软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(2):251−266 [doi: 10.13328/j.cnki.jos.005428]

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

深度网络模型压缩综述*

雷 杰, 高 鑫, 宋 杰, 王兴路, 宋明黎

(浙江大学 计算机科学与技术学院,浙江 杭州 310027)

通讯作者: 宋明黎, E-mail: brooksong@zju.edu.cn

摘 要: 深度网络近年来在计算机视觉任务上不断刷新传统模型的性能,已逐渐成为研究热点.深度模型尽管性能强大,然而由于参数数量庞大、存储和计算代价高,依然难以部署在受限的硬件平台上(如移动设备).模型的参数在一定程度上能够表达其复杂性,相关研究表明,并不是所有的参数都在模型中发挥作用,部分参数作用有限、表达冗余,甚至会降低模型的性能.首先,对国内外学者在深度模型压缩上取得的成果进行了分类整理,依此归纳了基于网络剪枝、网络精馏和网络分解的方法;随后,总结了相关方法在多种公开深度模型上的压缩效果;最后,对未来的研究可能的方向和挑战进行了展望.

关键词: 深度神经网络;网络压缩;网络剪枝;网络精馏;网络分解

中图法分类号: TP301

中文引用格式: 雷杰,高鑫,宋杰,王兴路,宋明黎.深度网络模型压缩综述.软件学报,2018,29(2):251−266. http://www.jos.org.cn/1000-9825/5428.htm

英文引用格式: Lei J, Gao X, Song J, Wang XL, Song ML. Survey of deep neural network model compression. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):251−266 (in Chinese). http://www.jos.org.cn/1000-9825/5428.htm

Survey of Deep Neural Network Model Compression

LEI Jie, GAO Xin, SONG Jie, WANG Xing-Lu, SONG Ming-Li

(School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

Abstract: Deep neural networks have continually surpassed traditional methods on a variety of computer vision tasks. Though deep neural networks are very powerful, the large number of weights consumes considerable storage and calculation time, making it hard to deploy on resource-constrained hardware platforms such as mobile system. The number of weights in deep neural networks represents the complexity to an extent, but not all the weights contribute to the performance according to recent researches. Specifically, some weights are redundant and even decrease the performance. This survey offers a systematic summarization of existing research achievements of the domestic and foreign researchers in recent years in the aspects of network pruning, network distillation, and network decomposition. Furthermore, comparisons of compression performance are provided on several public deep neural networks. Finally, a perspective of future work and challenges in this research area are discussed.

Key words: deep neural network; network compression; network pruning; network distillation; network decomposition

深度网络压缩是指利用数据集对已经训练好的深度模型进行精简操作,进而得到一个轻量且准确率相当的网络.压缩后的网络具有更小的结构和更少的参数,可以降低计算和存储开销,因而可以被部署在受限的硬件环境中(如移动设备等).

本文尽可能全面地整理了近年在深度网络压缩方面的研究工作,系统地评价和比较了它们在公开模型上

* 基金项目: 国家自然科学基金(61572428, U1509206)
Foundation item: National Natural Science Foundation of China (61572428, U1509206)
收稿时间: 2017-05-02; 修改时间: 2017-07-24; 采用时间: 2017-10-16; jos 在线出版时间: 2017-12-01
CNKI 网络优先出版: 2017-12-04 08:57:35, http://kns.cnki.net/kcms/detail/11.2560.TP.20171204.0857.018.html

的压缩效用.第 1 节介绍深度网络压缩提出的背景和合理性.第 2 节~第 4 节分别从网络剪枝、网络精馏和网络分解这 3 个角度归纳相关研究方法的演变和主要思想,并讨论各方法中的核心步骤.第 5 节从多种压缩指标上比较各类压缩方式在公开深度模型上的效用.第 6 节探讨深度网络压缩的进一步发展方向.第 7 节对全文进行总结.

1 研究背景

深度学习[1]作为近年来机器学习领域最炙手可热的子领域,在许多领域取得了非常成功的应用.从最初的手写数字识别[2]到近年来图像识别[3]、检测[4]、追踪[5]以及智能问答系统[6],深度网络模型都取得了传统方法无法企及的成就.深度学习的成功一方面得益于其更多的参数和更大、更深的模型,另一方面得益于学术界、工业界贡献的大规模标注或者未标注数据.具体来说,庞大的模型结构增强了其非线性拟合的能力,而大规模数据增强了模型的泛化能力.

尽管深度网络模型在许多问题的实验中表现优越,但在实际应用中依然受到时间和空间上的制约.大而深的深度网络模型运算量大,即使借助图形处理器(graphics processing unit,简称 GPU)加速[7],时间上也依然不能满足许多应用场景的需求.此外,大规模模型参数也要占用大量的内存空间,这对于手机等移动设备来说是无法适用的.因此,在不影响深度网络模型效果的前提下,压缩网络模型是一个重要的研究问题.

传统的深度网络模型主要由卷积层、非线性激活层、下采样层以及全连接层等模块堆叠起来组成.卷积层具有局部连接、权重共享的特点,虽然需要训练的参数不多,但一次前向的耗时较大;相比之下,全连接层虽然参数可达到网络全部参数的 80%以上,但占用前向推断的时间不多.经典的深度网络模型可以参考最初用于图像识别的 AlexNet[8]或者 VGG[9]等网络模型.这些模块大致可分为两类:一类是包括卷积层、全连接层等在内的含有训练参数的模块,其中,参数的数量往往是人为设定的;另一类是包括非线性激活层以及下采样层等在内的不含有任何训练参数的层.模型参数在一定程度上代表了模型的复杂度,也在一定程度上决定着模型所占据的空间大小.人为设定的参数数量往往是在实验室经过重复实验调出来的,这种局部最优的超参数并不代表网络的“真正需求”:它们既存在一定程度上的冗余,也没有权衡成本和效果之间的关系.因此,网络压缩的一个方向是通过压缩模型的参数数量来降低模型的复杂度,比如后面将要介绍的小模型拟合大模型方法等.

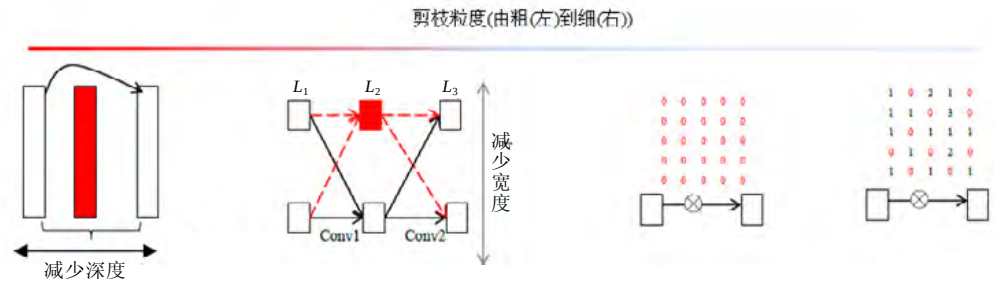
模型的运算时间成本并不仅仅依赖模型的参数数量,也依赖于模型的深度.以残差网络[10]为例,尽管在何凯明经典论文中,1 000 多层的残差网络参数数量不到 AlexNet 的 1/10,但其训练以及测试耗时都明显大于 AlexNet.更深的网络还会在训练阶段产生更多的中间变量,而这些中间变量是反向传播算法必不可少的.换句话说,更深的网络模型也有着更大的内存空间的需求.从这个角度看,深度网络模型的压缩不仅仅是减少模型的参数,更重要的是能够降低模型运算时间,将模型的深度控制在合理的范围之内,从而满足实际应用的需要.

2 网络剪枝

网络剪枝(network pruning)[11]早期指删除网络中冗余参数,提高网络泛化能力.文献[12]通过考量权重衰减作为权重参数的泛化成本,提出了一种在反向传播网络中动态地选择隐含节点的个数的方式.文献[13]则将遗传算法引入进来,用于优化前向神经网络中的权重链接以及发掘新的连接模式和结构.在 CNN 出现之后,网络剪枝主要指减少冗余的浮点计算(floating point operations,简称 FLOP)[14],从而提高网络运行效率.

