计算机科学
COMPUTER SCIENCE

神经网络模型轻量化方法综述

高杨，曹仰杰，段鹏松

引用本文

高杨，曹仰杰，段鹏松. 神经网络模型轻量化方法综述[J]. 计算机科学，2024，51(6A)：230600137-11.

GAO Yang, CAO Yangjie, DUAN Pengsong. Lightweighting Methods for Neural Network Models:A Review [J]. Computer Science, 2024, 51(6A): 230600137-11.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230600137

神经网络模型轻量化方法综述

高 杨 曹仰杰 段鹏松
郑州大学网络空间安全学院 郑州 450000
(510910342@qq.com)

摘 要 近年来,神经网络模型凭借着较强的特征提取能力在各行各业的应用越来越广泛,并取得了不错的效果。然而,随着数据量的不断增大以及人们对高准确率的不断追求,神经网络模型的参数规模急剧增大,网络复杂度不断提高,导致计算、存储等资源开销不断扩大,使其在资源受限场景下的部署面临极大挑战。因此,如何在不影响模型性能的前提下实现模型轻量化,进而降低模型训练和部署的成本成为当前的研究热点之一。为此,文中从复杂模型压缩以及轻量化模型设计两方面入手,对当前典型的模型轻量化方法进行总结和分析,以期厘清模型压缩技术的发展脉络。其中,复杂模型压缩技术从模型剪枝、模型量化、低秩分解、知识蒸馏及混合方式5方面进行归纳,而轻量化模型设计则从空间卷积设计、移位卷积设计和NAS架构搜索3方面进行梳理。

关键词:神经网络;模型压缩;模型剪枝;模型量化;模型轻量化
中图分类号 TP183

Lightweighting Methods for Neural Network Models:A Review

GAO Yang,CAO Yangjie and DUAN Pengsong
School of Cyberspace Security,Zhengzhou University,Zhengzhou 450000,China

Abstract In recent years,with its strong feature extraction capability,neural network models have been more and more widely used in various industries and have achieved good results. However,with the increasing amount of data and the pursuit of high accuracy,the parameter size and network complexity of neural network models increase dramatically,leading to the expansion of computation,storage and other resource overheads,making their deployment in resource-constrained scenarios extremely challenging. Therefore,how to achieve model lightweighting without affecting model performance,and thus reduce model training and deployment costs,has become one of the current research hotspots. This paper summarizes and analyzes the typical model lightweighting methods from two aspects:complex model compression and lightweight model design,so as to clarify the development of model compression technology. The complex model compression techniques are summarized in five aspects:model pruning,model quantization,low-rank decomposition,knowledge distillation and hybrid approach,while the lightweight model design is sorted out in three aspects:spatial convolution design,shifted convolution design and neural architecture search.

Keywords Neural networks,Model compression,Model pruning,Model quantization,Model lightweight

1 引言

作为目前被广泛应用的一类网络结构,神经网络自20世纪40年代被提出以来,由于计算能力受限、缺乏优化手段等一直发展缓慢。直到1989年,LeCun教授提出的卷积神经网络模型在手写字体识别中取得了较好效果,才使神经网络模型得到了一定关注。此后,神经网络模型开始被广泛应用于各个领域,在某些任务中甚至表现出了超越人类的识别能力,神经网络模型研究由此步入高潮。目前,凭借强大的算力及完备的数据集支持,基于深度学习的神经网络模型能够完成的任务越来越复杂,模型规模越来越大。

随着物联网时代的到来,模型部署已不局限于服务器端,越来越多的边缘设备也有较强的AI应用需求。但是,边缘设备的计算和存储资源有限,运行和训练复杂度较高的神经网络模型存在诸多限制。因此,如何在保持神经网络模型有效识别精度的同时实现其轻量化,成为AI领域的一个重要研究方向。

目前关于模型轻量化的综述论文较少,也缺少相关内容的最新跟踪。为此,本文梳理了近年来模型轻量化相关的经典方法及未来研究方向,以帮助研究者快速了解模型轻量化

基金项目:郑州市协同创新重大专项(20XTZX06013);河南省高等学校重点科研项目计划(21A520043);中国工程科技发展战略河南研究院战略咨询研究项目(2022HENYB03);河南省科技攻关项目(232102210050)
This work was supported by the Collaborative Innovation Major Project of Zhengzhou(20XTZX06013),Research Foundation Plan in Higher Education Institutions of Henan Province (21A520043), Strategic Research and Consulting Project of Chinese Academy of Engineering (2022HENYB03) and Science and Technology Project of Henan Province(232102210050).
通信作者:段鹏松(duanps@163.com)

领域的最新进展，如表 1 所列。

表 1 2015－2023 年相关综述论文的参考文献年限分布总结

Table 1 Summary of the distribution of review papers from 2015 to 2023

综述论文来源	2015－2019 年文献数量	2020－2023 年文献数量
深度学习模型压缩与加速综述[1]	157	0
深度神经网络压缩方法综述[2]	65	10
深度神经网络模型压缩综述[3]	83	0
紧凑的神经网络模型设计研究综述[4]	60	6

2 技术背景

神经网络模型的发展可以追溯到 20 世纪 50 年代。早期的神经网络模型包括感知机和递归神经网络等，但由于计算资源的限制和模型设计方法的缺失，这些模型的应用范围十分有限。近年来，随着算力的提升及算法的优化，深度神经网络在图像分类[5]、自然语言处理[6]、语音识别[7]等多个领域取得了显著成果，成为当前主要研究热点之一。神经网络模型是一种由多个神经元组成的数学模型，用于模拟人脑中神经元的运作方式。早期的神经网络模型发展缓慢，Hinton 等于 1986 年提出了多层感知机模型（Multi-Layer Perceptron，MLP），标志着神经网络模型新时代的开端。其中，卷积神经网络（CNN）作为一种应用非常广泛的神经网络模型，在图像分类、目标检测、人脸识别等领域都取得了重要的成果。CNN 主要由卷积层、池化层和全连接层组成，可以有效地提取图像中的空间特征。此外，循环神经网络（RNN）和长短时记忆网络（LSTM）等也是常用的神经网络模型，在自然语言处理和语音识别等领域得到了广泛应用。

然而，由于神经网络模型具有复杂性，因此它们需要庞大的参数规模和计算资源才能进行训练和推理。那么如何在一些资源受限的边缘设备上进行部署成了一个严重的问题。为解决这个问题，模型轻量化技术应运而生。模型轻量化技术通过减小神经网络模型的规模和降低计算复杂度，在资源受限的边缘设备上实现高效的推理和部署。因此，模型轻量化已成为神经网络模型研究领域中备受关注的话题之一。

近年来，研究人员提出了许多模型轻量化的方法，包括模型压缩[8]、模型量化[9]、知识蒸馏[10]、网络剪枝[11]等。这些方法已经被广泛应用于各类神经网络模型中，帮助模型减小参数规模和降低计算复杂度，从而完成高效的分类和预测任务。此外，随着移动设备和嵌入式系统的快速发展，模型轻量化技术也成为实现神经网络模型在移动设备和嵌入式系统上部署的关键技术之一。

本文从模型是否进行预训练的角度将模型轻量化方法分为复杂模型压缩方法和轻量化模型设计方法两类。其中，复杂模型压缩方法通常是对预训练好的模型进行压缩，从而减小模型的参数规模和降低计算复杂度；轻量化模型设计方法则是不经过预训练，直接设计一个精简的、轻量化的神经网络模型。

3 复杂模型压缩方法

复杂模型压缩指通过消除神经网络模型中参数或者结构的冗余来精简模型规模。通过模型压缩可以在尽量不影响模型性能的前提下得到一个数据量更少、结构更精简、更易于部署的模型。被压缩后的模型计算资源需求和存储需求更小，训练速度更快，能更好满足实际应用的需求。常用的模型压缩方法有模型剪枝、模型量化等，如图 1 所示。

