文章编号:1673-0291(2017)06-0034-08 DOI:10.11860/j.issn.1673－0291.2017.06.006

面向嵌入式应用的深度神经网络模型压缩技术综述

王 磊,赵英海,杨国顺,王若琪
(中国航天科工集团三十五研究所,北京 100013)

摘 要:结合大数据的获取,深度神经网络关键技术广泛应用于图像分类、物体检测、语音识别和自然语言处理等领域.随着深度神经网络模型性能不断提升,模型体积和计算需求提高,以致其依赖高功耗的计算平台.为解决在实时嵌入式系统中的存储资源和内存访问带宽的限制,以及计算资源相对不足的问题,开展嵌入式应用的深度神经网络模型压缩技术研究,以便缩减模型体积和对存储空间的需求,优化模型计算过程.对模型压缩技术进行分类概述,包括模型裁剪、精细化模型设计、模型张量分解和近似计算和模型量化等,并对发展状况进行总结.为深度神经网络模型压缩技术的研究提供参考.

关键词:深度神经网络;模型压缩;模型裁剪;张量分解;嵌入式系统
中图分类号:TP391.4 **文献标志码**:A

A survey on model compression of deep neural network for embedded system

WANG Lei, ZHAO Yinghai, YANG Guoshun, WANG Ruoqi
(The 35th Research Institute of China Aerospace Science and Industry Corp., Beijing 100013, China)

Abstract: Combined the big data acquisition, the key technologies of deep neural network have widely applied in the field of image classification, object detection, speech recognition, natural language processing, et al. With the developing of the deep neural network model performance, the model size and the required calculation need to be improved, so that it is reliance on high power computing platform. This paper is focus on the deep neural network model compression technology for embedded applications in order to solve the problems of storage resource, memory access speed constraints and computing resources limit in embedded system. It aims to reduce the model size and the complex computation. Meanwhile, it could optimize the process of calculation. This paper has summarized the state-of-the-art model compression technologies including model pruning, fine model designing, tensor decomposition, model quantization, etc. Through the summary on the model development, it could provide the references for the studies of the deep neural network model compression technologies.

Keywords: deep neural network; model compression; model pruning; tensor decomposition; embedded system

收稿日期:2017-11-10
基金项目:国家自然科学基金(61572065)
Foundation item: National Natural Science Foundation of China (61572065)
第一作者:王磊(1986—),男,山东聊城人,高级工程师,博士.研究方向为人工智能与模式识别和智能导引头. **email**:wltongxing@163.com.
引用格式:王磊,赵英海,杨国顺,等.面向嵌入式应用的深度神经网络模型压缩技术综述[J]. 北京交通大学学报,2017,41(6):34－41.
WANG Lei, ZHAO Yinghai, YANG Guoshun, et al.A survey on model compression of deep neural network for embedded system [J]. Journal of Beijing Jiaotong University, 2017, 41(6): 34－41. (in Chinese)

深度学习(Deep Learning)技术[1]从人工神经网络技术发展而来,该技术获得快速发展的一个重要原因是其解决了网络层数较多时的训练问题,浅层网络通常用来提取底层特征,而深度神经网络擅长学习大量复杂数据的结构信息,可对数据提取底层特征,然后利用深层网络转化为高层、抽象的表达形式.目前,深度学习技术在图像和语音等领域取得突破性进展.例如图像识别中,卷积神经网络(Convolutional Neural Network,CNN)[2]可从数据中自动学习图像的特征表示,在图像分类中其性能超越了传统人工设计的特征.随着研究的深入,新型网络结构的提出(如 AlexNet,VGG[3]、GoogLeNet[4]和 ResNet[5]等)使识别性能不断上升,如表 1 所示,但随着网络层数不断增加,对存储资源和计算资源的需求越来越高.表 1 中,Top1 指标是判断预测结果中得分最高的目标类别与真实类别标签是否相同,而 Top5 指标则判断目标真实类别标签是否出现在预测结果得分最高的前五类预测类别中.

现阶段深度学习领域的研究方向可分为:1)研究复杂的网络结构[6-9],以实现性能突破;2) 研究深度神经网络压缩(Deep Neural Network Compression)与模型计算效率,将深度学习技术与具体硬件结合,以实现工程化应用.

