基于自适应层信息熵的卷积神经网络压缩

魏钰轩,陈 莹
(江南大学轻工过程先进控制教育部重点实验室,江苏无锡 214122)

摘 要: 网络剪枝是一种有效的卷积神经网络压缩方法. 多数现有压缩方法因迭代剪枝了“不重要”的网络结构,一方面破坏了网络结构的信息整体性,另一方面其迭代操作耗费了大量的计算资源与时间. 为了解决上述问题,论文从网络结构全局考虑,提出基于自适应层信息熵的卷积神经网络压缩方法. 首先,在获取压缩网络结构的过程中,本文设计了一种端到端的结构化网络剪枝方案,将卷积层看作一个整体,利用层信息熵之间的关联性直接确定各卷积层过滤器的保留率,避免迭代剪枝训练造成的信息损失. 其次,对剪裁后的网络进行重训练时,综合考虑压缩过程中使用的层信息熵指标,通过对卷积层与批归一化(Batch Normalization,BN)层进行自适应联合嫁接,让网络学习到更多的信息,提升网络性能. 针对3种主流网络在不同的数据集上进行了实验,验证了所提方法的有效性与优越性. 例如在CIFAR-10上,针对ResNet-56,相比于基线网络,在计算量压缩36.2%的情况下,本文方法准确率提升了1%;针对ResNet-110,在计算量压缩52.4%的情况下,本文方法准确率提升了1.42%;针对轻量型网络MobileNetV2,在计算量压缩55.2%的情况下,本文方法准确率提升了1.29%.

关键词: 卷积神经网络; 网络剪枝; 信息熵; 嫁接; 模型压缩
中图分类号: TP391.41 **文献标识码:** A **文章编号:** 0372-2112(2022)10-2398-11
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20201372

Convolutional Neural Network Compression Based on Adaptive Layer Entropy

WEI Yu-xuan, CHEN Ying
(*The Key Laboratory of Advanced Process Control for Light Industry* (*Ministry of Education*), *Jiangnan University*, *Wuxi*, *Jiangsu* 214122, *China*)

Abstract: Network pruning has proven to be an effective approach to compress convolutional neural network (CNN). However, most existing CNN compression methods iteratively prune the "least important" filters, which not only destroys the information integrity of network structures, but also results in significant computation cost due to the iterative operation. To solve the problems, a convolutional neural network compression method based on adaptive layer entropy(ALE) is proposed, considering a global network structure. Firstly, an end-to-end network pruning strategy is designed, in which the retention rate of each convolutional layer filter is directly determined based on the entropy correlation between layers. The pruning strategy takes the convolutional layer as a whole, which decreases the information loss and computation cost of iterative pruning. Then, an adaptive joint grafting method, in which both convolutional and batch normalization(BN) layers are considered, is presented for the pruned network retraining to learn more information from the network. The layer entropies used in the compression are also utilized for the grafting. Experiments are conducted on different benchmarks and three popular networks, which demonstrate the efficiency and superiority of the proposed ALE over other methods. For the experiments on CIFAR-10, ALE achieves 36.2%, 52.4% and 55.2% pruned rate in FLOPs for ResNet-56, ResNet-110 and MobileNetV2 while with increase of 1%, 1.42%, 1.29% accuracy respectively.

Key words: convolutional neural network; network pruning; entropy; grafting; model compression

1 引言

机器学习模型的复杂度与其学习能力息息相关. 为了在目标分类、检测与识别任务中取得更突出的表现[1~3],常采用更深更宽的卷积神经网络(Convolutional

收稿日期:2020-12-01;修回日期:2021-01-29;责任编辑:王天慧
基金项目:国家自然科学基金(No.61573168)

Neural Network,CNN). 然而,这些方法急剧增加了网络的计算量和参数量,同时,由于移动嵌入式设备计算能力有限、存储不足,这些CNN模型很难部署在嵌入式设备上. 为了解决这一问题,将CNN模型压缩受到了人们的广泛关注.

模型压缩方法主要分为以下几种:网络剪枝[4]、参数量化[5]、知识蒸馏[6]、低秩分解[7]以及紧凑型网络设计[8]. 网络剪枝因具有以下2个优点而获得了广泛关注:(1)网络剪枝与低秩分解、参数量化等其他模型压缩方法正交,可以综合使用从而获得更好的表现;(2)网络剪枝可以被应用于任何复杂的卷积神经网络结构,包括residual block[9],inception module[10]等.

网络剪枝分为非结构化剪枝与结构化剪枝. 非结构化剪枝又称为细粒度剪枝,通过剪枝网络的神经元来压缩网络,剪枝某个神经元就是将该神经元的值设置为0,本质上是一种稀疏化的过程. 这种方法可以通过稀疏化存储方式减少内存占用,压缩网络存储体积,但没有减少计算量. 同时,这种剪枝方式会导致不规则的内存访问,对网络的在线推理效率产生负面影响,需要特殊的软硬件进行加速. 例如,Miguel等[11]将剪枝权重后最小化损失问题转化为优化问题. Han等[12]将权重绝对值大小作为剪枝不重要权重的指标.