网络剪枝按剪枝粒度(pruning granularities)可分为 4 类,如图 1 所示,从粗到细为:中间隐层(layer)剪枝、通道(feature map/channel/filter)剪枝、卷积核(kernel)剪枝、核内权重(intra kernel weight)剪枝、单个权重(weight)剪枝.其中,Feature Map/Channel 都指网络中一层产生的特征图张量的一个通道.Filter 是网络中的权重参数,Feature Map 是网络输出,在网络剪枝中两者等价,因为减去一个 Filter 会导致少产生一个 Feature Map.从剪枝目标上分类,可分为减少参数/网络复杂度、减小过拟合/增加泛化能力/提高准确率、减小部署运行时间(test run-time)/提高网络效率以及减小训练时间等.不同的剪枝方法侧重也会有所不同,有的剪枝方法完全依赖网络参数,剪枝后不需要调优恢复准确率;有的剪枝方法则只适用于全连接层剪枝.下面按照剪枝粒度的分类从细到

粗加以叙述.



(a) 层间剪枝 (b) 特征图剪枝 (c) $k\times k$ 核剪枝 (d) 核内剪枝

Fig.1 Four possible pruning granularities [15]

图 1 4 种剪枝粒度[15]

2.1 单个权重粒度

早期 Le Cun[16]提出的 OBD(optimal brain damage)将网络中的任意权重参数都看作单个参数,能够有效地提高预测准确率,却不能减小运行时间;同时,剪枝代价过高,只适用于小网络.设目标函数为 $E=f_{NN}(X,U)$,其中,X 为输入数据,U 为神经网络参数,通过矢量函数的泰勒展开[17]并基于对角假设、极值假设和二次假设,可以近似得到:

$$\delta E=\frac{1}{2}\sum_i h_{ii}\delta u_i^2 \tag{1}$$

由此,Le Cun 提出利用二阶导来近似参数的显著性.

Hassibi 等人[18,19]对 Le Cun 使用的对角假设提出了质疑,提出了 OBS(optimal brain surgeon)的方法,增加了基于手术恢复(surgery)权重更新的步骤,在准确率、泛化能力上获得了更大的提升.但是两者都需要在一轮迭代中计算更新全部参数的显著度量值,文献[14]更是需要计算 Hessian 矩阵及其逆矩阵.因此,该剪枝方法无法在大型网络上使用.

Srinivas 等人[20]对稠密的全连接层进行的参数剪枝与 OBS 极其相似.该方法通过大量推导近似,极大地减小了计算复杂度.通过探索神经元之间的相似性和冗余性,该方法在剪枝时完全不依赖任何训练数据,只依赖网络中的权重,因而通过手术恢复准确率时不需要任何训练数据调优.

这种方法的主要思路是:将剪枝看作是将小权重置零的操作,手术恢复则相当于找到相似权重补偿被置零的权重造成的激活值损失.两个权重的相似程度定义如下:

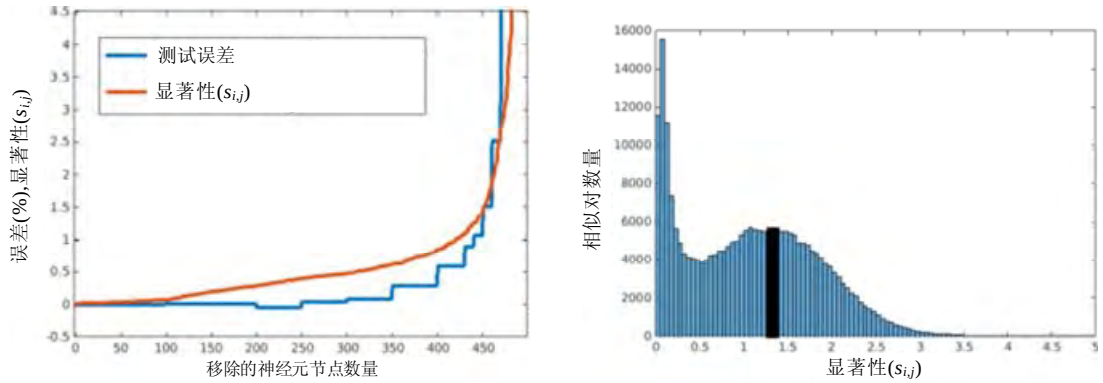
$$s_{i,j}=\langle a_j^2\rangle \| \varepsilon_{i,j} \|^2 \tag{2}$$

其中,

- $\varepsilon_{i,j}=W_i-W_j$,用来度量输入节点 i 和节点 j 之间权重矢量的相似程度;
- $\langle a_j^2\rangle$ 为输出节点 j 权重的均值,表示节点 j 与 0 的接近程度.

整体剪枝步骤为:对所有可能权重矢量的组合,初始化时计算 $s_{i,j}$ 构成的矩阵,找到矩阵中最小的一项(i',j'),删去第 j' 个神经元,并更新权重 $a_{j'}\leftarrow a_{i'}+a_{j'}$;然后,再通过简单的删除与叠加操作更新 S 矩阵,就完成了一次剪枝与手术恢复.

最后,通过实验观察,Srinivas 提出了完全只依赖于权重的参数的自动化剪枝方法[20].如图 2(b)所示,统计直方图的第 2 个峰值对应的 $s_{i,j}$ 即为图 2(a)错误率即将大幅上升的临界点.因此,剪枝时只需将 $s_{i,j}$ 小于该临界值的节点全部删除即可.



(a) 网络错误率、当前被删神经元显著性与神经元删除数目的关系(*Y* 轴经过一定的缩放)

(b) 一层全连接层中权重统计直方图[20]

Fig.2 Automatic pruning method based on weights

图 2 依赖于权重的参数的自动化剪枝方法

2.2 核内权重粒度

在第 2.1 节中,网络中的任意权重被看作是单个参数并进行随机非结构化剪枝,该粒度的剪枝导致网络连接不规整,需要通过稀疏表达来减少内存占用,进而导致在前向传播预测时,需要大量的条件判断和额外空间来标明零或非零参数的位置,因此不适用于并行计算.Han 虽然在文献[21]中将 VGG-16 压缩了 49 倍,获得了很好的效果.但文献[22]中也指出,如果在前向传播预测时利用这种非结构化的稀疏性,则需要使用专门的软件计算库或者寄希望于未来的硬件.

Anwar 等人[23]提出了结构化剪枝的概念,可以很方便地使用现有的硬件和 BLAS 等软件库进行矩阵相乘,利用剪枝后网络的稀疏性来加速网络效率.粗粒度剪枝,如通道粒度和卷积核粒度本身就是结构化的,Anwar 的创新之处在于提出了核内定步长粒度(intra kernel strided sparsity),将细粒度剪枝转化为结构化剪枝.

该方法首先随机初始化步长 m 和偏置 n.考虑到卷积核一般选取 $k\times k$ 的方阵,起始项的下标(i,j)选为 $i=j=n$,则遍历的位置如(n,n),$(n+m,n)$,$(n,n+m)$等.核内定步长粒度剪枝的关键思想在于:作用到同一输入特征图上的 Kernel 必须采用相同的步长和偏置.当卷积层不是稠密连接时,作用在不同特征图上的 Kernel 步长与偏置可以不同,但是,如果卷积层的连接为一般的全连接(即一个特征图需要被所有 Kernel 作用一遍再加和生成新的特征图),那么所有 Kernel 必须采用相同的步长和偏置.这是由于只有相同的步长与偏置,才能在 Lowering(cuDNN 中的 im2col)操作时形成大小匹配的 Lowering Kenrel Matrix,从而减小核矩阵和特征图矩阵的大小,极大地节约计算资源.

除了使用之前提到的定义显著性度量,并进行贪婪剪枝的方法以外,Anwar 还提出了一种使用进化粒子滤波器决定网络连接重要性的方法[23].设 x_k 是状态向量,用来决定是否剪去某一连接权(由 3 位张量降维得到);Z_k 观测值决定该粒子的权重;选用训练好的网络作为观察函数 $h(\cdot)$.粒子滤波由下列方程描述:

$$\left.\begin{aligned} x_k &= f(x_{k-1})+\mu_k \\ Z_k &= h(x_k)+V_k \end{aligned}\right\} \tag{3}$$

其中,观测过程具体为:通过一次前向测试误分类率(misclassification rate,简称 MCR),$h(x_k)=1-MCR$,在噪声 V_k 的干扰下得到观测值 Z_k.