表 1　新型网络结构模型对比

Tab.1　Comparison of the models in novel network structure

模型名称	提出时间	层数	Top5 错误率/%
AlexNet	2012	8	16.40
VGG	2014	19	7.30
GoogLeNet	2014	22	6.70
ResNet	2015	152	3.57

文献[10]研究了深度神经网络模型压缩技术,主要工作包括对模型网络连接的裁剪(Pruning)处理,仅保留重要性较高的连接,将模型参数进行量化(Quantization),减少模型体积,提高运行效率,并利用霍夫曼编码(Huffman Coding)进一步压缩模型.本文作者对深度学习的模型压缩技术进行概述,研究内容包括模型裁剪和稀疏化(Sparsity)、精细模型设计、模型参数量化和张量分解(Tensor Decomposition)等.

深度神经网络的模型压缩技术有利于减少模型内存占用、计算量和降低功耗等,一方面可以提高模型的运行效率,有利于云计算等对速度要求较高的平台,另一方面有利于将模型部署到嵌入式系统.研究发现深度神经网络的结构具有冗余性,基于该性质可进行模型压缩.模型压缩常采用的方法是模型裁剪[11],通过有效的评估方法确定模型参数的重要程度,基于参数重要性对影响性能较小的网络连接进行裁剪.模型稀疏化[12]利用正则项对模型参数更新进行限制,有利于获取稀疏度较高的模型,即较多的权重等于或者接近 0.模型优化设计[13]也是模型压缩的一种有效方法,通过技巧设计精细和高效的小模型,以较小尺寸的模型实现较高的性能.模型参数量化(例如聚类算法实现权值共享[10],浮点转定点[14]等)有利于提高计算效率和进行硬件优化,还可减小模型尺寸,一种极限的量化方式是进行二值化[15].张量(矩阵)分解法[16]是对深度神经网络的参数矩阵进行张量分解,利用低秩特性实现近似计算,达到减少计算量的目的.

1　深度神经网络模型裁剪

模型裁剪通常对训练完毕的深度神经网络模型进行处理,其核心是寻找确定模型参数重要性的判别依据,将不重要的网络连接关系删除.该方法在卷积层和全连接层中应用较多.

一种直接的方法是根据模型参数值的大小评判其重要性,在卷积神经网络中,对卷积核全部元素的绝对值求和,如果值较低则认为该卷积核对最终结果的影响较小,可删除该卷积核和其所对应的激活值.在裁剪过程中,通过裁剪前和裁剪后的性能分析每一层的裁剪敏感程度,并减少裁剪敏感层的裁剪力度.裁剪后通过微调的方式训练现有模型参数,最终取得与模型裁剪前相近的性能.该方法较易实现,具有较强的实用价值.文献[17]根据神经元连接权值的大小确定其重要性,将低于设定阈值的连接裁剪后,稠密网络连接模型改变为稀疏模型.

文献[18]研究发现较多的神经元的激活值趋近于 0,将该神经元剪除可降低模型大小和减少运算量,对模型性能的影响较小.基于验证样本集,文中定义神经元激活为 0 的平均比例(Average Percentage of Zeros,APoZ),该指标作为评价神经元重要性的标准.但该标准保留了本该剪除的激活值接近 0(例如 0.001)的神经元,同样如果神经元总是输出相同数值,该神经元包含较少信息量,应该被剪除.文献[19]提出基于激活响应熵值的裁剪指标(Entropy),基于验证数据集统计每个通道的平均激活值分布,计算分布的熵,剪除熵值较小的通道所对应的滤波器,为了恢复裁剪前的模型性能,采用两步微调的方式,在裁剪一层模型后进行少量的迭代优化,当全部层完成裁剪后再进行较多次的微调训练,该训练策略可减少所需微调训练次数以及有效避免

模型参数的局部最优解.文献[20]提出一种减少模型计算能量消耗的裁剪方法,作者通过可计算硬件能耗的工具将深度神经网络模型每层的能耗进行排序,优先裁剪能耗较高的层,并使用局部与全局相结合的微调方式进行性能恢复.

文献[21]通过求解网络权重参数的组合优化问题实现网络裁剪,通过分析将每个权重剔除后对模型损失的改变程度,认为损失改变较小权重的重要性较低,但由于权重众多,逐一进行评估的资源和时间成本较高,提出将模型损失依据模型参数进行泰勒展开(Taylor)表示,根据模型损失的变化表示为参数的函数,从而实现对参数重要性的判断,该方法可在训练中实现裁剪.