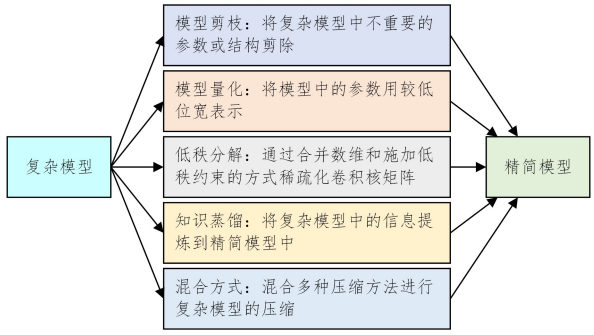

图 1 复杂模型压缩方法分类

Fig. 1 Classification of complex model compression methods

3.1 模型剪枝

模型剪枝是针对预训练模型进行的一种轻量化处理方法，通过删减不重要的参数或结构来实现。该方法旨在最大限度地减小模型的规模，同时尽量不影响模型性能，实现对模型的压缩和加速。目前，模型剪枝从剪枝对象的不同可以分为参数剪枝和结构化剪枝两个大类。

3.1.1 参数剪枝

参数剪枝主要关注模型中的参数，通过识别和删除对模型性能影响较小的参数来减小模型的规模。这种方法通常基于参数的重要性指标进行选择，例如参数的绝对值大小或其对损失函数的贡献。剪枝后，被删除的参数将不再参与模型的计算，从而减小了计算资源和存储需求。如 Hanson 等[12]提出的基于参数的幅值进行剪枝的模型压缩方法，该方法通过对网络中每个隐藏层神经元施加与其重要性相关的权重衰减来优化神经网络模型。相比该剪枝方法，OBD 算法[13]和 OBS 算法[14]通过基于损失函数的 Hessian 矩阵来衡量网络中参数的重要性，从而对模型参数进行剪枝以压缩模型，展现出了更好的剪枝效果、更高的压缩比率以及更少的性能损失。2015 年，Han 等[15]为了解决传统网络的不足，提出了一种在尽量保持神经网络性能的情况下只修剪网络中不重要连接的方法，该方法首次给出了三阶段剪枝流程，如图 2 所示。

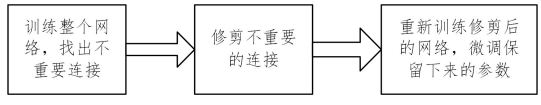

图 2 三阶段剪枝流程

Fig. 2 Three-stage pruning process

该剪枝方法先对神经网络进行训练，然后通过增加正则化训练神经网络快速找出重要连接并剪除。Han 等按照三阶段的剪枝流程，提出根据神经元连接权值的正则化范数值大小衡量重要性的剪枝方法，该方法删除不重要连接后再重新训练从而恢复部分精度，完成剪枝。与之类似，Dong 等[16]提出了一种基于损失函数对应参数的二阶导数进行分层剪枝的方法，该方法先逐层进行 OBS 算法，剪枝后经过简单的再训练以恢复性能。上述方法都是对整个神经网络中的连接进行重要性评估再剪枝，如图 3 所示，除此之外也可以直接对神经元进行评估再剪枝。Mariet 等[17]提出通过 DPP（Determinantal Point Process）方法来模拟神经元可以更有原则、更灵

活地进行神经元重要性的评估。该方法帮助网络结构有效地自动调整而不影响性能，不需要再微调模型，从而节省了时间成本。Kingma 等[18]提出了变分 Dropout 方法来避免神经网络训练过程中的过拟合。受此启发，Molchanov 等[19]将其用于模型压缩，使得卷积层和全连接层的连接变得稀疏。实验结果表明，该方法使得 LeNet 网络减少了 280 倍的参数量，在 VGG 网络上的参数量减少为原来的 1/68，而性能的损失基本可以忽略不计。传统的模型剪枝需要大量数据来训练模型，而这在实际应用中不利于保护用户的隐私安全。因此，Srinivas 等[20]定义了另一种神经元冗余，并且提出了一种不依赖训练数据直接剪枝的方法。然而参数剪枝往往会将滤波器中的元素置 0，从而指定了一个固定的子空间来约束滤波器，特别是在训练前或者训练中进行剪枝往往会产生很大的误差。对此，Wimmer 等[21]提出了一种改进现有剪枝方法的通用工具——空间剪枝。空间剪枝使用在底层自适应的滤波器基础的线性组合在动态空间表示滤波器，并将未剪枝的参数和滤波器基础进行联合训练。经过实验证明，该方法优于现在所有的参数剪枝方法，这得益于其改进的可训练性和卓越的泛化性能。

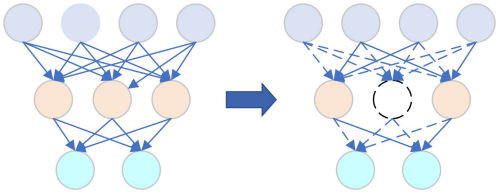

图 3　裁剪连接

Fig. 3　Pruning connections

参数剪枝操作简单，但也存在一些问题。参数剪枝方法通常基于参数重要性来选择要进行裁剪的参数。然而，这种裁剪可能会导致某些卷积层中的部分滤波器被删除，从而破坏原始的卷积层结构。同时这也会破坏卷积操作中的数据局部性和权重共享特性，进而降低卷积运算的效率。

3.1.2　结构化剪枝

结构化剪枝专注于删除整个参数组或层级结构，以进一步减小模型的规模。常见的结构化剪枝方法包括通道剪枝和层剪枝。通道剪枝通过删除卷积层中的冗余通道来减小模型的规模，层剪枝则通过删除不重要的层或网络块来实现模型的轻量化。相比参数剪枝，结构化剪枝后的结构更加规整，因此在软硬件层面可以获得更有效的加速。但是大范围的剪枝结构也会带来精度大幅度下降的问题，因此需要最后对模型进行微调来恢复性能。Zhuang 等[22]介绍了 network slimming 方法，通过给网络模型中的每个通道加上一个新的参数并将其作为缩放因子来进行结构化剪枝。他们在进行模型训练时采用缩放因子和网络权重进行联合训练，并将该方法中的目标函数定义为：

$$L=\sum_{(x,y)} l(f(x,W),y)+\lambda\sum_{\gamma\in\Gamma} g(\gamma) \tag{1}$$

其中，(x,y) 分别是训练输入和目标，W 是网络中可训练参数，第一个多项式是 CNN 网络的训练损失函数，γ 是缩放系数，在训练过程中与通道相乘从而完成对通道的筛选，γ 越小则通道越不重要。而 $g(\cdot)$ 是在缩放因子上的惩罚项，λ 作为平衡因子，可以看作是对后边项进行了归一化处理。在训练的过程中将 BN 层和缩放因子 γ 结合，BN 层执行式(2)所示的转换：

$$\hat{z}=\frac{z_{\text{in}}-\mu_\beta}{\sqrt{\sigma_\beta^2+\varepsilon}};z_{\text{out}}=\gamma\hat{z}+\beta \tag{2}$$

其中，z_{in} 和 z_{out} 分别作为 BN 层的输入和输出，γ 作为缩放因子对神经网络模型进行剪枝，以此通过 L1 正则化对该权重进行稀疏，方便对权重小于阈值的通道进行剪枝，最后再训练得到一个精简的网络模型。对于 VGG 网络，剪枝后的模型参数量变为原来的 1/20，运算操作数量变为原来的 1/5，大大精简了原来的复杂神经网络。

结构化剪枝解决了参数剪枝结构不规整的问题，而且无需硬件的支持便能进行有效的加速，然而由于结构化剪枝采用固定且不变的剪枝策略，其适应性较低，难以推广应用到各种场景。这种固定的剪枝策略可能无法适应不同模型结构和任务的特点，还可能会导致剪枝过程中丢失一些重要的参数或结构，从而影响模型的性能和泛化能力，无法根据不同数据集或应用场景的需求进行个性化调整。