剩下的步骤,如权重重采样(sequential importance resampling,简称 SIR)仍采用传统方法.文献[15]通过实验证明:蒙特卡洛方法比人为定义显著性度量结合贪婪剪枝的方法要好,在同样的剪枝程度,使用粒子滤波可以保证准确率降低得更少.

2.3 卷积核粒度与通道粒度

卷积核粒度与通道粒度属于粗粒度剪枝,不依赖任何稀疏卷积计算库及专用硬件;同时,能够在获得高压缩率的同时大量减小测试阶段的计算时间.由于减去一个特征图意味着相连的卷积核将被一同减去,因此,本节将卷积核粒度与通道粒度放在一起叙述.

设第 i 层卷积层的卷积核矩阵 $\mathcal{F}_i \in \mathcal{R}^{n_i \times n_{i+1} \times k \times k}$, 则 Kernel 为 $\mathcal{F}_{i,\alpha,\beta} \in \mathcal{R}^{k \times k}$,Filter 为 $\mathcal{F}_{i,\beta} \in \mathcal{R}^{n_i \times k \times k}$. 设第 i 层输入 Feature Map 矩阵为 $x_i \in \mathcal{R}^{n_i \times h_i \times w_i}$, 则 Feature Map(即 channel)为 $x_{i,\alpha} \in \mathcal{R}^{h_i \times w_i}$. 用上述记号表示降维可视化,如图 3 所示.图中第 i 层卷积层的卷积核矩阵可以看作由 $n_i \times n_{i+1}$ 个 $k \times k$ 的卷积核组成(即图中每一个网格).一个 Filter 作用在全部 n_i 个输入 Feature Map(即为 Channel)上,产生 1 个新的 Feature Map,因而 n_{i+1} 个 Filter 产生 n_{i+1} 个 Feature Map.从而 Filter 将三维输入 x_i 转换为三维输出 x_{i+1}.

按照如图 3 所示的降维可视化方式,可以进一步理解 Feature Map/Filter 粒度剪枝的具体步骤是减去第 i 层的 filter,进而减去第 i 层产生的部分 Feature Map 和第 i+1 层的部分 Kernel.

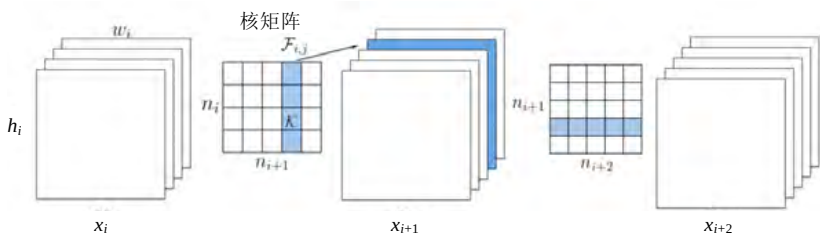


Fig.3 Effect on the i-th and (i+1)-th layers of reducing some filters of layer i [24]

图 3 减去第 i 层部分 Filter 对第 i 层和 i+1 层的影响[24]

Kernel 粒度的显著性度量可以简单地采用 Kernel 的权重和来判断,也可以采用 Polyak[25]提出的 InboundPruning 方法.输出的一个特征图为 n_i 个卷积核作用在对应的输入特征图上的加和:

$$x_{i+1,\beta} = \sum_{\alpha=1}^{n_i} x_{i,\alpha} \times \mathcal{F}_{i,\alpha,\beta} \tag{4}$$

使用通道贡献方差 $\sigma_{i,\alpha}$ 度量第 α 个卷积核的显著程度,定义如下:

$$\sigma_{i,\alpha} = var(\|x_{i,\alpha} \times \mathcal{F}_{i,\alpha,\beta}\|_F) \tag{5}$$

其中,$\|\cdot\|_F$ 表示矩阵的欧式范式,最终的方差是随机输入数据产生的特征图范式的方差.

FeatureMap 粒度的显著性度量也可以简单地选取 Filter 权重和作为显著性度量,关键在于如何确定剪枝数量以及如何对网络整体剪枝[26].Li 等人提出了全局贪婪剪枝(holistic Global pruning)[24],选取 Filter 权重和作为显著性度量.对每一层中的 Filter 按照显著性从大到小排序,进而画出权重和关于排序后下标的曲线.若曲线陡峭,则在这一层减去更多的 Filter;若曲线平缓,则减去较少的 Filter,为剪枝数量提供了经验性的指导.具体剪枝数量则作为超参数优化,在每一层剪枝数量确定之后,开始对整个网络进行全局贪婪剪枝.全局是指在全部剪枝完成后,再通过一次训练恢复准确率;贪婪是指减去上一层 Filter 后,更新下一层部分 Kernel 内的权重,从而在下一层剪枝时,已经减掉的 Kernel 不再对 Filter 贡献任何显著性.

文献[27]通过实验发现,大部分神经元经过激活函数后的输出基本为 0.即这部分网络结构基本不受输入的影响,可以看作是冗余的.进而,文中提出基于统计的方法删除那些对大部分不同输入都输出零值的单元,并进行交替的重新训练.文献[28]则结合贪婪剪枝和基于反向传播的微调来确保剪枝后的网络的泛化性.具体地,文中提出了一种基于泰勒展开来近似计算去除部分参数后网络的损失函数的变化.文献[29]也提出一种对整个神经元进行剪枝的策略,文中提出一种最大化输出(maxout)单元,将多个神经元合并为更加复杂的凸函数表达,并根据各个神经元在训练集上的响应的局部相关性进行选择.

3 网络精馏

早期的相关研究集中于在神经网络进行二次学习.Zhou 在文献[30]中指出了神经网络规则中保真度和准确性两难的问题,其中,保真度度量规则模拟神经网络行为的能力,准确性描述规则的泛化能力.针对这种两难的现象,该文献指出,应该分别针对两种目标分别进行规则提取.进一步地,Zhou 等人在文献[31]提出一种 NeC4.5 策略,利用决策树的可解释性,将多个网络中的抽象规则进行集成表达.

二次学习的思想也逐渐扩展到深度网络上,其中,Bucila 等人[32]在 2006 年形象化地使用了网络精馏(network distillation)的概念,这一概念在后续 Hinton 等人[33]的研究中得到了延用.网络精馏是指利用大量未标记的迁移数据(transfer data),让小模型去拟合大模型,从而让小模型学到与大模型相似的函数映射.网络精馏可以看成在同一个域上迁移学习[34]的一种特例,目的是获得一个比原模型更为精简的网络,整体的框架图如图 4 所示.

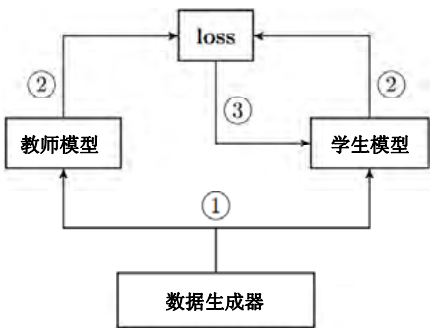


Fig.4 Framework of network distillation

图 4 网络精馏框架

大模型作为教师模型(teacher model)是预先训练好的,小模型作为学生模型(student model),由教师模型指导,步骤①首先由数据生成器生成大量的迁移数据(transfer data),分别送入教师模型和学生模型中.步骤②将教师模型的输出作为真实值,衡量学生模型的输出与它之间的损失.步骤③通过梯度下降等方法更新学生模型的权重,使得学生模型的输出和教师模型的输出更加接近,从而达到利用小模型拟合大模型的效果.

3.1 利用迁移数据训练学生模型

Bucila 等人的工作中[32]提出的模型压缩的方法涉及到了网络精馏的思想,通过让学生模型的输出 label 和教师模型的输出 label 尽量接近来拟合教师模型.这样的方式通常比直接用训练集训练一个小网络的效果要好很多,但是需要大量的迁移数据.迁移数据通常比训练数据还要大几十倍,因此,文献[35]采用人工合成数据的方式,提出了 RANDOM、NBE[33]、MUNGE 这 3 种方案来获得大量未标记的迁移数据.