在模型裁剪过程中,可能将重要连接删除,并且由于网络之间的互相连接关系,将权重裁剪后可能影响其他权重的重要性,文献[22]提出一种动态网络裁剪的方法,根据定义的判别函数,在训练过程中判断权重的重要性,可重新恢复被裁剪的网络连接,取得较好的裁剪效果.文献[23]提出一种基于向量形式的乘法,实现密集矩阵与稀疏矩阵之间的高效乘法计算,提高了稀疏卷积神经网络的运算速度.文献[24]针对卷积神经网络提出通道裁剪的方法(Channel Pruning),首先通过 Lasso 回归的方法对卷积通道进行选择性删除,然后对权重进行学习调整,利用最小二乘法重构通道删除之前的网络响应,在保证模型性能的条件下减少模型参数和计算量.

一些研究工作针对具体问题进行网络裁剪,文献[25]在性别分类问题研究中,利用线性判别分析方法(LDA)分析深度网络模型的激活响应与类别的相关性,利用类间相关性确定模型连接的重要程度,在裁剪过程中仅保留对分类任务具有较强判别作用的网络参数.

模型稀疏化学习是在模型训练过程中对参数的优化过程增加限制条件,使模型的参数稀疏化,有利于提高模型裁剪程度和提升模型运算效率.文献[12]提出一种结构化的稀疏学习方式,在模型优化目标函数增加 Group Lasso 损失函数项[26],实现卷积通道、卷积核和卷积核内部的参数稀疏化.文献[27]通过在目标函数中增加参数的 L_0 范数约束,实现模型的稀疏化,但 L_0 范数求解较困难,因此提出一种阶段迭代算法,首先仅更新权值较大的参数,然后恢复所有网络连接,迭代更新所有参数,在训练中可实现模型裁剪.几种模型裁剪方法及性能对比如表 2 所示.

表 2 典型模型裁剪方法性能对比

Tab.2 Comparison of several pruning methods and their performances

代表方法	网络类型	优化网络层	Top5 精度损失/%	压缩倍数	速度提升倍数
Entropy[19]	VGG-16	卷积层、全连接层	1.0	16.64	3.30
	ResNet-50		1.0	1.47	1.54
Taylor[21]	AlexNet	卷积层	0.3	—	2.00
	VGG-16		2.3		2.20
Channel[24]	VGG-16	卷积层	0.3	—	5.00
	ResNet		1.4		2.00

2 精细深度网络模型设计

精细化模型设计是基于技巧和经验设计精细、高效和体积小的深度网络模型,该方法主要设计新型的网络结构,例如网络层的组合和网络连接关系的改变等.

MobileNets 模型[28]是谷歌公司针对嵌入式设备设计的小型卷积神经网络模型,主要设计思路是将卷积核进行分解,将卷积核为 $s\times s\times c$ 的卷积过程分解为 $s\times s\times 1$ 和 $1\times 1\times c$ 的两次卷积,其中:s 是卷积核尺寸,c 为通道数,核心思想是将每个通道单独计算后再进行线性组合,可较大程度地减少计算量.相比 VGG-16 模型,MobileNets 模型在 ImageNet 的识别精度损失约为 1%,但模型预测过程中的乘法和加法计算量及模型参数量均减少数十倍.

ResNeXt[29]设计可平行放置的具有相同拓扑结构的网络块,替换 ResNet 的基本网络块,并增加网络通道数目,通过采用组卷积(Group Convolution)的方式减少计算量和参数.ShuffleNet 模型[30]中将通道信息进行线性组合的 1×1 卷积过程的运算量较大,拟通过组卷积运算的方式减少计算量,但组卷积的每一组仅可接受前一层的部分信息,无法整合前一网络层的所有通道,因此提出通道重新排列(Channel Shuffle)的操作方法,每个卷积组可接受前一网络层不同卷积组的数据.SqueezeNet[31]采用模块化的设计思路,基本单元称为 Fire 模块,该模块包含压缩部分和扩展部分,压缩部分由 1×1 的卷积核组成,扩展部分由 1×1 和 3×3 的卷积核组成,以较少的模型参数实现较高的性能,同时由于模型参数的减少,对内存访问带宽的

要求降低,有利于嵌入式应用.DeepRebirth 模型[32]将网络层分为权重层(如卷积层和全连接层)和非权重层(如 Pooling 层、ReLU 层等),非权重层的理论计算量较小,但由于内存数据访问速度等原因,其计算耗时较多,提出将非权重层与权重层进行合并的方法,去除独立的非权重层后,运行时间显著减少.精细模型设计方法对比分析如表 3 所示.