2.1.3　其他

随着研究方法的不断进步，除了结构化剪枝和参数剪枝外，还涌现了很多新颖的剪枝方式，其中就包括动态剪枝。与固定剪枝相比，动态剪枝的剪枝部位会随着训练过程参数的变化而发生调整。通过动态剪枝，模型可以适应训练数据和模型结构的变化动态调整，更好地适应不同的任务和数据特征，从而达到更好的模型压缩和性能优化。

Guo 等[23]提出了根据训练过程中权重的变化动态调整标签的动态剪枝(Dynamic Network Surgery，DNS)方法，该方法依靠拼接避免不正确的剪枝，从而完成对模型的连续维护。Ji 等[24]提出了一个新的剪枝框架——Runtime Neural Pruning(RNP)，将剪枝建模为马尔可夫决策过程，并通过强化学习来学习剪枝策略。该框架不再探索静态的剪枝策略，而是保留复杂网络的全部表达能力动态地去选择路径，为模型剪枝提供了一种新思路。实验结果表明，该框架在 ImageNet 数据集上用 VGG-16 模型加速 10 倍还是能保持较好图片识别效果。Liu 等[25]借鉴了 DNS 方法，提出了一种频域动态剪枝方案，将空间域的卷积转换为频域的乘法，然后进行参数剪枝以压缩模型。Gao 等[26]提出利用卷积层计算出的特征与输入数据高度相关这一事实来降低剪枝的成本，而不像静态剪枝一样剪除通道，提出的特征提升和抑制策略(Feature Boosting and Suppression，FBS)通过预测下一层卷积通道的重要性，在模型训练时跳过不重要的通道来加速训练。该方法有效实现了计算量的减少和训练的加速，在 VGG-16 和 ResNet-18 数据集上的计算量分别减少为原来的 1/5 和 1/2，并在 top-5 accuracy 上只损失了 0.6% 的精度。动态剪枝加速效果出色，但为了实现高灵活性，通常会存储所有的权重参数，因此不能像传统剪枝那样节省存储空间。而 Chen 等[27]提出基于深度强化学习来找到每一层最合适的剪枝阈值的动态剪枝方法。该方法还加入了“静态”的部分，就是彻底删去一些非常不重要的通道来减少参数的存储量，减小了动态剪枝的存储空间需求。为了实现动态高灵活性和静态存储效率之间的平衡，他们还提出了两种对数据重要性进行评估的方法：runtime importance 方法和 static importance 方法。前者用来考察不同输入对每个通道的重要性，后者考察每个通道对所有输入数据的重要性。然而在动态剪枝过程中，由于

索引、权重复制以及零掩码等额外负担，卷积滤波器上的动态剪枝往往不能起到加速作用。对此，Li 等[28]提出了一种新型的动态剪枝策略，并提出了一个动态瘦身网络。该方法通过测试时动态调整滤波器的数量来达到良好的硬件效率，并通过动态宽度门控来对网络进行瘦身，在 ResNet-50 和 MobileNetV1 中都取得了很好的加速效果。

尽管动态剪枝方法具有灵活性和自适应性的优势，但也面临着一些挑战。动态剪枝需要更复杂的剪枝算法和实时的参数更新机制，以确保在训练过程中剪枝部位的准确性和稳定性。此外，动态剪枝方法的计算和存储开销可能较高，需要权衡剪枝效果和计算资源的利用率。

3.2 模型量化

模型量化利用较低位宽整型数据来表示模型中较高位宽的浮点型参数，包括权重、激活值、梯度和误差等，进而实现模型的压缩。运用较低位宽数据能够显著地减小模型存储空间、提高运行速度以及降低读取数据的时间成本。如将 32 位浮点型数据量化成 8 位整型数据可以节省约 75%的存储空间，整型数的运算也比浮点数更加快速。虽然模型量化显著减小了模型规模，提高了运算速度，但也存在一定问题。数据位宽的减少会导致神经网络丢失部分信息，从而带来精度损失。虽然通过微调可以恢复部分精度，但这又带来了时间成本的增加。而且随着硬件加速部件的不断推出，一些特殊位宽不再适用于现有的训练方法以及部署的硬件平台。

目前，模型量化方法从量化映射的均匀与否可以分为两个大类，分别是均匀量化和非均匀量化，又称线性量化和非线性量化。量化的过程可以理解为一个实现高精度参数向低精度参数映射的函数，均匀量化映射的间隔都是相同的，非均匀量化映射的间隔不同。而根据量化精度以及方式的不同，均匀量化主要分为极低精度量化、聚类量化、INT8 量化以及混合精度量化。

3.2.1 极低精度量化

极低精度量化的目的是使网络的权重和激活值等参数得到极限压缩，以便最大限度地压缩模型的规模。本文将低于 8 bit 的量化称为极低精度量化，主要包括二值化和三值化量化。通过使用二值化和三值化量化方法，可以将权重和激活值等参数从浮点数形式压缩为极低精度的形式，大大减小模型的存储空间和计算需求。二值化量化将参数量化为两个离散值(通常为－1 和＋1)，而 3 值化量化将参数量化为 3 个离散值(通常为－1，0 和＋1)。这种极低精度量化可以在一定程度上保持模型的性能，并提供高度压缩的模型规模，有利于在资源受限的环境中部署深度学习模型。

Courbariaux 等[29]提出了 Binary Connect 方法将权重量化到 1 或者－1。他们在梯度运算的过程中不对权重进行操作，而是选择在反向传播和前向传播中对其进行量化。二值化网络[30](Binarized Neural Networks，BNN)对激活值和权重进行二值化就受到了该方法的启发。通过轻量级 XNOR 操作代替浮点矩阵乘法的方法，极大减少了计算带来的延迟。Rastegari 等[31]在文献[30]的基础上提出了 XNOR-Net 网络，将卷积分为 XNOR 操作和位操作两部分来从头训练一个二值化网络。

可是仅使用＋1 和－1 进行二值化量化存在精度方面的不足，为了减少精度损失，引入 0 来进行三值化量化。Li 等[32]提出了一个 Ternary Weight Networks(TWN)方法，将网络的权值限制为{－1，0，＋1}。Zhu 等[33]在前人研究的基础上提出了 TTQ 方法，该方法通过训练正负两个尺度因子，使尺度因子不对称，从而使模型性能更强；针对神经网络所有层提出变量 r，调整 r 可以改变量化阈值获得不同稀疏度的三值网络。2017 年，Zhou 等[34]在三值量化的基础上提出了一种渐进量化方法，该方法先分组进行量化，然后冻结已量化的部分并训练未量化的部分，多次重复以上步骤直到所有权重都被量化，这种渐进式量化的方法可以把一个预训练的复杂模型近乎无损地压缩成一个精简模型。通过实验可以发现，该方法在 AlexNet，VGG，ResNet 等模型上都表现良好。Faraone 等[35]提出了基于梯度的对称量化方法 SYQ，该方法在 3 个不同粒度(pixel-wise，row-wise 和 layer-wise)定义缩放因子，以估计网络参数并设计网络模型，每一层的激活输出则采用线性量化表示为 2～8 bits 定点数值。Leng 等[36]在 AAAI2018 上发表的论文提出将极低精度量化建模成一个离散约束优化问题。借助于交替方向乘子法(Alternating Direction Method of Multipliers，ADMM)思想，该方法将连续参数从网络的离散约束中分离出来，并针对过程中出现的问题分别采用了 extragradient 以及 iterative quantization 算法来解决。

3.2.2 聚类量化

除了极低精度量化，在面对庞大的参数数量时，往往会采用聚类量化的方法。Gong 等[37]最先提出将 k-means 聚类用于参数量化，过程如图 4 所示。首先对权重聚类形成码本，为权值分配码本中的索引，这样只需精简存储码本和索引而无需存储复杂的原始权重信息。Xu 等[38]参考 k-means 聚类的思想并结合分层量化、共享权值以及同一层分块量化设计出了 SLQ 方法，利用参数的分布改善分块量化的精度，并在分层量化中通过增量层次量化的方法来补偿前面层数的量化损失。