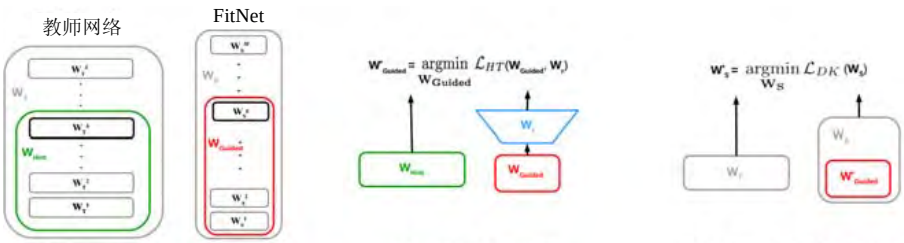
当训练数据是海量的时候,考虑到教师模型不会过拟合,可以使用训练数据集作为迁移数据集.例如,Google 的 JFT 数据集包含了 1 亿张图片和 15 000 个分类.利用网络精馏的方法,Hinton 等人在 JFT 数据集上进行实验[33],首先采用网络精馏的方法训练了 61 个专家模型(每一个专家模型就是一个学生模型),每一个专家模型的迁移数据是包含某些容易混淆的类的不同的数据子集,从而使得每一个专家模型都擅长区分特定的容易混淆的类.然后联合教师模型一起进行预测,将测试集准确率提高了 1.1%.更为关键的是,使用传统的模型融合方法,训练一个模型需要 6 个月的时间,而采用网络精馏的方法可以同时训练若干个专家模型,只需几天就可以完成,极大地缩短了训练时间.

3.2 学生模型的结构

相对于教师模型,学生模型的网络参数更少,运行时间更快.但是学生模型的网络结构没有一个固定的选择方案,通常都是人为经验设定.

在 Ba 等人[36]的工作中,他们认为网络不需要特别深.因此,他们设计了一个相对于教师模型更浅的学生模型,同时保证两者的网络参数相同,这也意味着学生模型的每一层更宽.经过实验发现,浅的网络也可以表现出与深的网络相类似的性能.

与上述工作提出的结论相反,Romero 等人则认为,一个越深的网络可以带来更高的分类准确率.他们提出并设计了 FitNets[37],相对于教师模型,可以将学生模型的网络结构设计得更窄、更深,但是总的参数还是少于教师模型.但是这样做同时也带来了一个问题:越深的网络越难训练.当网络层数增加时,使用 Hinton 等人提出的网络精馏的方法会造成无法很好地训练优化的困境.因此,Romero 等人提出,引入教师模型中间层的输出 Hints 作为监督信息,先训练学生模型的前半部分,让学生模型中间层的输出和教师模型中间层的输出尽量接近,如图 5 所示.



(a) 教师和学生网络 (b) Hints方式训练 (c) 知识蒸馏

Fig.5 Training the student model with hints [37]

图 5 使用 hints 来训练学生模型[37]

由于学生模型中间层的输出和教师模型中间层的输出维度可能不一致,所以需要增加一个回归器来让两者的维度保持一致.训练过程中,首先采用 Hints 方式(如图 5(b)所示)训练 FitNet 的前半部分,然后采用传统的网络精馏来训练整个 FitNet.Romero 等人通过实验发现:采用 FitNet,在增加网络深度、减少网络参数的同时,准确率还比教师模型更好,因而证明了深度是提高网络性能的一个关键因素[38].

Chen 等人在文献[39]中提出了一种网络生长的方法来获得学生模型的网络结构,类似的工作还有文献[40],都是希望利用子网络已训练好的权重中的隐含知识.具体包括 Net2WiderNet 和 Net2DeeperNet 这两种策略,分别从宽度和深度上进行网络生长,然后利用网络蒸馏的方法训练学生模型.图 6 是 Net2Net 方法与传统方法的对比,Net2Net 的核心思想是以复制的方式重用先前训练好的网络参数,并在此基础上进行结构扩展.

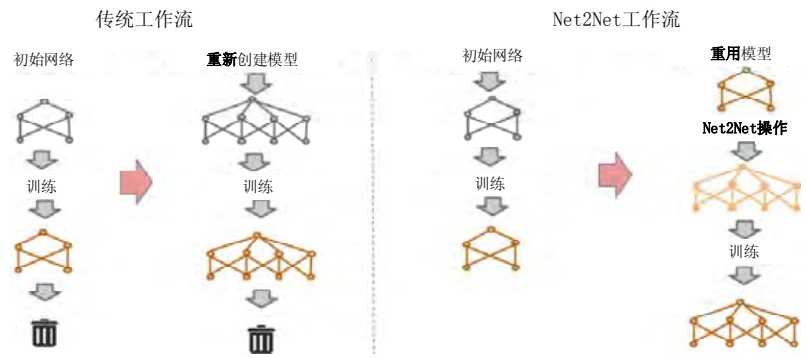


Fig.6 Comparing Net2Net workflow with traditional method [39]

图 6 Net2Net 工作流与传统方法的对比[39]

3.3 损失函数的选取

在 Bucila 等人提出的模型压缩方法中,采用学生模型的标签与教师模型的标签尽量接近的方法来拟合教

师模型的函数映射.Ba 等人提出:与其将标签作为输出的监督信息,不如将 Softmax 层的输入 Logits 值作为监督信息,让学生模型输出的 Logits 去拟合教师模型的 Logits[36].例如,教师模型预测出 3 个目标类的预测概率为[2e−9,4e−5,0.9999],如果我们直接使用这些概率作为预测目标来最小化交叉熵[41]损失函数,那么学生模型将会更侧重于第 3 个类而忽略前两个类;但如果使用 Logits 作为预测目标时,得到的新目标是[10,20,30],学生模型在学习到第 3 类的同时也会学到前两类.这个例子说明:相对于学习标签,Logits 包含了更多的知识信息,从而能够让教师模型更好地指导学生模型.

Hinton 等人在上述工作的基础上做了进一步的改善.首先,Hinton 等人指出:通过加入一个温度变量 T,对所有的 logits 都除以 T,再作为输入送入 Softmax 层,将其输出的软标签(soft target)作为监督信息.整个过程可以用公式(6)表示.

$$q_i=\frac{e^{z_i/T}}{\sum_j e^{z_i/T}} \tag{6}$$

该公式在标准 Softmax 的基础上增加了温度变量 T.这里,z_i 表示 Softmax 第 i 类的输入 Logits,q_i 表示 Softmax 第 i 类的输出 Soft Targets.此外,Hinton 等人还证明了使用 Logits 可以看作是当温度 T 趋向于无穷时使用 Soft Targets 的一个特例.此外,如果迁移数据是有分类标记的,那么可以同时利用分类标记和 Soft Targets 共同作为监督信息来训练学生模型,从而取得更高的准确率.

4 网络分解

神经网络中的计算代价为卷积操作所主导,模型大小为全连接层参数所主导.针对这一现状,网络分解的目的是将矩阵二维张量的奇异值分解(singular value decomposition,简称 SVD)[42]推广到三维卷积核,并且减小前向传播的时间.

4.1 卷积核的低秩表达

对于二维张量,可以简单地使用 SVD 进行低秩表达,将原始矩阵分解为上三角矩阵、对角矩阵、下三角矩阵的乘积后,取对角矩阵中最显著的几个特征值保留,达到减小参数数量的目的.对于更高阶的张量,一方面可以铺展成二维张量,使用标准 SVD 方法;另一方面,也可以使用多个一维张量外积求和逼近的方法,即,使用 3 个一维张量外积求和进行 K 次累加来逼近一个秩为 K 的三维张量.

训练好的 CNN 通常在 3 个颜色通道间的冗余度较大,因此可以将颜色维度投射到一维单色子空间,这样,卷积只需在一维空间中进行即可得到输出特征图.对于后续的卷积层,使用双聚类估计[43],输入和输出的特征被聚类成相等大小的组,权重张量在每一对输入/输出类之间进行估计.由于每一个类内的特征都具有大量相似的元素,因此更容易使用低秩表达进行近似.低秩表达压缩权重之后,会在一定程度上造成 CNN 的准确率下降.越是粗糙的近似,准确率下降越为严重,因此,需要将近似的后续卷积层固定,对前面层的卷积层进行微调.