表 3　精细模型设计方法性能对比

Tab.3 Analysis comparison of several fine model designing methods

代表方法	网络类型	优化网络层	Top5 精度损失/%	压缩倍数	速度提升倍数
SqueezeNet[31]	AlexNet	卷积层	0	50	—
DeepRebirth[32]	GoogLeNet	卷积层与池化层	0.4	—	3～5
ShuffleNet[30]	AlexNet	卷积层	0	—	13

3　教师-学生网络方法

设计精细化模型对技巧和经验要求比较高,文献[13]提出一种迁移学习(Transfer Learning)的方法:教师-学生网络(Teacher-Student),也称知识提取方法(Knowledge Distillation),利用训练完毕的复杂模型指导简单模型的训练过程,实验结果显示该方法优于简单模型单独进行训练的性能.复杂网络具有较好的性能,训练过程的输出信息表示各类别的分布概率,相比数据标签其所包含的信息更丰富,因此学生模型在训练中的优化目标包含两部分,一部分是学生模型输出的类别概率与真值的交叉熵;另一部分是学生模型与教师模型的类别概率输出的交叉熵,学生模型在训练中获取的信息量更丰富,有利于提高模型训练速度和模型性能.

文献[33]指出在模型的网络层较深时,让学生网络直接模拟教师网络的输出比较困难,文中提出 FitNets 模型,在深度神经网络模型的中间添加监督学习的信号,要求学生模型和教师模型的中间层激活响应尽可能一致.训练过程分为两阶段,首先利用教师网络中间层信息训练学生模型前部参数,再利用教师模型的最终输出信息训练学生模型的全部参数.文献[34]认为 FitNets 方法要求学生模型模仿教师模型的全部特征图,该设定过于严格,由于学生模型与教师模型的能力存在较大差异,FitNets 的限制条件可能不利于参数学习过程的收敛和模型性能,因此利用学生-教师模型网络激活分布的一致性指导学生模型的训练过程,使用最大平均差异(Maximum Mean Discrepancy, MMD)作为损失函数,优化该损失函数促使学生模型和教师模型的网络响应分布尽可能相似.文献[35]引导学生模型学习教师模型网络层之间的数据流信息,该信息定义为层与层之间的内积,两个多通道网络层的数据流关系可用 FSP(Flow of Solution Procedure)矩阵表示,在教师-学生框架中,优化目标函数为教师-学生模型对应层之间 FSP 差异的 L_2 范数.该方法促使学生模型学习教师模型层与层之间的抽象关系,取得较好效果,并可用于其他迁移学习任务.典型的教师-学生机制性能分析如表 4 所示.

表 4　教师-学生机制性能分析

Tab.4　Performance analysis of teacher-student mechanism

代表方法	网络类型	优化网络层	Top5 精度损失/%	压缩倍数	速度提升倍数
FitNets[33]	Maxout	全网络	1.2	36	13.36

4　权重张量分解

张量分解的目的是降低模型的时间复杂度,通常基于张量的低秩近似理论和方法,将原始的权重张量分解为两个或者多个张量,并对分解张量进行优化调整.

文献[36]对深度神经网络模型的权重低秩分解问题进行研究,探索多种张量分解方法,例如二维张量分解可采用奇异值分解法,三维张量可转化为二维张量进行分解,以及单色卷积分解和聚类法低秩分解等.作者利用卷积参数的冗余性获得近似计算过程,较大的减少所需的计算量,在保持原始模型浮动 1%精度的条件下,基于 CPU 和 GPU 的计算过程均取得近 2 倍的加速.文献[37]提出将卷积神经网络大小为 $k\times k$ 的卷积核分解为 $1\times k$ 和 $k\times 1$ 的卷积核,通过基于共轭梯度下降法的权重重构实现权重误差最小化,在模型反向传播过程基于随机梯度下降法的数据重构实现输出响应误差的最小化.在字符识别任务中,在无精度损失的情况下实现 2.5 倍加速,在 1%精度损失的情况下可加速 4.5 倍.