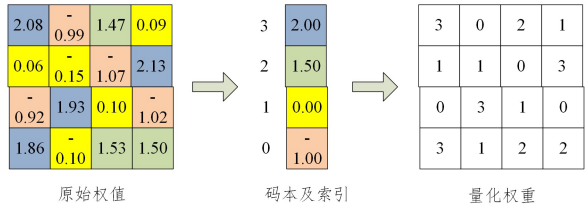

图 4 聚类量化流程图

Fig. 4 Flow chart of clustering quantization

3.2.3 INT8 量化

INT8 量化就是将 32 位的浮点型数据量化为 8 位的整型数据，因其实现方法简单，压缩效果出色，所以是目前量化领域最常用的方法。其中 INT 代表整型，8 代表 8 位，量化过程中浮点型参数通过量化操作映射到有限的整型范围内，通常是－128 到 127 之间的整数。

Dettmers 等[39]最早开发并测试 8 bit 量化算法，将 32 bit 的权重和激活值压缩到 8 bit，在 MNIST，CIFAR-10 和 ImageNet 数据集上进行了测试，发现 8bit 量化算法不仅没有降低预测性能而且相比 32 位并行数据传输速度提高了 2 倍。Wang 等[40]提出了一种根据硬件反馈来调整量化策略的 HAQ 量化方法。该方法利用强化学习方法将权值和激活值的位宽作为输出动作行为，将硬件加速器作为环境，将短期重

训练后的精度提升作为奖励信号；与传统量化方法相比，所提框架是完全自动化的，可以针对不同的神经网络架构和硬件架构进行专门的量化策略。实验结果表明，所提框架相比其他 8 bit 量化有效地将延迟减少为原来的 51%～71%，能耗减少了 48%，并且精度损失可忽略不计。

INT8 量化虽然应用广泛，但目前需要追求极致压缩效果，其固定不变的压缩比率已经逐渐无法满足人们的需求。

3.2.4　非均匀量化

均匀量化方法实现简单，效果明显，但更低位宽的量化会带来无法弥补的精度损失，因此在实际应用中往往只会采用 8 bit 量化。为有效解决该问题，非均匀量化方法也被提出，如 PWLQ 和 APoT 等。在非均匀量化中，对模型参数进行量化的步长（即间隔）不是固定的，而是根据参数的分布情况进行动态调整，从而更好地平衡模型的精度和存储/计算成本。

来自三星团队的 Oral Presentation 团队[41]介绍了一种两段式量化的模型压缩方法 Post-training Piecewise Linear Quantization（PWLQ），其具体操作便是根据模型中数据分布的稀疏与密集程度来将数据划分成两段，分布稀疏的数据采用 4 bit 的量化，分布密集的数据则采用 8 bit 的量化。与均匀量化相比，分段量化有着明显更高的精度，尤其在 4 bit-Mobilnet-v2 的对比上，PWLQ 有高出 27.42% 的精度。Li 等[42]在经典的二次幂量化方法的基础上进行创新，提出了可加性二次幂量化策略，即 APoT 的非均匀量化方法。此量化的标准相比两段式量化更符合权重值和激活值的长尾钟型分布情况，综合考虑了计算上的有效性以及低比特量化导致的模型精度下降问题。在 ImageNet 上的 3 bit 量化 ResNet-34 仅下降了 0.3% 的 Top-1 和 0.2% 的 Top-5 的准确性，与统一量化相比，此量化方案使模型降低了 22% 的计算成本。来自冲电气工业株式会社的 Yamamoto 提出了一种可学习的压缩拓展量化，改进了 APoT 中的非均匀量化技术，从而使得量化训练更加稳定[43]。该方法在量化过程中引入额外的可学习变量，更进一步减少低比特量化带来的精度损失，并在 CIFAR-10，ImageNet-1k 和 COCO 等数据集上都取得了不错的压缩效果。然而这些量化方法运用在大型数据集中往往会造成沉重的搜索成本。为解决该问题，Wang 等[44]提出了一种用于推理的可通用混合精度量化方法，通过有效的容量感知属性模仿来保持量化模型和其全精度模型之间的属性等级一致性，以进行可推广的混合精度量化策略搜索。通过大量的实验可以证明，该方法与最先进的混合精度量化相比有着更低的搜索成本以及更有竞争力的精度与复杂性权衡。

在实际应用中，非均匀量化也会面对数据缺失和数据安全等问题，为解决该问题，Cai 等[45]提出了一种训练合成数据的后量化方法，提出了蒸馏数据的方法，通过解决一个蒸馏优化问题来学习神经网络模型中 BN 层的统计信息，寻找到能最好匹配它的输入数据分布。最后通过这些合成数据能够有效分辨出每个网络层对量化的敏感度，从而通过帕累托边界的方法自动选择不同网络层的位宽配置，完成混合精度量化。在学习合成数据的基础上，Zhong 等[46]观察到了真实数据的类内异质性，而 Cai 等[45]提出的方法无法在合成数据中保留这一属性。对此，Zhong 等[46]提出了一种新型零样本量化方法，通过局部对象增强，将目标对象定位在合成对象的不同比例和位置，然后引入边缘距离约束来形成区域内的类相关特征，实验证明该方法很好地保留了合成图像中的类内异质性。

3.3　低秩分解

在神经网络模型中，过滤器可以看作是一个由宽、高、通道数以及卷积核数等属性组成的四维张量。通道数与卷积核数对模型结构的整体影响较大，因此低秩分解主要根据卷积核冗余的特点来进行模型的压缩。通过合并数维和施加低秩约束的方式可以稀疏化卷积核矩阵[47]。由于权值向量大多分布在低秩子空间，因此可以用低存储量的基础向量来重构卷积核矩阵，从而达到缩小存储空间的目的。

Jaderberg 等[48]将秩为 1 的卷积核作用在输入图上来产生相互独立的 M 个基本特征图，通过将卷积神经网络中大小为 $k\times k$ 的卷积核分解为 $1\times k$ 和 $k\times 1$ 的卷积核来降低存储成本。该方法利用线性组合对学习到的字典权重重构出输出特征图，并对训练好的网络进行卷积核分解。Tai 等[49]提出从零开始训练低秩约束卷积神经网络模型的方法，不仅速度得到了提升，一些模型的性能也有所提高；还提出了低秩分解的新方法，通过消除卷积核中的冗余来找到矩阵分解的全局优化器，优于迭代算法。

但在近几年，随着压缩方法的优化，低秩分解的方法已经开始逐渐退出舞台。低秩分解成本高昂且不利于全局参数的压缩，很多时候又需要通过大量的重新训练来达到收敛的效果，增加了时间成本。而且最近几年提出了一种新型的 1 * 1 轻量化卷积核，这种卷积核被大量运用，而低秩分解的方法却不适用于这种卷积核，因此逐渐被淘汰。

3.4　知识蒸馏

与剪枝量化等压缩方法不同，知识蒸馏需要两种不同类型的网络来分别作为学生模型和教师模型来进行压缩，其过程如图 5 所示。教师模型作为训练好的模型，由它来为之后的学生模型提供知识，也就是更丰富的分类信息，学生模型就可以通过蒸馏训练的方式获取这些知识。该方法能以轻微的性能损失为代价将一个复杂的教师模型中的知识迁移到精简的学生模型中，从而完成模型压缩。

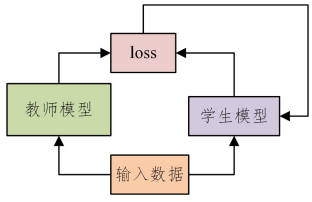

图 5　知识蒸馏示意图

Fig. 5　Schematic diagram of knowledge distillation

Shen 等[50]运用了一种选择性学习方案，对于每个未标记的样本让学生模型从预测模糊度最小的教师模型中自适应地学习知识。该压缩方法效果出色，但在现实生活中训练数据因为隐私或者安全问题经常是不可知的。面对这种情况，Lopes 等[51]提出了一种无数据知识蒸馏方法，用于将大规模数据集上训练的深度神经网络压缩到小部分。该方法只需要一些额外的元数据来提供一个预训练的模型，在降低训练成本的同时解决了隐私问题。除此之外，与多个教师模型进行知识蒸馏也是一种简单方法，主要训练学生的预测与多个教师模型的平均预测之间的交叉熵损失。You 等[52]基于该思想提出了一个更有效的策略，用一个相对差异损失来扩大它

