法可以使很深的模型变得更浅，从而帮助网络减少大量计算量．但是目前网络蒸馏主要适用在分类任务上，对于检测、分割等复杂任务应用有很大的局限性，而且学生模型的设计缺少重要指导导致学生模型很难能训成功，这些都导致网络蒸馏方法还无法成为模型压缩的主流方法．

6 紧凑网络设计

以上模型压缩方法大多是针对已经训练好的模型，通过各种手段降低其模型大小．然而紧凑网络设计是为了设计更加精简有效的网络，这是模型压缩另一个新颖的思路．近年来，紧凑网络设计取得了巨大进展，大致方向可以分为紧凑卷积结构和网络结构搜索．

紧凑卷积结构：卷积操作是深度神经网络中最重要的操作，网络的参数量与计算量占比颇高，因此很多工作都尝试对通常的卷积操作做出改进．Iandola 等[75]提出 SqueezeNet，大量使用 1×1 卷积核来替换 3×3 卷积，降低 3×3 卷积核的通道数，延迟下采样等，最终参数量比 AlexNet[2]少 50 倍但精度类似．谷歌团队 Howard 等[76-78]提出了 MobileNet 系列移动端网络模型．MobileNetv1 提出使用深度可分离卷积（depth-wise separable convolution）来替代传统卷积，极大的减少了网络的参数量；MobileNetv2 在 MobileNetv1 的基础上引入 Linear Bottlenecks 逆残差结构来实现非线性修正单元的输入信息完整性，并提出 Inverted Residuals 来提升模型的表现能力；MobileNetv3 采用了神经网络搜索的方法，网络中引入了 SENet[79]中的轻量级注意力模块，在表现与

MobileNetv2 类似的情况下速度明显提升. 旷视科技 Zhang 等[80-81]提出了 ShuffleNet 系列轻量化模型. ShuffleNetv1 采用了组卷积(group convolution)来降低模型参数量,利用通道重排(channel shuffle)来实现不同组之间的信息交互. ShuffleNetv2 探讨了网络设计时所需要考虑的问题以及设计理念,并针对 ShuffleNetv1 中设计的不足进行了改进. 此外,Xception[82]、ResNeXt[83]、ChannelNet[84] 以及 Inceptionv4[85]等等也都是在卷积设计上做了相关工作.

网络结构搜索:传统的网络模型均为人工设计,网络的结构以及参数均是人为设定,这些都导致了设计出的模型并非最优,因此神经网络结构搜索(neural architecture search, NAS)应运而生. Tan 等[86]提出了用于寻找在移动端最优的神经网络架构搜索方法 MnasNet,例如 MobileNetv3 以及谷歌大脑 Tan 与 Le[87]提出的 EfficientNet. Liu 等[88]提出了 Auto-DeepLab,在图像语义分割问题上超越了很多业内最佳模型,甚至可以在未经过预训练的情况下达到预训练模型的表现.

此外,Huang 等[89]提出根据不同难度的输入进行动态推断,实现了网络的多级输出,提高了推断的效率. Shelhamer 等[90]提出了 ClockNet 对特征进行复用,实现了更高效的推断.

评价:紧凑网络设计致力于设计出更加精简有效的网络,目前已经有很多工作在尝试. 对于未来的工业界应用,此方法是很重要的选择. 尤其是神经网络结构搜索,一旦此方法发展成熟,那么将彻底摆脱手工设计网络模型的问题,这将极大的推动工业应用的发展. 但目前来看神经网络结构搜索还有待发展,并且使用神经网络结构搜索需要很大的硬件与时间消耗,如何使用更少的资源、花费更少的时间来搜索出想要的网络是今后的重要目标.

7　存在的问题及未来研究方向

上文总结了当前用于模型压缩的几种主流方法,接下来主要讨论模型压缩存在的问题和未来研究方向.

7.1　存在的问题

(1)目前大多数最先进的方法都是建立在精心设计的卷积神经网络模型之上,这些模型对网络结构以及超参数配置有限. 为了处理更复杂任务,应该提供更合理的方法来配置压缩模型.

(2)对于工业界的应用,网络剪枝需要大量的人力以及物力的消耗,并且很难获得想要的提升;参数共享技术难度很高,容易使网络的精度下降过大,不宜使用;网络分解很难实现模型大小的明显降低,收效甚微;网络蒸馏实现起来比较困难,需要额外设计教师网络,且针对不同任务蒸馏难度也不一致,比如针对一个无人超市行人监测的任务,使用蒸馏的方法就很难取得巨大提升而对超市里的物品进行分类,蒸馏的成功率就会明显提升. 目前工业界更偏向于使用量化的方式进行模型压缩,因为目前使用半精度浮点数以及 8 位整型数对网络进行训练的技术已经成熟并可以达到商用的标准. 但对于未来的工业界应用,使用神经网络结构搜索的方法对网络进行搜索是必然的趋势,脱离手工设计是迈向真正智能的重要一步.