CP 分解法(CP-decomposition)可将张量表示为有限个秩-张量之和,文献[38]采用 CP 分解法将一层网络分解为五层低复杂度的网络层,但在基于随机梯度下降法的模型权重微调过程中难以获取张量分解的优化解.作者利用两个卷积神经网络模型对该方法进行评估,结果表明该方法以较低的性能损失实现更高的计算速度.在 36 类字符分类实验中,该方法获得 8.5 倍的加速,在数据集 ImageNet 的实验结果表明,该方法以增加 1%分类误差代价使第 2 个卷积层的速度提升 4 倍.

文献[16]针对深度卷积神经网络的非线性神经元设计张量分解算法，基于广义奇异值分解(Generalized Singular Value Decomposition，GSVD)进行非线性问题的求解，同时不需要通过随机梯度下降过程进行优化，并在非对称重构中考虑前一网络层的累计重构误差，在不需要随机梯度下降(SGD)的情况下，开发了一种有效的非线性优化问题求解方法.现有研究方法主要侧重优化一两个层，而提出的非线性方法可减少多层累积误差.针对 VGG 16 模型进行优化，在 ImageNet 数据集的实验结果表明，该方法在只增加 0.3%的 Top 5 分类误差的情况下实现 4 倍的全模型加速.

文献[39]采用类似的方法，并提出从零开始训练低秩约束卷积神经网络模型的方法，不仅速度得到提升，而且在一些情况下模型性能也有所提高.作者提出一种低阶张量分解的新算法，用于消除卷积核中的冗余.该算法找到矩阵分解的精确的全局优化器，比迭代方法更有效.研究发现，该方法利用低阶约束在实现显著加速的同时，取得性能的提升.在 CIFAT-10 数据集上，提出的低阶 NIN 模型可以达到 91.31%的精度.在 CIFAR-10 和 ILSVRC12 的基础上，对 AlexNet、NIN、VGG 和 GoogleNet 等各种模型进行评估，并取得运算效率的提升，表明低阶张量分解是一种有效的 CNN 模型加速的工具.

其他张量分解方法有 Tucker 分解法[40]和张量列分解法(Tensor Train，TT)[41]等，在深度神经网络模型压缩中也表现出较好的效果.几种张量分解方法性能分析如表 5 所示.

表 5 张量分解方法性能对比

Tab.5 Performance analysis of tensor decomposition methods

代表方法	网络类型	优化网络层	Top5 精度损失/%	压缩倍数	速度提升倍数
低秩近似[36]	BaselineCNN	卷积层	1.0	—	4.50
CP 分解[38]	AlexNet	卷积层	1.0	—	4.00
GSVD[16]	AlexNet	卷积层	0.4	5.00	1.82
	VGG-16		0.3	2.75	2.05
	GoogLeNet		0.4	2.84	1.20

5 深度神经网络模型量化

对模型进行参数量化的主要目的为减小模型存储体积，解决模型在移动设备的存储问题与计算过程中的数据传输瓶颈问题.参数量化还用来改变模型参数的数据表示方法，通常模型参数类型为浮点型，而有些硬件平台仅支持定点数据类型，需对浮点数进行定点化处理.

参数量化主要分为两个主要研究方向，一个研究方向是权值共享，即多个网络连接的权重共用一个权值，主要代表性研究工作如下.文献[10]利用 k 均值聚类算法计算权重的多个聚类中心，将权重量化为距离最近的聚类中心，通过训练微调的方式对权重进行补偿.文献[42]设计了一种新型的网络架构 HashedNets，利用哈希函数随机将网络连接权重分组到哈希桶(Hash Bucket)，每个哈希桶内的网络连接共享相同的权重参数，该方法与特征哈希类似(Feature Hashing)，将高维数据降到低维空间，该方法可显著减小模型体积，并对输出精度影响较小.文献[43]提出一个量化卷积神经网络框架(Q-CNN)，基于 k 均值聚类算法加速和压缩模型的卷积层和全连接层，通过减小每层输出响应的估计误差可实现更好的量化结果，并提出一种有效的训练方案抑制量化后的多层累积误差.