们之间的交叉熵损失。该损失定义在为学生和教师集合之间的一个三重实例生成的中间层输出上。对于学生模型，选择了中间层，对于每个教师模型选择的这一层，使大多数教师模型与投票策略下的顺序关系一致。随着 AI 技术的发展，知识蒸馏也开始和其他技术结合起来进行优化，如知识蒸馏和生成对抗性网络(Generative Adversarial Networks，GAN)。GAN 由生成器网络和判别器网络组成，生成器网络接收一个随机向量作为输入，之后就生成一个假样本作为输出，判别器则作为二分类器去尽可能地分辨输入图像的真假。若输入的图像是真实图片，判别器则需要给出很高的分数；若输入的图像是生成器生成的图片，判别器则要打低分来进行区分。而生成器的目标，就是让自己生成的图像尽可能地被打出一个高分从而混淆判别器。GAN 就是通过这种对抗的形式来实现生成器和判别器彼此性能的提升和优化。在知识蒸馏中，就可以直接把教师网络的数据集丢给一个随机初始化的 GAN，从头开始训练然后生成假图片用来进行蒸馏。GAN 效果优异，但由于其计算成本过高，内存占用过大，在资源受限的设备上难以部署。为解决这个问题，Ren 等[53]提出了一种新颖的多粒度在线知识蒸馏方法，用于得到轻量级的 GAN 模型，放弃了复杂的多级压缩过程，设计了一种面向 GAN 的在线蒸馏策略，一步获得压缩模型，从多个层次和粒度中挖掘潜在的互补信息，以帮助优化压缩模型。

虽然知识蒸馏能够显著压缩模型规模，减小计算成本，但是依然存在两个缺点：1)只适用于具有 Softmax 损失函数分类任务，阻碍了其广泛应用；2)应用条件过于严格，在性能方面比其他压缩方法差。

3.5 混合方式

模型剪枝、模型量化以及知识蒸馏等压缩方法都有着不错的压缩效果，然而在单独运用时也都存在着各自的局限性。因此提出的混合方式压缩，即混合多种压缩方法，可以取得更好的压缩效果并弥补互相的缺陷。如剪枝和量化，剪枝用于减少模型中的冗余连接，而量化用于将模型参数从高精度表示转换为低精度表示。通过先剪枝减小模型规模，然后对剪枝后的模型进行量化，可以同时获得模型尺寸的显著减小和计算量的显著减少。

Ullrich 等[54]基于 Soft weight-sharing 的正则化项进行量化和剪枝，并在过程中揭示了压缩和最小描述长度原则之间的关系。Han 等[55]提出了深度压缩(Deep Compression)方法，不仅结合了非结构化剪枝和参数量化，还通过哈夫曼编码进行存储压缩。该方法在不影响精度的前提下实现了 AlexNet 参数从 240×10^6 减少到 6.9×10^6，VGG-16 参数量从 552×10^6 减少到 11.3×10^6，运行速度提高了 3～4 倍。随后又在此基础上考虑到软件和硬件的协同压缩设计，提出了 Efficient Inference Engine 框架[56]。Dubey 等[57]提出了一种基于过滤器核心集表示的卷积神经网络压缩算法，该算法同样结合了剪枝、量化以及哈夫曼编码，实现了良好的压缩效果，与深度压缩类似。Louizos 等[58]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，使用后验不确定性来确定最优的定点精度以编码权重。Ji 等[59]通过重新排序输入/输出维度进行剪枝，并将具有较小值的不规则分布权重聚类到结构化组中，以实现更好的硬件利用率和更高的稀疏性。

3.6 小结

为了更好地比较复杂模型压缩方法的优劣，本文在表 2 和表 3 中对这些经典压缩方法的压缩效果和性能损失进行了对比和分析。从表 2 可以看出，模型剪枝压缩方法对性能的影响都微乎其微，而且对参数的压缩效果较好。其中，Liu 等[25]提出的 FDNP 方法实现了最高的参数压缩比，压缩效果最好，性能损失很低，说明动态剪枝相比静态剪枝有较好的性能提升。文献[51]提出的方法的压缩效果最差，说明知识蒸馏压缩方法相比剪枝和量化，不仅造成了很高的性能损失，压缩比率也不理想。在混合压缩方法中，深度压缩(Deep Compression)能够有效提升模型性能，但其压缩比率在这几种方法对比中并不出色。与这些方法相比，文献[57]提出的新型卷积神经网络压缩方法能够实现最高的压缩比率，而且其性能损失也在可接受范围内。

表 2 不同模型压缩方法在 LeNet-5 on MNIST 上的压缩效果对比

Table 2 Compression effect comparison of different compression methods on LeNet-5 on MNIST

分类	方法	ΔAccuracy	♯Params
模型剪枝	Ref.[15]	+0.03	12x
	DNS[23]	0	108x
	Ref.[19]	−0.1	63x
	FDNP[25]	0	130x
知识蒸馏	Ref.[51]	-6.18	2x
混合方式	Soft weight-sharing[54]	−0.09	162x
	Deep compression[55]	+0.06	39x
	Ref.[57]	−0.01	193x
	Ref.[58]	−0.1	156x
	Ref.[59]	0	10x

表 3 不同量化方法在 AlexNet 上的压缩效果对比

Table 3 Compression effects comparison of different quantization methods on AlexNet

方法	Top-1 Δ accuracy	Top-5 Δ accuracy	Weight bits	Activation bits
XNOR-Net[31]	−12.4	−11.0	1	1
SYQ[35]	0	−0.8	1	8
	+1.5	+0.6	2	8
Ref.[36]	−3.0	−2.7	1	32
	−1.8	−1.8	2	32
TTQ[33]	+0.3	+0.6	2	32
SLQ[38]	+0.46	+0.3	5	32
INQ[34]	+0.15	+0.23	5	32

在表 3 中，weight bits 代表参数量化的位数，为 1 代表二值化量化，为 2 则代表三值化量化。从表 3 可以看出，XNOR-Net 虽然有较高的压缩比例，但其性能损失较为严重。SYQ 在实现二值化量化时能够较好地压缩模型，实现三值化量化时甚至有不错的性能提升，相比其他压缩方法具有优越性。而 SLQ 和 INQ 方法在将权值量化到 5 位时，能够在有效压缩的同时提升准确率，改善模型的性能。

本节从复杂模型压缩的角度，介绍了模型剪枝、模型量化、低秩分解、知识蒸馏以及混合方式 5 种压缩方法，先通过介绍概念帮助读者理解模型压缩的基本方法，然后通过对经典方法的引用和分析，帮助读者更好地理解这 5 种模型压缩方法之间的优劣。通过对比可以发现，混合压缩方法因为集合了多种压缩方法的优点，在多个方面的表现的确优于其他方法。模型剪枝、模型量化以及知识蒸馏作为热门的压缩方法，相关研究较多且压缩方式丰富，出现了动态剪枝、无数据量化以及知识浓缩蒸馏等新颖的压缩方法，发展前景广阔。

4　轻量化模型设计方法

复杂模型压缩是利用神经网络模型中参数以及结构的冗余来精简神经网络模型，主要特点是拥有一个已经训练出来的复杂神经网络。而轻量化模型设计没有预训练神经网络模型，而是直接构建了一个精简的、轻量化的神经网络模型。轻量化模型设计方法可以分为人工设计方法和自动设计方法，人工设计方法主要通过优化卷积结构、精简卷积操作来完成，主要包含了空间卷积设计和移位卷积设计两种；而自动设计方法则是通过某种策略来自动设计出高性能、精简的轻量化模型，也就是神经架构搜索（Neural Architecture Search，NAS）。