结构化剪枝是一种移除网络中整个冗余特征图或过滤器的网络剪枝方法,剪枝后的网络受到各种离线深度学习平台的支持. 尽管相对于非结构化剪枝,结构化剪枝压缩率较低,但结构化剪枝可以降低模型在设备上的内存占用,减少网络的计算量,加速网络推理. 根据是否依赖训练数据,结构化剪枝方法分为数据依赖型和数据独立型2种. 数据依赖型需要使用训练数据来选择哪些过滤器进行剪枝. 例如,Wen等[13]提出组稀疏化的概念,将$l_{2,1}$正则化作用于过滤器,训练生成带类别标签的稀疏化网络;Luo等[14]利用网络下一层的统计信息来指导过滤器剪枝;Wang等[15]采用特征图空间聚类的方法来剪枝每个卷积层的特征图和过滤器. 数据独立型的剪枝不依赖网络训练数据. 例如,Li等[16]采用权重绝对值的方法来衡量网络结构的重要性,然后采用贪心算法移除权重绝对值小的部分;Ye等[17]利用BN层的缩放输出强迫网络更加稀疏;Zhuo等[18]对过滤器进行谱聚类来挑选冗余过滤器.

上面提到的这些方法通常采用多步优化、逐层迭代剪枝的方案,再通过大量重训练恢复网络精度. 寻找最佳网络结构时,逐层剪枝与训练,低效耗时. Ye等[17]重新思考了“Smaller-Norm-Less-Important(SNLI)”准则在通道剪枝中使用的必要性,认为不使用该准则也能获得好的结果. He等[19]指出SNLI准则的使用存在诸多局限. Liu等[20]认为,网络结构决定网络性能. 利用逐层迭代剪枝方式,没有考虑到被剪枝网络层与层结构的整体性和网络全局信息的关联性,使得压缩后的网络结构通常是次优的方案,即使经过大量的重训练也无法获得更好的表现. 近年来,人们研究表明[21~23],从网络全局出发,综合考虑网络结构之间的关联性能获得更好的性能. Lin等[21]利用生成对抗学习,通过努力对齐基线网络与生成器网络的输出,每轮训练直接剪枝整个网络中的冗余结构. Lin等[22]利用智能群算法寻找网络最优结构,同样是通过每轮训练寻找全局网络结构. 虽然这些端到端迭代方案在每次剪枝过程中考虑了剪枝后网络各层结构的整体性与关联性,但在获取第一个剪枝网络结构时,网络各层过滤器被剪枝的个数是随机的,破坏了原始网络与被剪枝网络相同层之间的关联性,需要经过少量的重训练来尽量恢复被剪枝网络不同层之间信息的关联性,然而仅经过少量的重训练,各结构之间信息关联性重建不够完全,在此基础上,采用各种优化方法寻求压缩网络结构,会导致在下一次更新时,形成“次优方案迭代次优方案”. 同时,不论是逐层迭代剪枝还是端到端迭代剪枝,剪枝过程与微调过程缺少联系,未能建立二者的联合机制,进而影响模型压缩表现.

为了解决迭代剪枝出现的问题,本文提出一种基于自适应层信息熵(Adaptive Layer Entropy,ALE)的模型压缩方法,该方法利用层信息熵建立模型剪枝过程中剪枝与重训练的联系,这是一种不需要迭代剪枝的端到端结构化剪枝方案:首先,利用预训练模型的层信息熵直接确定剪枝网络结构,避免迭代剪枝;重训练时,基于层信息熵建立卷积层与BN层的联合嫁接机制,丰富网络信息,获得性能优良的剪枝网络. ALE适用于不同的网络结构与网络深度,可以根据不同硬件限制和压缩需求快速获取满足要求的模型. 本文的主要贡献有以下3点:

(1)提出一种基于自适应层信息熵的卷积神经网络压缩方法,利用层信息熵之间的关联性直接确定各卷积层过滤器的保留率,避免迭代剪枝训练造成的信息损失;

(2)利用层信息熵建立模型剪枝过程中剪枝与重训练的联系,通过对卷积层与批归一化层进行自适应联合嫁接,增加网络中学习到的信息多样性,提升剪枝后模型微调的精度;

(3)针对3种主流卷积神经网络,在3个标准分类数据集和2个行人重识别数据集上进行了实验,均验证了所提方法的有效性与优越性,尤其是在具有短连接的网络结构中,如ResNet与MobileNetV2,均获得了优异表现.

2 网络剪枝总体方案

2.1 迭代剪枝过程及问题分析

网络剪枝的过程主要包括2种:逐层迭代剪枝和端到端迭代剪枝. 逐层迭代剪枝过程如图1所示:首先,训练初始化网络,选取精度最高的模型作为预训练网络;然后,利用算法逐层衡量网络中各结构的重要性,裁减掉算法认为该层不重要的部分,剩余的部分形成新的网络结构,并通过少量重训练恢复剪枝后的网络精度,每层网络剪枝时通常需要重复多次该步骤,以实现在尽可能小的精度损失下裁剪更多的网络结构;最后,将剪枝后网络结构进行大量重训练,恢复网络精度.