参数量化的第 2 个研究方向是模型权重的低比特表示方法.文献[44]对深度卷积神经网络模型参数的数据表示方法进行研究，针对浮点数和定点数建立分析模型，探索两种数据表示形式对模型性能的影响以及相应硬件的实现代价.文献[45]开发并测试 8-Bit 近似算法，将 32-Bit 的梯度和激活值压缩到 8-Bit，通过 GPU 集群测试模型和数据的并行化性能，在保证模型预测精度的条件下，提出的方法取得两倍的数据传输加速.文献[46]基于 Maxout 模型在 3 个数据集上对浮点型、定点型和动态定点型 3 种运算方式进行评估，分析计算精度对最终训练误差的影响，实验显示低精度的运算不仅可应用于模型预测，还可用于模型训练过程，但模型参数、激活值和梯度值需根据实际应用条件而采用不同的精度.文献[47]提出一种模型近似框架 Ristretto，用于分析模型卷积层和全连接层的权重和输出的数值分辨率，进而将浮点型参数转化为定点型数值，并通过训练过程对定点型模型进行微调，在容许误差为 1%的条件下，将 CaffeNet 和 SqueezeNet 压缩到 8-Bit 参数形式.

文献[48]提出渐进式网络量化(Incremental Network Quantization，INQ)无损低比特权重量化技术(5-Bit)，可将浮点型深度神经网络模型转化为无损的低比特二进制化数据模型，在硬件通过移位计算实现乘法过程，有利于在移动平台的部署和加速.该方法首先将模型的每一层参数按照其绝对值分为两组，将权重值较大一组的参数量化后保持固定，另外一组参数通过再训练过程进行调整，以补偿参数量化所造成的精度损失，重复该过程直到权重

全部量化完毕.

DoReFa-Net[49]利用低比特的梯度参数训练低比特的模型权重,且激活值也为低比特数据,该技术可对训练和预测过程进行加速,并可有效地应用到 CPU,FPGA,ASIC 和 GPU 等硬件.二值神经网络模型[15,50]可视为一种极限化的量化模型,将权值和激活值二值化为 1 或者 −1,模型压缩比例较高,同时可利用位操作实现乘法运算,有效减少模型前向传播过程的运算时间.几种参数量化方法对比见表 6.

表 6　参数量化方法性能对比

Tab.6　Performance comparison of parameter quantization methods

代表方法	网络类型	优化网络层	Top5 精度损失/%	压缩倍数	速度提升倍数
Q-CNN[43]	AlexNet	卷积层、全连接层	0.8	15.4	4.1
	CNN-S		0.8	16.3	5.7
	VGG-16		0.5	16.6	4.1
8-Bit[45]	32Bit-CNN	卷积层、全连接层	0	4.0	2.0
5-Bit[48]	AlexNet	卷积层、全连接层	0	53.0	—

6　总结与展望

本文作者对深度神经网络模型压缩的主要技术方法进行分类概括和总结分析.模型压缩的主要目标是在满足精度损失要求的情况下,提高模型压缩比例和提升模型预测速度.模型压缩是嵌入式深度学习工程化实现的基础,其可在算法层面对模型体积和计算量进行优化,在此基础上可开展嵌入式架构设计、模型与数据的调度优化等工作.

模型压缩的每种技术手段均存在一定的局限性,例如模型裁剪技术是对现有模型进行剪枝操作,并根据需要进行再训练,在满足性能要求的条件下,其对模型尺寸的压缩力度较小,并且在计算量方面的优化能力较弱;精细模型设计可实现较高的压缩比和模型运算速度的提升,但需要较高的设计技巧;教师-学生机制较容易推广使用,具有较大的压缩比,但需要具备性能良好的教师模型,且学生模型的设计需经验指导;张量分解法具有良好的理论基础,其缺点是代码实现过程较复杂,在满足较高性能的条件下其压缩比例难以较大幅度的提升;参数量化技术的优点是具有较高的模型压缩比,但需要编译器和硬件的支持.

未来的研究工作主要集中在探索新型模型压缩技术,以及将多种技术进行结合,同时需了解嵌入式系统的工作机制,通过软硬件协同的方式进行模型压缩,在减小模型体积的同时提升模型的运行速度.

参考文献(References):

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. Nature, 2015, 521: 436−444.

[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems, 2012:1097−1105.

[3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]// ICLR, 2015:1−14.

[4] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1−9.

[5] 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.

[6] HUANG G,LIU Z,WEINBERGER K Q,et al.Densely connected convolutional networks[C]//CVPR,2016.

[7] CHEN Y, LI J, XIAO H, et al. Dual path networks [EB/OL]. (2017-08-01)[2017-10-01]. https://arxiv.org/abs/1707.01629.

[8] ZHANG X, LI Z, LOY C C, et al. Polynet: a pursuit of structural diversity in very deep networks [EB/OL]. (2017-07-17) [2017-10-01]. https://arxiv. org/abs/1611.05725.