(3)尽管这些压缩方法取得了巨大成就,但黑盒机制仍然是采用这些方法的关键障碍. 探索知识的解释能力仍然是一个重要的问题.

7.2　未来研究方向

(1)神经网络结构搜索:对于未来的工业界应用,使用神经网络结构搜索的方法对网络进行搜索是必然的趋势,脱离手工设计是迈向真正智能的重要一步.

(2)自动模型压缩:基于自动机器学习的自动模型压缩,利用强化学习提供模型压缩策略,在模型大小、速度和准确率之间做出权衡.

(3)任务驱动的压缩:目前存在的网络都是基于存在的公开数据集进行训练与测试,但是针对具体任务以及场景,可能仅需要其中的几个子类别. 因此,如何从一个大网络压缩得到适用于具体任务的模型具有丰富的市场需求.

(4)网络可解释性的探索:可解释性一直都是卷积神经网络的重要问题,一旦此问题得到突破,那么对于基于卷积神经网络的方法都将带来巨大的提升,都将有法可寻. 因此,研究网络的可解释性将会是卷积神经网络领域的核心问题.

模型压缩是为了在保证模型精度情况下尽量的减少模型参数并尽量降低模型的计算量. 本文从网络剪枝、参数共享,量化、网络分解、网络蒸馏以及紧凑网络设计六个方向对当前模型压缩方法进行总结,并对当前模型压缩存在的问题和未来的发展方向进行阐述. 当前模型压缩技术尚未完全成熟,仍有很多方面等待挖掘与研究,希望本文可以使读者对模型压缩有比较全面的认识,并能针对具体任务加以利用.

参　考　文　献

[1]　LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015,

521(7553)：436

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks // *Advances in Neural Information Processing Systems*. Lake Tahoe, 2012：1097

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J/OL]. *ArXiv Preprint* (2015-04-10) [2019-03-22]. https://arxiv.org/abs/1409.1556

[4] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, 2015：1

[5] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Washington DC, 2016：770

[6] Huang G, Liu Z, van der Maaten L, et al. Densely connected convolutional networks // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Hawaii, 2017：4700

[7] Le Q V, Ngiam J, Coates A, et al. On optimization methods for deep learning // *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011：265

[8] Han Y F, Jiang T H, Ma Y P, et al. Compression of deep neural networks. *Comput Appl Res*, 2018, 35(10)：2894
(韩云飞, 蒋同海, 马玉鹏, 等. 深度神经网络的压缩研究. 计算机应用研究, 2018, 35(10)：2894)

[9] Setiono R, Liu H. Neural-network feature selector. *IEEE Trans Neural Networks*, 1997, 8(3)：654

[10] LeCun Y, Denker J S, Solla S A, et al. Optimal brain damage // *Advances in Neural Information Processing Systems*. Denver, 1989：598

[11] Hassibi B, Stork D G, Wolff G J. Optimal brain surgeon and general network pruning // *IEEE International Conference on Neural Networks*. San Francisco, 1993：293

[12] Hassibi B, Stork D G. Second order derivatives for network pruning：optimal brain surgeon // *Advances in Neural Information Processing Systems*. Denver, 1993：164

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

[14] Han S, Mao H, Dally W J. Deep compression：compressing deep neural networks with pruning, trained quantization and huffman coding [J/OL]. *ArXiv Preprint* (2016-02-15) [2019-03-22]. https://arxiv.org/abs/1510.00149

[15] Srinivas S, Subramanya A, Venkatesh Babu R. Training sparse neural networks // *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Hawaii, 2017：138

[16] Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks. *ACM J Emerg Technol Comput Syst*, 2017, 13(3)：32

[17] Wen W, Wu C P, Wang Y D, et al. Learning structured sparsity in deep neural networks // *Advances in Neural Information Processing Systems*. Barcelona, 2016：2074

[18] Lin S H, Ji R R, Li Y C, et al. Toward compact ConvNets via structure-sparsity regularized filter pruning. *IEEE Trans Neural Networks Learn Syst*, 2019：1.