文献[44]首先使用秩为 1 的卷积核(由于秩为 1,故可以分解为行向量与列向量乘积)作用在输入图上,产生相互独立的 M 个基本特征图,然后通过学习到的字典权重,利用线性组合重构出输出特征图.这种方法是在二维上对卷积核的近似,仅仅利用了 N 个输出通道之间的冗余性,可以利用输出与输入通道之间的冗余性推广到三维上:首先,利用列向量将每一个卷积层分解为一长宽相等的正方形卷积层;随后,再通过多个行向量重构输出特征图.类似地,文献[45]也探讨了用卷积网络中的线性分解结构来减少计算量.

文献[46]提出了稀疏卷积神经网络(sparse convolutional neural network,简称 SCNN),利用通道间和通道内冗余,结合微调步骤,最小化含有稀疏性最大化正则项的损失函数,获得了极高的稀疏性与压缩率.

在 ILSVR2012 数据集上的实验显示,这个过程在精确度损失小于 1%的情况下消减了超过 90%的参数.该方案首先在通道上进行稀疏分解,然后将高计算代价的卷积操作转换成矩阵相乘,随后再将矩阵稀疏化,接着在微调继续训练整个网络的过程中引入了在网络参数上的稀疏约束,相对于原始网络的训练过程,能够获得极大的稀疏性.整个网络使用上一步稀疏化的权重作为初始值,然后使用包含稀疏约束、低秩约束、重建约束的目

标损失函数联合优化,使得权重参数中的零元素比例不断增加(如图 7 所示).

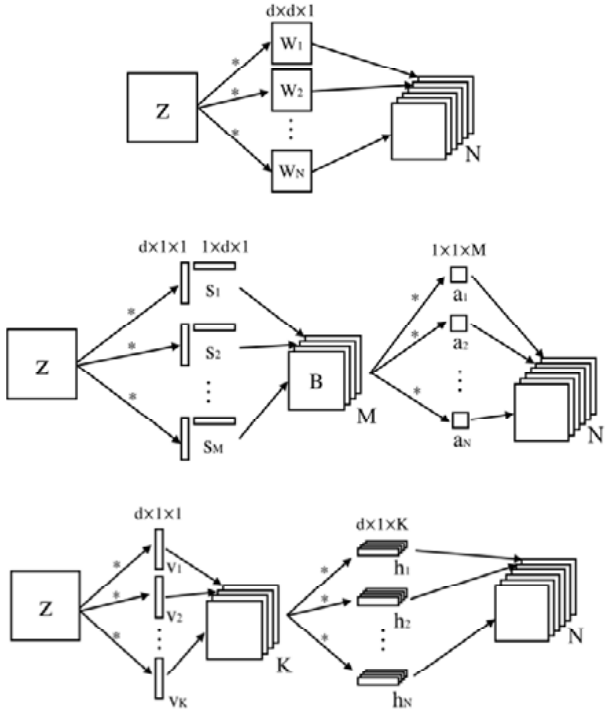


Fig.7 Decomposition of convolution kernels [44]

图 7 卷积核分解方法[44]

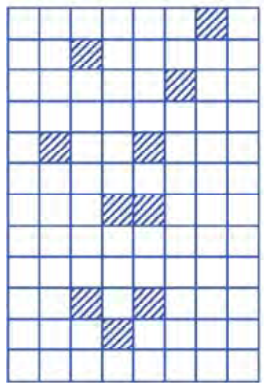
4.2 稀疏矩阵乘法

为了能够有效利用低秩表达获得的稀疏性,文献[46]提出了高效稀疏矩阵乘法的 CPU 实现算法来减小神经网络运行的时间.传统稀疏矩阵相乘,会根据非零元素的位置索引将它们连续存储在一些特定的结构中,以避免零值的存储和计算,导致在遍历矩阵的过程中只能间接跳转读取内存,比稠密情况下连续的内存读取要慢得多,成为巨大的性能瓶颈.此外,这种不规则的输入矩阵模式也很难充分发挥单指令多数据(single instruction and multiple data,简称 SIMD)微架构的威力,而 SIMD 在稠密矩阵算法中是提升性能的要素.

在完全训练后的网络中,卷积核是常数,而中间生成的特征图会随着输入图像变化.卷积核参数为稀疏矩阵,特征图为稠密矩阵.卷积核稀疏矩阵中非零元素的位置是已知的,因而能够在编译后的乘法代码中直接编码.文献[46]据此设计了一种高效的稀疏-稠密矩阵乘法算法,如公式(7)所示.

$$c_{*,j}=\sum_{i=1}^{k}\bar{a}_{*,i}\bar{b}_{i,j},\ 1\leqslant j\leqslant 8 \tag{7}$$

既然卷积核矩阵在训练完成后是固定已知的,那么其中的权重就可以被固化到代码空间中,生成串向量操作的指令.指令的源操作数和目的操作数根据非零元素所在位置确定.以如图 8 和公式(8)为例,考虑矩阵 B 为稀疏卷积核,由于 B 的列数为 8,矩阵相乘可以简化为公式(8)的向量的线性运算,同时,目的操作数向量只有 8 个.对应到图 8 中,阴影部分为非零元素,空白部分为零元素,从而目的操作数 $c_7=\bar{c}_{*,7}=a_1\times b_{1,7}$,即 B 矩阵的第 1 行中只有第 7 个元素非零.由此推断,想要获得 C 矩阵中的第 7 列,只需用 A 矩阵中的第 1 列乘以该非零元素.其余伪代码生成以此类推.



输入：

$\mathbf{A}$: 8×12 稠密矩阵

$\mathbf{B}$: 12×8 稀疏矩阵

输出：

$\mathbf{C} = \mathbf{A} \times \mathbf{B}$

操作：

$\mathbf{c}_7 += \mathbf{a}_1 \times b_{1,7}$

$\mathbf{c}_3 += \mathbf{a}_2 \times b_{2,3}$

$\mathbf{c}_6 += \mathbf{a}_3 \times b_{3,6}$

$\mathbf{c}_2 += \mathbf{a}_5 \times b_{5,2}$

$\mathbf{c}_5 += \mathbf{a}_5 \times b_{5,5}$

$\mathbf{c}_4 += \mathbf{a}_7 \times b_{7,4}$

$\mathbf{c}_5 += \mathbf{a}_7 \times b_{7,5}$

$\mathbf{c}_3 += \mathbf{a}_{10} \times b_{10,3}$

$\mathbf{c}_5 += \mathbf{a}_{10} \times b_{10,5}$

$\mathbf{c}_4 += \mathbf{a}_{11} \times b_{11,4}$

Fig.8 Multiplying a sparse matrix with a dense matrix[46]

图 8 稀疏矩阵与稠密矩阵相乘[46]

4.3 权重量化

文献[47]提出了二值化的权重策略来对网络的权重做较为极端的量化,即限制权重的取值只能是–1 或者 1.通过这种方式,可以将很多乘法转化为消耗较少的加法,极大地简化了专用于深度学习的硬件的设计.为了在使用随机梯度下降(stochastic gradient descent,简称 SGD)的过程中降低二值化权重的影响,文中提出:仅将二值化的策略用于前向和反向传播中,而在 SGD 的参数更新中仍采用足够的精度.

文献[48]通过实验证明了:通过权重量化的方式,可以获得比矩阵稀疏分解更好的效果.对于 1000 类的 ImageNet 分类任务,作者获得了 16 倍~24 倍的压缩率,同时,准确率下降小于 1%.首先,使用 *K*-均值方法量化权重,在聚类之后,对同类的权重使用同一索引和索引所对应的均值中心来表示,并存储聚类之后的类索引号和码本[49,50].从而在只需要极少额外辅助空间的条件下,获得了极大的压缩率.

进一步地,为了减小量化对模型准确率的影响,文献[51]提出了一种全新的增量式网络量化方法,将训练好的 32 位全精度 CNN 模型转化为低精度指数型模型,该模型中的权重均为 2 的指数幂,从而可以很方便地在 FPGA[52]设备上,通过二进制移位操作实现.

该方法包含 3 种独立的操作:权重划分、分组量化、再训练.

- 首先进行权重划分,该策略受到剪枝策略的启发,选取幅值较大的权重,认为它们对准确率的贡献最大,将它们划入训练组,将剩余幅值较小的权重划入量化组;同时,针对不同程度(比特数)的量化,应当采取不同增长率的量化比例.比如,对于 2 比特量化,可以采取{0.2,0.4,0.6,0.7,0.8,0.85,0.9,0.95,0.975,1}的量化比例顺序.
- 然后进行分组量化,本文通过变量长度编码量化权重,当然也可以采用文献[53]提出的随机舍入的方法.舍入的概率正比于当前权重值与舍入目标值之间的概率.
- 最后,通过再训练恢复准确率.