[9] HU J, SHEN L, SUN G. Squeeze and excitation networks[EB/OL]. (2017-09-05)[2017-10-01]. https://arxiv.org/abs/1709.01507.

[10] HAN S, MAO H, DALLY W J. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding [EB/OL]. (2016-02-15) [2017-10-01]. https://arxiv. org/abs/1510.00149.

[11] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient convnets[EB/OL]. (2017-05-10) [2017-10-01]. https://arxiv.org/abs/1608.08710.

[12] WEN W, WU C, WANG Y, et al. Learning structured sparsity in deep neural networks [C]//Advances in Neural Information Processing Systems, 2016: 2074−2082.

[13] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[EB/OL]. (2015-05-09) [2017-11-01]. https://arxiv.org/abs/1503.02531.

[14] SHIN S, HWANG K, SUNG W. Fixed-point performance analysis of recurrent neural networks[C]// IEEE

International Conference on Acoustics, Speech and Signal Processing, 2016: 976－980.

[15] COURBARIAUX M, HUBARA I, SOUDRY D, et al. Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or －1[EB/OL]. (2016-05-17)[2017-11-01]. https://arxiv.org/abs/1602.02830.

[16] ZHANG X, ZOU J, HE K, et al. Accelerating very deep convolutional networks for classification and detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(10): 1943－1955.

[17] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network [C]//Advances in Neural Information Processing Systems, 2015: 1135－1143.

[18] HU H, PENG R, TAI Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures [EB/OL]. (2016-07-12) [2017-11-01]. https://arxiv.org/abs/1607.03250.

[19] LUO J H, WU J. An entropy-based pruning method for CNN compression [EB/OL]. (2017-06-19) [2017-11-01]. https://arxiv.org/abs/1706.05791.

[20] YANG T J, CHEN Y H, SZE V. Designing energy-efficient convolutional neural networks using energy-aware pruning[EB/OL]. (2017-04-18)[2017-11-01]. https://arxiv.org/abs/1611.05128.

[21] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient transfer learning[EB/OL]. (2017-06-08)[2017-11-01]. https://arxiv.org/abs/1611.06440.

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

[23] PARK J, LI S, WEN W, et al. Faster CNNs with direct sparse convolutions and guided pruning[EB/OL]. (2017-07-28) [2017-11-01]. https://arxiv. org/abs/1608.01409.

[24] HE Y, ZHANG X, SUN J. Channel pruning for accelerating very deep neural networks[EB/OL]. (2017-08-21)[2017-11-01]. https://arxiv.org/abs/1707.06168.

[25] TIAN Q, ARBEL T, CLARK J J. Efficient gender classification using a deep LDA-pruned net[EB/OL]. (2017-10-31) [2017-11-01]. https://arxiv. org/abs/1704.06305.

[26] YUAN M, LIN Y. Model selection and estimation in regression with grouped variables[J]. Journal of the Royal Statistical Society, 2006, 68(1): 49－67.

[27] JIN X, YUAN X, FENG J, et al. Training skinny deep neural networks with iterative hard thresholding methods [EB/OL]. (2016-07-19) [2017-11-01]. https://arxiv.org/abs/1607.05423.

[28] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications [EB/OL]. (2017-04-17) [2017-11-01]. https://arxiv.org/abs/1704.04861.

[29] XIE S, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks[EB/OL]. (2017-04-11) [2017-11-01]. https://arxiv. org/abs/1611.05431.

[30] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: an extremely efficient convolutional neural network for mobile devices[EB/OL]. (2017-07-04) [2017-11-01]. https://arxiv.org/abs/1707.01083.

[31] 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]. (2017-11-04)[2017-11-10]. https://arxiv.org/abs/1602.07360.

[32] LI D, WANG X, KONG D. DeepRebirth: accelerating deep neural network execution on mobile devices[EB/OL]. (2017-08-16) [2017-11-01]. https://arxiv. org/abs/1708.04728.

[33] ROMERO A, BALLAS N, KAHOU S E, et al. Fitnets: hints for thin deep nets[EB/OL]. (2015-05-27) [2017-11-01]. https://arxiv.org/abs/1412.6550.

[34] HUANG Z, WANG N. Like what you like: knowledge distill via neuron selectivity transfer[EB/OL]. (2017-07-05)[2017-11-01]. https://arxiv.org/abs/1707.01219.

[35] YIM J, JOO D, BAE J, et al. A gift from knowledge distillation: fast optimization, network minimization and transfer learning [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4133－4141.