[19] Guo Y W, Yao A B, Chen Y R. Dynamic network surgery for efficient DNNs // *Advances in Neural Information Processing Systems*. Barcelona, 2016：1379

[20] Jia H P, Xiang X S, Fan D, et al. DropPruning for model compression [J/OL]. *ArXiv Preprint* (2018-12-05) [2019-03-22]. https://arxiv.org/abs/1812.02035

[21] Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient convnets [J/OL]. *ArXiv Preprint* (2017-03-10) [2019-03-22]. https://arxiv.org/abs/1608.08710

[22] Hu H Y, Peng R, Tai Y W, et al. Network trimming：a data-driven neuron pruning approach towards efficient deep architectures [J/OL]. *arXiv preprint* (2016-07-12) [2019-03-22]. https://arxiv.org/abs/1607.03250

[23] Tian Q, Arbel T, Clark J J. Deep LDA-pruned nets for efficient facial gender classification // *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Hawaii, 2017：10

[24] Luo J H, Wu J X, Lin W Y. ThiNet：a filter level pruning method for deep neural network compression // *Proceedings of the IEEE International Conference on Computer Vision*. Venice, 2017：5058

[25] He Y, Kang G L, Dong X Y, et al. Soft filter pruning for accelerating deep convolutional neural networks [J/OL]. *ArXiv Preprint* (2018-08-21) [2019-03-22]. https://arxiv.org/abs/1808.06866

[26] He Y H, Zhang X Y, Sun J. Channel pruning for accelerating very deep neural networks [J/OL]. *ArXiv Preprint* (2017-08-21) [2019-03-22]. https://arxiv.org/abs/1707.06168

[27] Hu Y M, Sun S Y, Li J Q, et al. Multi-loss-aware channel pruning of deep networks [J/OL]. *ArXiv Preprint* (2019-02-27) [2019-03-22]. https://arxiv.org/abs/1902.10364

[28] Zhuang Z W, Tan M K, Zhuang B H, et al. Discrimination-aware channel pruning for deep neural networks [J/OL]. *ArXiv Preprint* (2019-01-14) [2019-03-22]. https://arxiv.org/abs/1810.11809

[29] He Y H, Han S. ADC：automated deep compression and acceleration with reinforcement learning [J/OL]. *ArXiv Preprint* (2019-01-16) [2019-03-22]. https://arxiv.org/abs/1802.03494v1

[30] Appuswamy R, Nayak T, Arthur J, et al. Structured convolution matrices for energy-efficient deep learning [J/OL]. *ArXiv Preprint* (2016-06-08) [2019-03-22]. https://arxiv.org/abs/1606.02407

[31] Sindhwani V, Sainath T N, Kumar S. Structured transforms for small-footprint deep learning [J/OL]. *ArXiv Preprint* (2015-10-06) [2019-03-22]. https://arxiv.org/abs/1510.01722

[32] Cheng Y, Yu F X, Feris R S, et al. An exploration of parameter redundancy in deep networks with circulant projections [J/OL]. *ArXiv Preprint* (2015-10-27) [2019-03-22]. https://arxiv.org/abs/1502.03436

[33] Chen W L, Wilson J T, Tyree S, et al. Compressing neural net-

works with the hashing trick // *Proceedings of the* 32*nd International Conference on Machine Learning*. Lille, 2015: 2285

[34] Shi L, Feng S K, Zhu Z F. Functional hashing for compressing neural networks [J/OL]. *ArXiv Preprint* (2016-05-20) [2019-03-22]. https://arxiv.org/abs/1605.06560

[35] Wu J R, Wang Y, Wu Z Y, et al. Deep k-Means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions [J/OL]. *ArXiv Preprint* (2018-06-24) [2019-03-22]. https://arxiv.org/abs/1806.09228

[36] Lu Z Y, Sindhwani V, Sainath T N. Learning compact recurrent neural networks [J/OL]. *ArXiv Preprint* (2016-04-09) [2019-03-22]. https://arxiv.org/abs/1604.02594

[37] Jin X J, Yang Y Z, Xu N, et al. WSNet: compact and efficient networks through weight sampling [J/OL]. *ArXiv Preprint* (2018-05-22) [2019-03-22]. https://arxiv.org/abs/1711.10067

[38] Yang Y Z, Jojic N, Huan J. FSNet: Compression of deep convolutional neural networks by filter summary [J/OL]. *ArXiv Preprint* (2019-02-13) [2019-03-22]. https://arxiv.org/abs/1902.03264

[39] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision [J/OL]. *ArXiv Preprint* (2015-02-09) [2019-03-22]. https://arxiv.org/abs/1502.02551

[40] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, 2018: 2704

[41] Courbariaux M, Bengio Y, David J P. BinaryConnect: training deep neural networks with binary weights during propagations // *Advances in Neural Information Processing Systems*. Canada, 2015: 3123