4.1　空间卷积设计

空间卷积设计就是通过运用一种经过优化的、简便的卷积操作来替代卷积神经网络中的普通卷积，以达到压缩模型规模、减少参数运算量的目的。该类方法主要包含空间可分离卷积和深度可分离卷积两种，它们通过拆分常规卷积操作来减少冗余计算，从而实现模型参数和计算量的压缩。

在空间可分离卷积中，将常规卷积操作拆分为逐通道卷积和逐像素卷积两个独立的阶段，以降低计算复杂度。逐通道卷积独立地在每个输入通道上进行操作，而逐像素卷积则对每个像素点进行卷积计算。而深度可分离卷积也就是优化了的空间分离卷积操作，其操作更加复杂，效果也更明显，比如 Howard 等[60]提出的轻量级 MobileNet 新型神经网络便运用了深度可分离卷积，运用 depth-wise 操作和 point-wise 操作也就是深度卷积和逐点卷积来代替普通卷积操作，大大减小了参数规模和降低了运算成本。深度可分离卷积中，一个卷积核负责一个通道并只被一个卷积核卷积。逐点卷积的运算与常规卷积运算非常相似，它的卷积核的尺寸为 $1\times1\times M$，M 为上一层的通道数，因此卷积运算会将上一步的 map 在深度方向上进行加权组合，生成新的 Feature map。MobileNetV2 是 Google 公司为移动设备和嵌入式视觉应用提出的一个尺寸更小、延迟更低的进阶版 MobileNet。整体网络用 MobileNetV1 的深度可分离卷积，其作为基础运算单元，从而完成轻量化模型的构建。MobileNetV2 在结构上则借鉴了 ResNet[61]，并进行了两方面的改进：1）引入了线性瓶颈结构，即删除了在低维度输出层后所连接的非线性激活层来保证信息的完整性；2）引入了反向残差结构，采用先“升维”再“降维”的方式来保证特征信息的有效传递。

ShuffleNet 是一个和 MobileNet 一样采用深度可分离卷积作为基础运算单元并部署在边缘端的轻量化模型，这个效率极高的 CNN 架构由 Zhang 提出[62]，该架构相比 MobileNet 主要有两个创新操作：逐点群卷积和通道混洗。与现有的先进模型相比，该架构在相近的准确率下大大减少了计算量，在 ImageNet 和 MS COCO 数据集上，ShuffleNet 相比其他先进模型表现出了优越性能。ShuffleNetV2[63]是旷视科技提出的一种高效的采用深度可分离卷积作为基础运算单元的 CNN 模型，和 MobileNetV2 一样适用于移动端的轻量级网络模型。文献[63]对目前的一些主流网络进行对比实验，并从中进行一定的理论分析和总结，最后得出帮助 CNN 神经网络更高效训练的 4 条准则：1）输入通道数与输出通道数保持相等可以最大化地减少内存访问成本；2）分组卷积中使用过多的分组数会增加内存访问成本；3）网络结构太复杂，分支和基本单元过多会降低神经网络结构的并行程度；4）Element-wise 的操作成本也不可忽略，包括 ReLU 和 Tensor 的相加以及偏置的相加等操作。最后基于上述准则在 ShuffleNet 的基础上，提出了 ShuffleNet V2，并通过实验验证了该网络结构的优越性。

但这些轻量级模型在实际应用中有一定的局限性，网络尺寸大小应高于一定数值，即网络通道数量不应设置过少，否则会导致信息传递过程中丢失过多信息，进而大幅度降低精确度。

4.2　移位卷积设计

移位卷积设计主要通过将基础神经网络模型中的卷积运算替换为移位卷积运算，来设计出紧凑的神经网络模型。

ShiftNet[64]是由伯利克大学的研究人员提出的一种无参数、无 FLOP 的新型网络结构。文献[64]提出运用无参数、无 FLOP 的移位卷积操作来替代空间卷积实现准确性和成本之间的平衡，融合移位和逐点卷积，以构建端到端可训练的基于移位的框架，使其有超参数特性。为了证明其效果，对 ResNet 的卷积操作进行替换，不仅减少了 60％的参数量，还提高了其在 CIFAR-10 和 CIFAR-100 数据集上的准确性。最后，展示了移位卷积操作在各个领域的适用性，以更少的参数量实现了出色的性能。ShiftNet 采用移位运算和逐点卷积代替空间卷积来降低计算复杂度和减少参数量，但网络中的移位量采用启发式进行分配，网络训练时间较长且优化困难，并不能达到网络压缩最优的目的。为了解决该问题，Jeon 提出了新型轻量级网络 As-ResNet[65]，该神经网络模型参照反向传播算法的特点，结合 ShiftNet 的标准移位，提出了主动移位的思想，与其标准卷积运算相比，其主要对基础运算方式进行改进。主动移位将卷积操作解构为两个操作步骤进行实现，分别是移位运算和逐点卷积运算。As-ResNet 的创新点在于使用主动移位替代空间卷积，并以此设计了主动移位层。主动移位层将移位运算量化为具有参数的函数，通过反向传播来学习移位量，能够模仿出各类型卷积运算。但 As-ResNet 中移位运算依靠于内存操作，其直接影响网络运行效率，选用合适的硬件设备十分重要。

但移位卷积模型在实际应用中往往存在训练时间过长和网络运行效率过低等问题，因此选用合适的硬件帮助轻量级模型训练和运行是十分重要的。

4.3　NAS 架构搜索

NAS 是一种结合了优化和机器学习的交叉研究，也是一门用于自动设计神经网络结构的技术。NAS 在某些任务中甚至可以媲美人类专家的水准，发现一些人类之前未曾提出的网络结构。与空间卷积设计和移位卷积设计相比，NAS 这种自动设计轻量化神经网络模型的方法拥有广泛的应用前景。NAS 主要由搜索空间、搜索策略和性能评估策略 3 部分组成。

搜索空间定义了神经网络架构算法可以搜索到的神经网络结构类型，也定义了如何描述神经网络结构。如今常见的结构便是链式结构、多分枝结构以及基于 Cell 的结构这 3 种。链式结构就是神经网络中的每一层都与其前后两层相连，没有跨层连接的情况，如 LeNet5 网络便是一种典型的链式结构网络。多分支结构则是神经网络模型中的层可以与其前面

的任意层进行连接，如 GoogleNet 中 Inception 模块的多分支以及 ResNet 中的残差结构。而基于 Cell 的结构就是研究者试图将多个操作组合成 Cell，并将 Cell 作为组成神经网络的基本单元，以此设计了基于 Cell 的网络结构。如 Liu 等[66]设计了深度神经网络的层次化表现方法，该方法第一次在初始的基本操作上演化来得到 Cell 结构。

搜索策略是在搜索空间被构建好之后运用的方法，用来确定最优的神经网络结构。而目前流行的搜索策略主要有 6 种，分别是随机搜索、贝叶斯优化、神经进化、强化学习和基于梯度的方法。随机搜索是一种简单直接的方法，通过在搜索空间中随机选择网络结构来探索可能的解。虽然随机搜索能够覆盖广泛的结构，但其效率较低，可能需要大量的实验和时间。贝叶斯优化是一种基于概率推断的优化方法，通过之前的实验结果构建高斯过程模型来估计网络结构的性能，并选择具有高性能概率的结构进行下一步探索。贝叶斯优化能够快速收敛于较好的解，减少实验次数。神经进化是受生物进化理论启发的一种方法，通过模拟进化过程中的选择、交叉和变异来优化网络结构。它通过不断迭代生成和评估一组候选解，来逐渐改进网络结构并找到更好的解。强化学习方法将网络结构搜索视为一个马尔可夫决策过程，通过代理智能体与环境交互，学习选择网络结构的策略。它利用奖励信号来引导搜索过程，逐步调整结构以获得更好的性能。基于梯度的方法是一种使用梯度信息来指导网络结构搜索的方法。它通过计算网络结构对性能的梯度，以及使用梯度下降等优化算法来更新网络结构，逐步改进性能。这些搜索策略在神经网络结构搜索领域发挥着重要作用，它们各自具有不同的特点和适用范围，根据具体的问题和需求选择合适的搜索策略。未来的研究还可以探索结合多种策略的混合方法，以进一步提高搜索效率和性能。