由于量化比例是增量式增加的,该方法也称为增量式权重量化方法.相对于全局量化的方法,该方法在一次量化操作之后通过立即重训练恢复准确率,最终达到了不损失准确率而极大地减小网络权重的目标.

方法的总览如图 9 所示,图中左侧网络为原始已训练好的网络,按照权重划分策略选取 50%(浅色连接权重)进行分组量化,剩余 50%(深色连接权重)进行重训练.重复上述操作,直到划分为量化组的权重达到 100%.

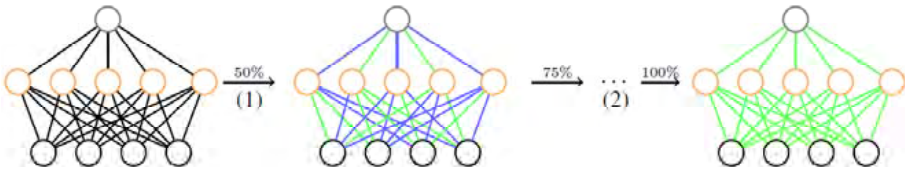


Fig.9 Incremental network quantization [51]

图 9 增量式网络量化方法[51]

5 压缩方式评价及比较

5.1 公开深度模型

自从 2012 年 AlexNet 在 ImageNet 图像分类竞赛中取得突破以来,准确率更高的深度模型层出不穷,其中,最具代表性的有 VGG-16,GoogLeNet,ResNet-101.

AlexNet 的突出贡献在于总结了一些深度网络设计的经验,比如使用 ReLU 替换非线性单元、使用 Dropout 避免过拟合、使用 MaxPooling.VGG 的贡献在于使用更小的感受野,即 3×3 的卷积核.VGG 的前几层卷积核均采用较大的通道数,为的是增大网络宽度,导致前向推断非常耗时.GoogLeNet 使用 Inception 结构来增大网络宽度,不仅减小了计算代价,更提高了准确率.ResNet 使用简单的旁路(bypass)促进信息的流动,降低了梯度消失的可能性,最终使得网络达到 152 层的深度,极大地提升了准确率.

基于 Keras 的开源深度模型在 ImageNet 分类数据集上的准确率及参数数量统计信息见表 1.

Table 1 Performance and numbers of parameters on ImageNet in typical deep models on Keras

表 1 Keras 框架下深度模型在 ImageNet 上的性能及参数数量

深度模型	Top-5 错误率(%)	可训练参数数量(M)
AlexNet	18.90	60.9
VGG_CNN_F	16.70	60
VGG-16	7.32	138.4
GoogLeNet	6.60	23.6
ResNet-50	6.71	23.7
ResNet-101	6.05	42.7
ResNet-152	4.49	58.5

一些使用多个模型聚合的结果[54]能够获得更低的错误率.为了方便比较,本文统计信息为单模型在 Top-5 指标上的错误率.可训练参数数量来自开源 Keras 模型的统计结果.

5.2 总体压缩效用评价指标

网络压缩评价指标包括运行效率、参数压缩率、准确率.与基准模型比较衡量性能提升时,可以使用提升倍数(speedup)或提升比例(ratio),两者可以相互转换,本文统一使用提升比例.

目前,大部分研究工作均会测量 Top-1 准确率,只有在 ImageNet 这类大型数据集上才会只用 Top-5 准确率.为方便比较,本文在后续的效果对比表中使用 Top-1 准确率.参数压缩率的评价指标较为统一,统计网络中所有可训练的参数,根据机器浮点精度转换为字节(byte)量纲,通常保留两位有效数字以作近似估计.在网络运行效率方面,可以从网络所含浮点运算次数(FLOP)、网络所含乘法运算次数(MULTS)或随机实验测得的网络平均前向传播所需时间这 3 个角度来评价.虽然 3 种指标都反映了网络运行效率的提升,但是相互之间不可比较,因此表 2 会在运行效率中说明所使用的指标.

此外,在网络蒸馏中有一类研究方案比较特别.这类研究通过使用庞大的模型蒸馏出多个专家模型并且作并行预测,最终将投票结果聚合,不仅提高准确率,同时提高运行效率.对于单个专家模型而言,模型权重数目是下降的,但是权重数目对于所有模型的总和相对于单个庞大的原始模型是增加的.文献[33]基于这一想法,在规模庞大的 JET 数据集上将准确率由 43%提升为 45.90%.

Table 2 Overall compression performance

表 2 总体压缩效果

参考文献	模型	Top-1 准确率(%)	参数(byte)	压缩率(%)	运行效率提升
Ref.[24]	VGG-16 基准	93.25	1.50E+07	–	使用 FLOP 评价指标,对不同深度模型,分别获得 34.20%,13.70%,38.60%, 15.5%的加速
	VGG-16 压缩	93.4	5.40E+06	64.00	
	ResNet-56 基准	93.04	8.60E+05	–	
	ResNet-56 压缩	93.06	7.30E+05	15.12	
	ResNet-110 基准	93.53	1.72E+06	–	
	ResNet-110 压缩	93.3	1.16E+06	32.56	
	ResNet-34 基准	73.23	2.16E+07	–	
	ResNet-34 压缩	72.56	1.99E+07	7.87	
Ref.[20]	LeNet 基准	99.06	1.96E+07	–	由于全连接层对网络运行时间影响不大,所以对其剪枝不能显著提升运行效率
	基于幅度剪枝	96.5	1.65E+07	16.01	
	随机剪枝	91.37	1.65E+07	16.01	
	数据无关剪枝	98.35	1.65E+07	16.01	
	AlexNet 基准	57.84	6.09E+07	–	
	对 FC6 数据无关剪枝	56.08	4.23E+07	30.57	
	对 FC7 数据无关剪枝	56	5.37E+07	11.80	
	对 FC6 和 FC7 数据无关剪枝	55.6	3.97E+07	34.89	
Ref.[25]	基于 CASIA 数据的基准模型	86.04	1.30E+04	–	使用实验测得平均前向传播所需时间作为评价指标,对于不同深度模型,分别获得 59.51%, 25.93%,37.11%,57.81%, 62.26%的加速
	Fitnet	82.08	8.64E+03	33.33	
	低秩分解	84.79	1.17E+04	9.91	
	Inbound Prune	84.74	1.16E+04	10.71	
	RR Prune	85.08	6.14E+03	52.61	
	Hyb. Prunne	85.32	6.06E+03	53.27	
Ref.[22]	LeNet-5 基准	80	4.31E+05	–	–
	剪枝+量化	77	3.60E+04	91.67	
	AlexNet 基准	57.22	6.10E+07	–	
	剪枝+量化	57.22	6.70E+06	88.89	
	VGG-16 基准	31.5	1.38E+08	–	
	剪枝+量化	31.34	1.03E+07	92.31	
Ref.[21]	AlexNet	57.22	2.40E+08	–	引入了非结构化的稀疏性,需要专门的软件计算库或者未来的硬件获得运行效率的提升
	Fastfood-32-AD	58.07	1.31E+02	50.00	
	Fastfood-16-AD	57.1	6.40E+07	72.97	
	Collins&Kohli	55.6	6.10E+07	75.00	
	SVD	55.98	4.78E+07	80.00	
	剪枝	57.22	8.90E+06	88.89	
	剪枝+量化	57.22	6.90E+06	96.30	
	剪枝+量化+编码	57.22	6.90E+06	98.97	
	VGG-16 基准	68.5	5.52E+08	–	
	VGG-16 压缩	68.83	1.13E+07	97.95	
Ref.[37]	Teacher	90.18	9.00E+06	–	使用 MULT 评价指标,对 FitNet1,FitNet2,FitNet3, FitNet4, 分别获得了 92.51%,78.45%,27.01%, 34.21%的加速(FitNet1~ FitNet4 采用不同的人工设计的网络架构)
	FitNet1	89.01	2.50E+05	97	
	FitNet2	91.06	8.62E+05	90.42	
	FitNet3	91.1	1.60E+06	82.22	
	FitNet4	91.61	2.50E+06	72.22	
	Mimic single	84.6	5.40E+07	–	
	Mimic ensemble	85.8	7.00E+07	–	

5.3 分层压缩效用评价

在网络分解的研究中,针对深度网络中的卷积层和全连接层通常采用不同的方法.因而,逐层分析可以帮助我们获得一些经验性的规律.逐层分析实验包含两种方法:(1) 逐层对照实验:固定其余所有层,每次对一层进行压缩.从而可以对压缩率、效率提升进行全面的评价,见表 3;(2) 只进行一次实验,统计压缩前和压缩后的各层参数数量.从而获得压缩率指标,见表 4.