[42] Courbariaux M, Hubara I, Soudry D, et al. Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1 [J/OL]. *ArXiv Preprint* (2016-03-17) [2019-03-22]. https://arxiv.org/abs/1602.02830

[43] Rastegari M, Ordonez V, Redmon J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks [J/OL]. *ArXiv Preprint* (2016-08-02) [2019-03-22]. https://arxiv.org/abs/1603.05279

[44] Li Z F, Ni B B, Zhang W J, et al. Performance guaranteed network acceleration via high-order residual quantization // *Proceedings of the IEEE International Conference on Computer Vision*. Venice, 2017: 2584

[45] Hwang K, Sung W. Fixed-point feedforward deep neural network design using weights + 1, 0, and -1 // 2014 *IEEE Workshop on Signal Processing Systems* (*SiPS*). Belfast, 2014: 1

[46] Hou L, Yao Q M, Kwok J T. Loss-aware binarization of deep networks [J/OL]. *ArXiv Preprint* (2018-05-10) [2019-03-22]. https://arxiv.org/abs/1611.01600

[47] Lee H, Battle A, Raina R, et al. Efficient sparse coding algorithms // *Advances in Neural Information Processing Systems*. Canada, 2007

[48] Gudovskiy D A, Rigazio L. ShiftCNN: generalized low-precision architecture for inference of convolutional neural networks [J/OL]. *ArXiv Preprint* (2017-06-07) [2019-03-22]. https://arxiv.org/abs/1706.02393

[49] Meller E, Finkelstein A, Almog U, et al. Same, same but different-recovering neural network quantization error through weight factorization [J/OL]. *ArXiv Preprint* (2019-02-05) [2019-03-22]. https://arxiv.org/abs/1902.01917

[50] Xu Y H, Zhang S, Qi Y Y, et al. DNQ: Dynamic network quantization [J/OL]. *ArXiv Preprint* (2018-12-06) [2019-03-22]. https://arxiv.org/abs/1812.02375

[51] Golub G H, Reinsch C. Singular value decomposition and least squares solutions // *Linear Algebra*. Springer, Berlin, 1971: 134

[52] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions [J/OL]. *ArXiv Preprint* (2014-05-15) [2019-03-22]. https://arxiv.org/abs/1405.3866

[53] Kim Y D, Park E, Yoo S, et al. Compression of deep convolutional neural networks for fast and low power mobile applications [J/OL]. *ArXiv Preprint* (2016-02-24) [2019-03-22]. https://arxiv.org/abs/1511.06530

[54] Denil M, Shakibi B, Dinh L, et al. Predicting parameters in deep learning // *Advances in Neural Information Processing Systems*. Lake Tahoe, 2013: 2148

[55] Calvi G G, Moniri A, Mahfouz M, et al. Tucker tensor layer in fully connected neural networks [J/OL]. *ArXiv Preprint* (2019-03-14) [2019-03-22]. https://arxiv.org/abs/1903.06133

[56] Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression // *Proceedings of the* 12*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, 2006: 535

[57] Ba J, Caruana R. Do deep nets really need to be deep? // *Advances in Neural Information Processing Systems*. Canada, 2014: 2654

[58] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J/OL]. *ArXiv Preprint* (2015-03-09) [2019-03-22]. https://arxiv.org/abs/1503.02531

[59] Romero A, Ballas N, Kahou S E, et al. FitNets: hints for thin deep nets[J/PL]. *ArXiv Preprint* (2015-03-27) [2019-03-22]. https://arxiv.org/abs/1412.6550

[60] Chen T Q, Goodfellow I, Shlens J. Net2Net: accelerating learning via knowledge transfer[J/OL]. *ArXiv Preprint* (2016-04-23) [2019-03-22]. https://arxiv.org/abs/1511.05641

[61] Li Z Z, Hoiem D. Learning without forgetting. *IEEE Trans Pattern Anal Mach Intell*, 2018, 40(12): 2935

[62] Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer [J/OL]. *ArXiv Preprint* (2017-02-12) [2019-03-22]. https://arxiv.org/abs/1612.03928

[63] Mirzadeh S I, Farajtabar M, Li A, et al. Improved knowledge distillation via teacher assistant: bridging the gap between student and teacher [J/OL]. *ArXiv Preprint* (2019-02-09) [2019-03-

22]. https://arxiv.org/abs/1902.03393

[64] Liu X, Wang X G, Matwin S. Improving the interpretability of deep neural networks with knowledge distillation [J/OL]. *ArXiv Preprint* (2018-12-28) [2019-03-22]. https://arxiv.org/abs/1812.10924