第三部分就是性能评估策略，当搜索策略在搜索空间搜索到一个神经网络结构后就需要性能评估策略来对该网络结构进行评估，之后将结果反馈给搜索算法，来指导搜索算法找到最优的网络结构。但是因为性能评估需要使用到反向传播对网络进行训练，所以需要大量的计算成本和时间成本，搜索的效率很低。因此，加速网络架构评估的方法依旧十分重要。最常见的方法包括 3 种，分别是低保真度、早停以及代理模型。低保真度主要指采用减少样本数量、降低图像分辨率、减少网络层数等方法，通过近似的数据集和近似的网络架构虽然加快了搜索过程，减少了计算成本，但是这不可避免地会引入偏差。早停指在网络未收敛时便停止训练。代理模型则是采用简单的近似任务来替代实际的训练任务，然后便将代理模型得到的结果当作神经网络结构的性能。

虽然神经架构搜索在时间成本上相比以往有很大降低，但是目前其时间成本问题依旧存在，未来仍需要改进搜索策略来大幅降低时间成本。同时，神经架构搜索对实验环境的要求较高，实验机器的采购成本过高，不利于普惠性推广。未来可以通过对策略以及评估等方面进行优化，降低神经架构搜索的成本消耗。

4.4 小结

表 4 列出了一些轻量级模型的参数量、浮点数计算量以及准确度数据对比情况。通过对比可以发现，轻量化模型设计方法在参数量和浮点数计算量方面都有减少，而准确率损失也都在接受的范围内，甚至一部分优秀轻量化模型的性能比原模型能有一定提升。MobileNet V2 模型在参数量、浮点运算量以及性能方面都非常优异。而 ShuffleNet V2 参数量略大，但性能提升明显。ShiftNet 相比其他轻量化模型进行了过多的浮点数运算，导致其计算成本太高。

表 4 经典轻量化模型的对比

Table 4 Comparison of classic lightweighting models

Model	Params	Top-1/%	MFLOPS
MobileNet V1[60]	4.20×10^6	70.6	569
MobileNet V2	3.40×10^6	72.0	300
ShuffleNet V1[62]	3.40×10^6	71.5	530
ShuffleNet V2[63]	5.30×10^6	73.7	300
ShiftNet[64]	4.10×10^6	70.1	1400
As-ResNet[65]	3.42×10^6	72.2	729
GoogleNet	6.80×10^6	69.8	1550
AlexNet	60.90×10^6	57.2	725
VGG-16	138.00×10^6	71.5	15300

本章从模型轻量化设计的角度了解 3 种设计方案，分别是空间卷积设计、移位卷积设计以及 NAS 架构搜索。

MobileNet 和 ShuffleNet 采用深度可分离卷积作为基础卷积运算单元，降低了卷积运算计算复杂度，减少了参数量，可以看出选用适当的基础卷积运算对设计紧凑神经网络模型有着重要意义。与卷积运算相比，采用标准移位运算同样能够实现聚合空间信息的目的，是一种有效的替代方法。标准移位先将输入信息进行移位运算，重新排列空间信息，后进行逐点卷积，聚合空间信息。标准移位运算在速度上优于标准卷积运算，且其计算代价与核大小无关。因此，这样的运算方式有利于紧凑神经网络模型的设计。而 NAS 架构搜索需要在搜索前对卷积层、卷积单位和卷积核的大小等参数进行预先设定，随后使用预设定的参数在一个巨大的网络空间上搜寻。但该方法的成本问题和对资源的消耗量等问题依然存在，有一定的局限性，而对于资源较为充沛的设计人员而言，不失为一种设计轻量化模型的好方法。

5 未来研究方向

随着物联网时代的到来以及 AI 的快速发展，在边缘侧部署神经网络模型的应用也越来越广泛。此外，AI 强大的学习能力也为提高产品的自动化程度提供了切实可行的方法，因此模型轻量化技术的重要性也日益提高。虽然已经存在轻量级模型运用的实例，但是在实际应用中进行部署以及产品化等方面仍旧有着非常大的提升空间。

1)数据安全问题：模型轻量化过程中存在数据安全问题。因为很多模型压缩方法需要通过训练集来先对模型进行训练，这时运用到的训练集就会涉及用户隐私。对此，一部分研究者在 GAN 生成对抗性网络上找到了解决方法。GAN 包含两个模型：生成模型和判别模型。生成模型主要是生成与原始数据相似的样本数据，判别模型就是判断样本数据是真实的还是伪造的。将两个模型进行对抗和博弈，以此来生成数据，实现无数据模型压缩，从而解决模型压缩过程中的数据安全问题。

2)硬件架构支持：将模型轻量化技术与硬件架构设计相结合是未来的研究方向。目前的轻量化方法大多从软件层面

对模型进行优化，并且不同方法的硬件平台不同，很难比较其压缩与加速效果的好坏。未来可以针对主流的模型轻量化方法设计出专门的硬件架构，既能与软件层面的模型轻量化方法结合，又能够比较不同方法。

3）更大范围的推广：如今的模型轻量化技术大多面对的是进行图片分类任务的卷积神经网络，而在实际应用中也有其他模型应用于人工智能领域。如用于语音识别的循环卷积神经网络（RNN）和用于知识图谱领域的图神经网络（GNN）。而用于卷积神经网络模型的压缩方法一般不能直接用于这两类神经网络模型。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神经网络模型的直接部署。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。

结束语　本文首先介绍了模型轻量化技术的研究背景以及本文的研究贡献；其次，从复杂模型压缩与轻量化模型设计两个层面对模型轻量化技术进行介绍，复杂模型压缩方法包含模型剪枝、参数量化、低秩分解、知识蒸馏以及混合方式5种方法，对这些方法分别引用了一些经典的文献来进行分类介绍，而模型设计则从空间卷积设计、移位卷积设计以及NAS架构搜索3个方面来进行分析；最后，展望了未来模型压缩与加速技术的研究方向。

参 考 文 献

[1] GAO H，TIAN Y L，XU Y，et al. A review of deep learning model compression and acceleration[J]. Journal of Software，2021，32(1)：68-92.

[2] TANG W H，DONG B，CHEN H，et al. A review of deep neural network model compression methods[J]. Intelligent IOT Technology，2021，4(6)：1-15.

[3] GENG L L，NIU B N. A review of deep neural network model compression[J]. Computer Science and Exploration，2020，14(9)：1441-1455.

[4] LANG L，XIA Y Q. A review of research on compact neural network model design[J]. Computer Science and Exploration，2020，14(9)：1456-1470.

[5] 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.

[6] YANG A，PAN J，LIN J，et al. Chinese CLIP：Contrastive Vision-Language Pretraining in Chinese[J]. arXiv：2211. 01335，2022.

[7] GULATI A，QIN J，CHIU C C，et al. Conformer：Convolution-augmented transformer for speech recognition[J]. arXiv：2005. 08100，2020.

[8] XU J，TAN X，LUO R，et al. NAS-BERT：task-agnostic and adaptive-size BERT compression with neural architecture search[C]// Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021：1933-1943.

[9] LIU Y，ZHANG W，WANG J. Zero-shot adversarial quantization[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021：1512-1521.

[10] ZHU J，TANG S，CHEN D，et al. Complementary relation contrastive distillation[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021：9260-9269.

[11] WIMMER P，MEHNERT J，CONDURACHE A. Interspace pruning：Using adaptive filter representations to improve training of sparse cnns[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022：12527-12537.

[12] HANSON S，PRATT L. Comparing biases for minimal network construction with back-propagation[J]. Advances in Neural Information Processing Systems，1988，1：177-185.

[13] LECUN Y，DENKER J，SOLLA S. Optimal brain damage[J]. Advances in Neural Information Processing Systems，1990，2：598-605.