[36] DENTON E L, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Advances in Neural Information Processing Systems, 2014: 1269－1277.

[37] JADERBERG M, VEDALDI A, ZISSERMAN A. Speeding up convolutional neural networks with low rank expansions [EB/OL]. (2014-05-15)[2017-11-01]. https://arxiv. org/abs/1405.3866.

[38] LEBEDEV V, GANIN Y, RAKHUBA M, et al. Speeding-up convolutional neural networks using fine-tuned CP-decomposition [EB/OL]. (2015-04-24) [2017-11-01]. https://arxiv.org/abs/1412.6553.

[39] TAI C, XIAO T, ZHANG Y, et al. Convolutional neural networks with low-rank regularization[EB/OL]. (2016-02-14)[2017-11-01]. https://arxiv.org/abs/1511.06067.

[40] KIM Y D, PARK E, YOO S, et al. Compression of deep

convolutional neural networks for fast and low power mobile applications[EB/OL]. (2016-02-24)[2017-11-01].https://arxiv.org/abs/1511.06530.

[41] NOVIKOV A, PODOPRIKHIN D, OSOKIN A, et al. Tensorizing neural networks[C]//Advances in Neural Information Processing Systems, 2015: 442－450.

[42] CHEN W, WILSON J, TYREE S, et al. Compressing neural networks with the hashing trick[C]//International Conference on Machine Learning, 2015: 2285－2294.

[43] WU J, LENG C, WANG Y, et al. Quantized convolutional neural networks for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 4820－4828.

[44] **王佩琪, 高原, 刘振宇, 等. 深度卷积神经网络的数据表示方法分析与实践[J]. 计算机研究与发展**, 2017, 54(6): 1348－1356.
WANG Peiqi, GAO Yuan, LIU Zhenyu, et al. A comparison among different numeric representations in deep convolution neural networks[J]. Journal of Computer Research and Development, 2017, 54(6): 1348－1356.(in Chinese)

[45] DETTMERS T. 8-bit approximations for parallelism in deep learning[EB/OL]. (2016-02-19)[2017-11-01]. https://arxiv.org/abs/1511.04561.

[46] COURBARIAUX M, BENGIO Y, DAVID J P. Training deep neural networks with low precision multiplications[EB/OL]. (2015-09-23)[2017-11-01]. https://arxiv.org/abs/1412.7024.

[47] GYSEL P, MOTAMEDI M, GHIASI S. Hardware-oriented approximation of convolutional neural networks[EB/OL]. (2016-10-20)[2017-11-01]. https://arxiv.org/abs/1604.03168.

[48] ZHOU A, YAO A, GUO Y, et al. Incremental network quantization: towards lossless cnns with low-precision weights[EB/OL]. (2017-08-25)[2017-11-01]. https://arxiv.org/abs/1702.03044.

[49] ZHOU S, WU Y, NI Z, et al. DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients[EB/OL]. (2016-07-17)[2017-11-01]. https://arxiv.org/abs/1606.06160.

[50] LIN Z, COURBARIAUX M, MEMISEVIC R, et al. Neural networks with few multiplications[EB/OL]. (2016-02-26)[2017-11-01]. https://arxiv. org/abs/1510.03009.

(上接第 33 页)

[9] TAIGMAN Y, YANG M, RANZATO M, et al. DeepFace: closing the gap to human-level performance in face verification[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1701－1708.

[10] LIU Z, LUO P, WANG X, et al. Deep learning face attributes in the wild[C]// IEEE International Conference on Computer Vision, 2015: 3730－3738.

[11] MASI I, TRAN A T, HASSNER T, et al. Do we really need to collect millions of faces for effective face recognition? [C]//European Conference on Computer Vision, 2016: 579－596.

[12] TRAN A T, HASSNER T, MASI I, et al. Regressing robust and discriminative 3D morphable models with a very deep neural network[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1493－1502.

[13] HU G, YANG H, YUAN Y, et al. Attribute-enhanced face recognition with neural tensor fusion networks[C]//IEEE International Conference on Computer Vision, 2017: 3744－3753.

[14] REDMO J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2016: 779－788.

[15] BANSAL A, NANDURI A, CASTILLO C, et al. UMDFaces: an annotated face dataset for training deep networks[EB/OL]. (2017-05-21)[2017-09-01]. https://arxiv.org/abs/1611.01484.