[65] Yang C L, Xie L X, Su C, et al. Snapshot distillation: Teacher-student optimization in one generation [J/OL]. *ArXiv Preprint* (2018-12-01) [2019-03-22]. https://arxiv.org/abs/1812.00123

[66] Wang J, Bao W D, Sun L C, et al. Private model compression via knowledge distillation [J/OL]. *ArXiv Preprint* (2018-11-13) [2019-03-22]. https://arxiv.org/abs/1811.05072

[67] Lee S H, Kim D H, Song B C. Self-supervised knowledge distillation using singular value decomposition [J/OL]. *ArXiv Preprint* (2018-07-18) [2019-03-22]. https://arxiv.org/abs/1807.06819

[68] Lan X, Zhu X T, Gong S G. Knowledge distillation by on-the-fly native ensemble [J/OL]. *ArXiv Preprint* (2018-09-08) [2019-03-22]. https://arxiv.org/abs/1806.04606

[69] Liu Y J, Che W X, Zhao H P, et al. Distilling knowledge for search-based structured prediction [J/OL]. *ArXiv Preprint* (2018-05-29) [2019-03-22]. https://arxiv.org/abs/1805.11224

[70] Li Q Q, Jin S Y, Yan J J. Mimicking very efficient network for object detection // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, 2017: 6356

[71] Chen G B, Choi W, Yu X, et al. Learning efficient object detection models with knowledge distillation // *Advances in Neural Information Processing Systems*. Long Beach, 2017: 742

[72] Wang T, Yuan L, Zhang X P, et al. Distilling object detectors with fine-grained feature imitation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, 2019: 4933

[73] Liu Y F, Chen K, Liu C, et al. Structured knowledge distillation for semantic segmentation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, 2019: 2604

[74] He T, Shen C H, Tian Z, et al. Knowledge adaptation for efficient semantic segmentation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, 2019: 578

[75] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size [J/OL]. *ArXiv Preprint* (2016-11-04) [2019-03-22]. https://arxiv.org/abs/1602.07360

[76] Haward A G, Zhu M L, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications [J/OL]. *ArXiv Preprint* (2017-04-17) [2019-03-22]. https://arxiv.org/abs/1704.04861

[77] Sandler M, Haward A, Zhu M L, et al. MobileNetV2: inverted residuals and linear bottlenecks // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, 2018: 4510

[78] Haward A, Sandler M, Chu G, et al. Searching for MobileNetV3 [J/OL]. *ArXiv Preprint* (2019-08-24) [2019-10-10]. https://arxiv.org/abs/1905.02244

[79] Hu J, Shen L, Sun G. Squeeze-and-excitation networks // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, 2018: 7132

[80] Zhang X Y, Zhou X Y, Lin M X, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, 2018: 6848

[81] Ma N N, Zhang X Y, Zheng H T, et al. ShuffleNet V2: practical guidelines for efficient CNN architecture design // *Proceedings of the European Conference on Computer Vision*. Munich, 2018: 116

[82] Chollet F. Xception: Deep learning with depthwise separable convolutions // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, 2017: 1251

[83] Xie S N, Girshick R, Dollar P, et al. Aggregated residual transformations for deep neural networks // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, 2017: 1492

[84] Gao H Y, Wang Z Y, Ji S W. ChannelNets: Compact and efficient convolutional neural networks via channel-wise convolutions // *Advances in Neural Information Processing Systems*. Salt Lake City, 2018: 5197

[85] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the impact of residual connections on learning [J/OL]. *ArXiv Preprint* (2016-08-23) [2019-03-22]. https://arxiv.org/abs/1602.07261

[86] Tan M X, Chen B, Pang R M, et al. MnasNet: Platform-aware architecture search for mobile // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, 2019: 2820

[87] Tan M X, Le Q V. EfficientNet: Rethinking model scaling for convolutional neural networks [J/OL]. *ArXiv Preprint* (2019-06-10) [2019-10-10]. https://arxiv.org/abs/1905.11946

[88] Liu C X, Chen L C, Schroff F, et al. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach, 2019: 82

[89] Huang G, Chen D L, Li T H, et al. Multi-scale dense networks for resource efficient image classification [J/OL]. *ArXiv Preprint* (2018-06-07) [2019-03-22]. https://arxiv.org/abs/1703.09844

[90] Shelhamer E, Rakelly K, Hoffman J, et al. Clockwork convnets for video semantic segmentation [J/OL]. *ArXiv Preprint* (2016-08-11) [2019-03-22]. https://arxiv.org/abs/1608.03609