[14] HASSIBI B，STORK D G，WOLFF G J. Optimal brain surgeon and general network pruning[C]// IEEE International Conference on Neural Networks. IEEE，1993：293-299.

[15] HAN S，POOL J，TRAN J，et al. Learning both weights and connections for efficient neural networks[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1. 2015：1135-1143.

[16] DONG X，CHEN S，PAN S. Learning to prune deep neural networks via layer-wise optimal brain surgeon[C]// Advances in Neural Information Processing Systems. 2017：4857-4867.

[17] MARIET Z，SRA S. Diversity networks：Neural network compression using determinantal point processes[J]. arXiv：1511. 05077，2015.

[18] KINGMA D P，SALIMANS T，WELLING M. Variational dropout and the local reparameterization trick[C]// Advances in Neural Information Processing Systems. 2015：2575-2583.

[19] MOLCHANOV D，ASHUKHA A，VETROV D. Variational dropout sparsifies deep neural networks[C]// International Conference on Machine Learning. PMLR，2017：2498-2507.

[20] SRINIVAS S，BABUA R V. Data-free parameter pruningfor deep neural networks[J]. arXiv：1507. 06149，2015.

[21] WIMMER P，MEHNERT J，CONDURACHE A. Interspace pruning：Using adaptive filter representations to improve training of sparse cnns[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022：12527-12537.

[22] ZHUANG L，LI J，SHEN Z，et al. Learning Efficient Convolutional Networks through Network Slimming[C]// 2017 IEEE International Conference on Computer Vision（ICCV）. IEEE，2017.

[23] GUO Y，YAO A，CHEN Y. Dynamic network surgery for efficient DNNs[C]// Advances in Neural Information Processing Systems. 2016：1379-1387.

[24] JI L，RAO Y，LU J，et al. Runtime neural pruning[C]// Neural Information Processing Systems. 2017.

[25] LIU Z，XU J，PENG X，et al. Frequency-domain dynamic pruning for convolutional neural networks[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018：1051-1061.

[26] GAO X，ZHAO Y，DUDZIAK L，et al. Dynamic Channel Pruning：Feature Boosting and Suppression：，10. 48550/arXiv. 1810. 05331[P]. 2018.

[27] CHEN J, CHEN S, PAN S J. Storage Efficient and Dynamic Flexible Runtime Channel Pruning via Deep Reinforcement Learning[C]// Neural Information Processing Systems. 2020.

[28] LI C, WANG G, WANG B, et al. Dynamic slimmable network [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8607-8617.

[29] 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.

[30] 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]. arXiv: 1602: 02830, 2016.

[31] RASTEGARI M, ORDONEZ V, REDMON J, et al. Xnor-net: Imagenet classification using binary convolutional neural networks[C]// Proc. of the European Conf. on Computer Vision. Cham: Springer-Verlag, 2016: 525-542.

[32] LI F, ZHANG B, LIU B. Ternary weight networks[J]. arXiv: 1605. 04711, 2016.

[33] ZHU C, HAN S, MAO H, et al. Trained ternary quantization [J]. arXiv: 1612. 01064, 2016.

[34] ZHOU A, YAO A, GUO Y, et al. Incremental network quantization: Towards lossless cnns with low-precision weights[J]. arXiv: 1702. 03044, 2017.

[35] FARAONE J, FRASER N, BLOTT M, et al. SYQ: Learning symmetric quantization for efficient deep neural networks [C]// Proc. of the IEEE Conf. on Computer Vision and Pattern Lopes R G, Fenu S, Starner T. Data-free Knowledge Distillation for Deep Neural Networks[J]. arXiv: 1710. 07535, 2017.

[36] LENG C, DOU Z, LI H, et al. Extremely low bit neural network: Squeeze the last bit out with ADMM[C]// Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[37] GONG Y, LIU L, YANG M, et al. Compressing deep convolutional networks using vector quantization[J]. arXiv: 1412. 6115, 2014.

[38] XU Y, WANG Y, ZHOU A, et al. Deep neural network compression with single and multiple level quantization[C]// Proc. of the 32nd AAAI Conf. on Artificial Intelligence. 2018.

[39] DETTMERS T. 8-bit approximations for parallelism in deep learning[J]. arxiv: 1511. 04561, 2015.

[40] WANG K, LIU Z, LIN Y, et al. HAQ: Hardware-Aware Automated Quantization With Mixed Precision[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.

[41] FANG J, SHAFIEE A, ABDEL-AZIZ H, et al. Post-training Piecewise Linear Quantization for Deep Neural Networks[C]// Computer Vision – ECCV 2020. Lecture Notes in Computer Science Vol. 12347. Cham: Springer, 2020.

[42] LI Y, DONG X, WANG W. Additive Powers-of-Two Quantization: An Efficient Non-uniform Discretization for Neural Networks[C]// International Conference on Learning Representations. 2020.

[43] YAMAMOTO K. Learnable Companding Quantization for Accurate Low-bit Neural Networks: , 10. 48550/arXiv. 2103. 07156 [P]. 2021.

[44] WANG Z, XIAO H, LU J, et al. Generalizable mixed-precision quantization via attribution rank preservation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 5291-5300.

[45] CAI Y, YAO Z, DONG Z, et al. Zeroq: A novel zero shot quantization framework[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 13169-13178.

[46] ZHONG Y, LIN M, NAN G, et al. Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12339-12348.

[47] GAO H, TIAN Y L, XU F Y, et al. A review of deep learning model compression and acceleration[J]. Journal of Software, 2021, 32(1): 68-92.

[48] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions[J]. arXiv: 1405. 3866, 2014.

[49] TAI C, XIAO T, ZHANG Y, et al. Convolutional neural networks with low-rank regularization[J]. arXiv: 1511. 06067, 2015.

[50] SHEN C, XUE M, WANG X, et al. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3504-3513.

[51] LOPES R G, FENU S, STARNER T. Data-free knowledge distillation for deep neural networks[J]. arXiv: 1710. 07535, 2017.

[52] YOU S, XU C, XU C, et al. Learning from multiple teacher networks[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 1285-1294.

[53] REN Y, WU J, XIAO X, et al. Online multi-granularity distillation for gan compression[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6793-6803.

[54] ULLRICH K, MEEDS E, WELLING M. Soft weight-sharing for neural network compression[J]. arXiv: 1702. 04008, 2017.

[55] SONG H, MAO H, DALLY W J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding[C]// ICLR. 2016.

[56] HAN S, LIU X, MAO H, et al. EIE: Efficient inference engine on compressed deep neural network[J]. ACM SIGARCH Computer Architecture News, 2016, 44(3): 243-254.

[57] DUBEY A, CHATTERJEE M, AHUJA N. Coreset-based neural network compression[C]// Proc. of the European Conf. on Computer Vision(ECCV). 2018: 454-470.

[58] LOUIZOS C, ULLRICH K, WELLING M. Bayesian compression for deep learning[C]// Advances in Neural Information Processing Systems. 2017: 3288-3298.

[59] JI Y, LIANG L, DENG L, et al. TETRIS: Tile-matching the tremendous irregular sparsity[C]// Advances in Neural Information Processing Systems. 2018: 4115-4125.

[60] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J].

arXiv:1704.04861,2017.

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

[62] 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.

[63] MA N,ZHANG X,ZHENGH 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.

[64] WU B,WAN A,YUE X,et al. Shift:A zero flop,zero parameter alternative to spatial convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:9127-9135.

[65] JEON Y,KIM J. Constructing fast network through deconstruction of convolution[C]// 32nd Conference on Neural Information Processing Systems(NeurIPS 2018). Neural Information Processing Systems Foundation,2018:5951-5961.

[66] LIU H,SIMONYAN K,VINYALS O,et al. Hierarchical representations for efficient architecture search[J]. arXiv: 1711.00436,2017.

GAO Yang, born in 2000, master candidate. His main research interests include deep learning and model compression.

DUAN Pengsong, born in 1983, Ph. D. His main research interests include edge computing and intelligent perception.