在网络分解的逐层对照实验中,对卷积层的压缩率在不断的改进中不断提升,但是总体上小于对全连接层的 SVD 分解.稀疏卷积操作采用固定卷积核矩阵的方法,对训练好的卷积核矩阵的稀疏性具有一定的依赖性,

但总体上更好地利用了卷积核的稀疏性,提升了运行效率.

Table 3 Performance evaluations of compressing single convolutional layer or fully-connected layer

表 3 卷积层和全连接层逐层压缩对照实验

参考文献	压缩层类别	压缩方法	压缩率(%)	准确率下降比(%)	运行加速比(%)
Ref.[45]	Conv	Monochromatic, C=4	87.90	1.90	66.33
	Conv	Monochromatic, C=6	84.52	0.43	66.10
	Conv	Monochromatic, C=8	81.13	0.20	65.99
	Conv	Monochromatic, C=12	74.36	0	65.64
	Conv	双聚类+外积分解	92.54	0.68	23.07
	Conv	双聚类+SVD,	71.42	0.90	37.50
	FC	SVD, K=250	92.54	0.84	20.19
	FC	SVD, K=950	71.42	0.09	17.31
Ref.[46]	Conv	Sparse CNN, kernel size=11	92.70	0.62	61.69
	Conv	Sparse CNN, kernel size=5	95.00	1.43	85.99
	Conv	Low rank, kernel size=5	89.00	0.61	60.00

在权重数量上,Han 进行了系统的实验,从中可以总结出一些经验性的结论.全连接层参数密集,不仅参数数量比卷积层高出一个数量级,而且对最终压缩率的贡献也远超过卷积层.同时,文献[22]中的方法进一步改进后,采用第 4 节中提到的量化编码方式,最高可以达到 97.95%的压缩率.

Table 4 Compression rate of convolutional layers and fully-connected layers

表 4 卷积层和全连接层的压缩率

参考文献	压缩层类别	压缩方法	压缩率(%)	总体压缩率(%)
Ref.[21]	AlexNet,Conv	剪枝+量化	67.40	88.89
	AlexNet,Conv	剪枝+量化+编码	79.47	
	AlexNet,FC	剪枝+量化	97.00	
	AlexNet,FC	剪枝+量化+编码	97.61	
	VGGNet,Conv	剪枝+量化	60.00	92.31
	VGGNet,Conv	剪枝+量化+编码	70.03	
	VGGNet,FC	剪枝+量化	98.40	
	VGGNet,FC	剪枝+量化+编码	98.90	
Ref.[22]	VGGNet,Conv	剪枝	76.20	98.97
	VGGNet,FC	剪枝	89.80	97.95

6 未来研究方向

网络剪枝、网络精馏和网络分解都能在一定程度上实现网络压缩的目的.回归到深度网络压缩的本质目的上,即提取网络中的有用信息,以下是一些值得研究和探寻的方向.

(1) 权重参数对结果的影响度量.深度网络的最终结果是由全部的权重参数共同作用形成的,目前,关于单个卷积核/卷积核权重的重要性的度量仍然是比较简单的方式,尽管文献[14]中给出了更为细节的分析,但是由于计算难度大,并不实用.因此,如何通过更有效的方式来近似度量单个参数对模型的影响,具有重要意义.

(2) 学生网络结构的构造.学生网络的结构构造目前仍然是由人工指定的,然而,不同的学生网络结构的训练难度不同,最终能够达到的效果也有差异.因此,如何根据教师网络结构设计合理的网络结构在精简模型的条件下获取较高的模型性能,是未来的一个研究重点.

(3) 参数重建的硬件架构支持.通过分解网络可以无损地获取压缩模型,在一些对性能要求高的场景中是非常重要的.然而,参数的重建步骤会拖累预测阶段的时间开销,如何通过硬件的支持加速这一重建过程,将是未来的一个研究方向.

(4) 任务或使用场景层面的压缩.大型网络通常是在量级较大的数据集上训练完成的,比如,在 ImageNet 上训练的模型具备对 1 000 类物体的分类,但在一些具体场景的应用中,可能仅需要一个能识别其中

几类的小型模型.因此,如何从一个全功能的网络压缩得到部分功能的子网络,能够适应很多实际应用场景的需求.

(5) 网络压缩效用的评价.目前,对各类深度网络压缩算法的评价是比较零碎的,侧重于和被压缩的大型网络在参数量和运行时间上的比较.未来的研究可以从提出更加泛化的压缩评价标准出发,一方面平衡运行速度和模型大小在不同应用场景下的影响;另一方面,可以从模型本身的结构性出发,对压缩后的模型进行评价.

7 结束语

网络压缩旨在减少模型参数、降低存储空间和减少运算开销,在深度网络的实际应用中发挥着越来越重要的作用.本文总结了网络剪枝、网络精馏和网络分解这 3 个方向的压缩方法,并在压缩性能上提供了相应指标评价.其中,网络剪枝关注于去掉模型中影响较小的卷积结构;网络精馏侧重于训练一个较小的学生网络结构去模拟较大的教师网络的性能;网络分解强调从数据存储方式和结构以及运算优化的角度来降低开销.希望通过以上介绍,读者能够对深度网络压缩有一个较为全面的了解,并在相关基于深度模型的实际任务中加以利用.

References:

[1] Le Cun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015,521(7553):436−444. [doi: 10.1038/nature14539]

[2] Plamondon R, Srihari SN. Online and off-line handwriting recognition: A comprehensive survey. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000,22(1):63−84. [doi: 10.1109/34.824821]

[3] Wan J, Wang D, Hoi SCH, Wu P, Zhu J, Zhang Y, Li J. Deep learning for content-based image retrieval: A comprehensive study. In: Proc. of the 22nd ACM Int'l Conf. on Multimedia (MM). Orlando: ACM Press, 2014. 157−166. [doi: 10.1145/2647868.2654948]

[4] Girshick R. Fast *r*-cnn. In: Proc. of the IEEE Int'l Conf. on Computer Vision (ICCV). Santiago: IEEE, 2015. 1440−1448. [doi: 10.1109/iccv.2015.169]

[5] Wang N, Yeung DY. Learning a deep compact image representation for visual tracking. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Tahoe: IEEE, 2013. 809−817.

[6] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: Proc. of the 38th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Santiago: ACM Press, 2015. 373−382. [doi: 10.1145/2766462.2767738]

[7] Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. On optimization methods for deep learning. In: Proc. of the 28th Int'l Conf. on Machine Learning (ICML). Bellevue: ACM Press, 2011. 265−272.

[8] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Tahoe: IEEE, 2012. 1097−1105.

[9] Sercu T, Puhrsch C, Kingsbury B, Le Cun Y. Very deep multilingual convolutional neural networks for LVCSR. In: Proc. of the Acoustics, Speech and Signal Processing (ICASSP). Shanghai: IEEE, 2016. 4955−4959. [doi: 10.1109/icassp.2016.7472620]

[10] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770−778. [doi: 10.1109/cvpr.2016.90]

[11] Setiono R, Liu H. Neural-Network feature selector. IEEE Trans. on Neural Networks, 1997,8(3):654−662. [doi: 10.1109/72.572104]

[12] Hanson SJ, Pratt LY. Comparing biases for minimal network construction with back-propagation. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Denver: IEEE, 1989. 177−185.

[13] Whitley D, Starkweather T, Bogart C. Genetic algorithms and neural networks: Optimizing connections and connectivity. Parallel Computing, 1990,14(3):347−361. [doi: 10.1016/0167-8191(90)90086-o]

[14] Oberman SF, Flynn MJ. Design issues in division and other floating-point operations. IEEE Trans. on Computers, 1997,46(2):154−161. [doi: 10.1109/12.565590]

[15] Anwar S, Sung WY. Coarse pruning of convolutional neural networks with random masks. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 134−145.

[16] Le Cun Y, Denker JS, Solla SA. Optimal brain damage. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Denver: IEEE, 1989. 598−605.

[17] Rosenblueth E. Point estimates for probability moments. Proc. of the National Academy of Sciences, 1975,72(10):3812−3814. [doi: 10.1073/pnas.72.10.3812]

[18] Hassibi B, Stork DG, Wolff GJ. Optimal brain surgeon and general network pruning. In: Proc. of the Int'l Conf. on Neural Networks (ICNN). San Francisco: IEEE, 1993. 293−299. [doi: 10.1109/icnn.1993.298572]

[19] Hassibi B, Stork DG. Second order derivatives for network pruning: Optimal brain surgeon. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Denver: IEEE, 1993. 164−171.

[20] Srinivas S, Babu RV. Data-Free parameter pruning for deep neural networks. In: Proc. of the 26th British Machine Vision Conf. (BMVC). Swansea: IEEE, 2015. 120−129. [doi: 10.5244/c.29.31]

[21] Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). San Juan: IEEE, 2016. 233−242.

[22] Han S, Pool J, Tran J, Dally WJ. Learning both weights and connections for efficient neural network. In: Proc. of the Advances in Neural Information Processing Systems. Montreal: IEEE, 2015. 1135−1143.

[23] Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2017,13(3):Article No.32. [doi: 10.1145/3005348]

[24] Li H, Kadav A, Durdanovic I, Samet H, Graf HP. Pruning filters for efficient ConvNets. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 34−42.

[25] Polyak A, Wolf L. Channel-Level acceleration of deep face representations. IEEE Access, 2015,3:2163−2175. [doi: 10.1109/access.2015.2494536]

[26] Figurnov M, Ibraimova A, Vetrov DP, Kohli P. PerforatedCNNs: Acceleration through elimination of redundant convolutions. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Barcelona: IEEE, 2016. 947−955.

[27] Hu H, Peng R, Tai YW, Tang CK. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 214−222.

[28] Molchanov P, Tyree S, Karras T, Aila T, Kautz J. Pruning convolutional neural networks for resource efficient transfer learning. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 324−332.

[29] Rueda FM, Grzeszick R, Fink GA. Neuron pruning for compressing deep networks using maxout architectures. In: Proc. of the German Conf. on Pattern Recognition (GCPR). Saarbrücken: Springer-Verlag, 2017. 110−120. [doi: 10.1007/978-3-319-66709-6_15]

[30] Zhou ZH. Rule extraction: Using neural networks or for neural networks? Journal of Computer Science and Technology, 2004, 19(2):249−253. [doi: 10.1007/BF02944803]

[31] Zhou ZH, Jiang Y. NeC4.5: Neural ensemble based C4.5. IEEE Trans. on Knowledge and Data Engineering, 2004,16(6):770−773.

[32] Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Philadelphia: ACM Press, 2006. 535−541. [doi: 10.1145/1150402.1150464]

[33] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Montrea: IEEE, 2014. 2644−2652.

[34] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans. on Knowledge and Data Engineering, 2010,22(10):1345−1359. [doi: 10.1109/TKDE.2009.191]

[35] Lowd D, Domingos P. Naive Bayes models for probability estimation. In: Proc. of the 22nd Int'l Conf. on Machine Learning (ICML). Bonn: ACM Press, 2005. 529−536. [doi: 10.1145/1102351.1102418]

[36] Ba J, Caruana R. Do deep nets really need to be deep? In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Montrea: IEEE, 2014. 2654−2662.

[37] Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y. Fitnets: Hints for thin deep nets. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 124−133.

[38] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston: IEEE, 2015. 1−9. [doi: 10.1109/cvpr.2015.7298594]

[39] Chen T, Goodfellow I, Shlens J. Net2net: Accelerating learning via knowledge transfer. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). San Juan: IEEE, 2016. 27−35.

[40] Li Z, Hoiem D. Learning without forgetting. In: Proc. of the European Conf. on Computer Vision (ECCV). Amsterdam: Springer Int'l Publishing, 2016. 614−629. [doi: 10.1007/978-3-319-46493-0_37]

[41] He ZF, Yang M, Liu HD. Joint learning of multi-label classification and label correlations. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):1967−1981 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4634.htm [doi: 10.13328/j.cnki.jos.004634]

[42] Golub GH, Reinsch C. Singular value decomposition and least squares solutions. Numerische Mathematik, 1970,14(5):403−420. [doi: 10.1007/BF02163027]

[43] Zhang M, Ge WH. Overlap bicuster algorithm based on probability. Computer Engineering and Design, 2012,33(9):3579−3583 (in Chinese with English abstract). [doi: 10.16208/j.issn1000-7024.2012.09.046]

[44] Jaderberg M, Vedaldi A, Zisserman A. Speeding up convolutional neural networks with low rank expansions. In: Proc. of the 26th British Machine Vision Conf. (BMVC). Swansea: IEEE, 2015. 100−109. [doi: 10.5244/c.28.88]

[45] Denton EL, Zaremba W, Bruna J, Le Cun Y, Fergus R. Exploiting linear structure within convolutional networks for efficient evaluation. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Montrea: IEEE, 2014. 1269−1277.

[46] Liu B, Wang M, Foroosh H, Tappen M, Penksy M. Sparse convolutional neural networks. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston: IEEE, 2015. 806−814. [doi: 10.1109/cvpr.2015.7298681]

[47] Courbariaux M, Bengio Y, David JP. Binaryconnect: Training deep neural networks with binary weights during propagations. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). Montreal: IEEE, 2015. 3123−3131.

[48] Gong Y, Liu L, Yang M, Bourdev L. Compressing deep convolutional networks using vector quantization. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). Toronto: IEEE, 2015. 102−110.

[49] Lee H, Battle A, Raina R, Ng AY. Efficient sparse coding algorithms. In: Proc. of the Advances in Neural Information Processing Systems (NIPS). IEEE, 2007. 789−801.

[50] Mairal J, Bach F, Ponce J, Sapiro G. Online dictionary learning for sparse coding. In: Proc. of the 26th Annual Int'l Conf. on Machine Learning (ICML). Montreal: ACM Press, 2009. 689−696. [doi: 10.1145/1553374.1553463]

[51] Zhou A, Yao A, Guo Y, Xu L, Chen Y. Incremental network quantization: Towards lossless cnns with low-precision weights. In: Proc. of the Int'l Conf. on Learning and Representation (ICLR). IEEE, 2017. 154−162.

[52] Monmasson E, Cirstea MN. FPGA design methodology for industrial control systems—A review. IEEE Trans. on Industrial Electronics, 2007,54(4):1824−1842. [doi: 10.1109/tie.2007.898281]

[53] Gupta S, Agrawal A, Gopalakrishnan K, Narayanan P. Deep learning with limited numerical precision. In: Proc. of the Int'l Conf. on Machine Learning (ICML). Lille: ACM Press, 2015. 1737−1746.

[54] Antipov G, Berrani SA, Dugelay JL. Minimalistic CNN-based ensemble model for gender prediction from face images. Pattern Recognition Letters, 2016,70:59−65. [doi: 10.1016/j.patrec.2015.11.011]

附中文参考文献:

[41] 何志芬,杨明,刘会东.多标记分类和标记相关性的联合学习.软件学报,2014,25(9):1967−1981. http://www.jos.org.cn/1000-9825/4634.htm [doi: 10.13328/j.cnki.jos.004634]

[43] 张敏,戈文航.基于概率计算的重叠双聚类算法.计算机工程与设计,2012,33(9):3579−3583. [doi: 10.16208/j.issn1000-7024.2012.09.046]

雷杰(1991－),男,湖北仙桃人,博士生,主要研究领域为计算机视觉,深度学习.

高鑫(1992－),男,硕士生,主要研究领域为计算机视觉,深度学习.

宋杰(1991－),男,博士生,主要研究领域为计算机视觉,深度学习.

王兴路(1996－),男,本科生,主要研究领域为计算机视觉,深度学习.

宋明黎(1976－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为计算机视觉,深度学习.