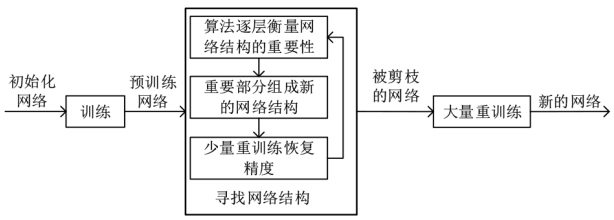

图1 逐层迭代剪枝过程

上述剪枝算法每次只针对网络的某一层进行剪枝,剪枝整个网络需要耗费大量的计算资源与时间,同时也没有考虑到网络层与层结构之间的整体性. 为了解决这些问题,端到端迭代剪枝方式成为近些年的研究热点,其过程如图2所示,该过程最大的创新在于算法直接将剪枝扩大到全局网络结构,在每次更新网络结构时,利用各种优化方法同时对所有层的网络结构进行更新,这种方法考虑了全局网络信息的整体性,在寻找网络结构的过程中需要耗费的资源相对于逐层迭代剪枝方案要少一些.

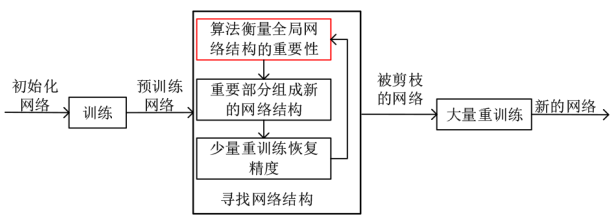

图2 端到端迭代剪枝过程

然而,端到端迭代剪枝方案每次更新网络结构时,仅通过少量重训练,在尚未重建好网络性能的情况下使用各种优化方法,极易形成“次优方案迭代次优方案”的情况. 不论是逐层迭代剪枝方案还是端到端迭代剪枝方案,在首次改变网络结构后,均破坏了耗费大量资源训练得到的预训练模型信息的整体性,在之后的迭代剪枝过程中,预训练模型的作用微乎其微. 同时,在该剪枝方案中,寻找网络结构与微调之间为分离的2个过程,二者之间缺少联系的纽带,将破坏剪枝方案在优化过程中的整体性.

2.2 ALE总体流程

为了解决上述迭代剪枝出现的问题,本文提出一种端到端的结构化网络剪枝方案,通过层信息熵直接获取剪枝后网络结构,再利用层信息熵进行联合嫁接重训练,建立起模型剪枝过程中剪枝与重训练的联系,适用于不同的网络结构与网络深度,可以根据不同硬件限制和压缩需求快速获取满足要求的模型,其整体架构如图3所示. ALE方法主要包括以下2部分内容.

(1)基于层信息熵确定被剪枝的网络结构:利用预训练网络的模型参数,通过网络层信息熵之间的大小关系确定各卷积层过滤器个数的保留率,形成被剪枝的网络.

(2)基于层信息熵联合嫁接恢复网络精度:对剪枝后的网络并行重训练,利用层信息熵之间的大小关系,每轮更新时,将并行训练的网络相同卷积层与BN层,通过自适应联合嫁接的方法,提升剪枝后的网络精度.

图3 ALE总体流程框架

该方法利用层信息熵将网络剪枝与重训练联系起来,剪枝过程中,ALE依据层信息熵指标获取被压缩的网络结构,被剪枝的网络损失了部分信息,但在联合嫁接重训练过程中,针对被压缩的网络结构,依据层信息熵指标进行自适应联合嫁接,让被压缩的网络获取丰富的信息,恢复网络精度. ALE是一种不需要迭代剪枝的端到端结构化剪枝方案,将更多的计算资源用作提升剪枝后网络精度的训练.

3 ALE具体实施方案

本文提出一种基于自适应层信息熵的模型压缩方法ALE,表1中给出了ALE方法中主要符号定义及描述.

表1 ALE中符号定义及描述

符号	描述	符号	描述
N	卷积神经网络模型	OHS	模型原始层信息熵区间
L	卷积层个数	$\alpha_{\max}$	卷积层过滤器最大保留率
c_i	第i层过滤器个数	HS	模型层信息熵区间
W_i	第i层网络的权重	α_i	第i层过滤器保留率
$H(\bullet)$	信息熵	$N_i\_l_1$	网络N_i的卷积第一层
β	自适应嫁接系数	$N_i\_b_1$	网络N_i的BN第一层
SE	最小信息熵	p_k^i	$\boldsymbol{W}_i$落在第k份区间的概率
BE	最大信息熵	$c_{\hat{i}}$	剪枝后第i层过滤器的个数

3.1 层信息熵

假设一个卷积神经网络模型N有L个卷积层,用$C=(c_1,c_2,\cdots,c_L)$表示各卷积层过滤器个数的组合,$\boldsymbol{W}_i\in\boldsymbol{R}^{N_i\times N_{i+1}\times K\times K}$表示第$i$层网络的权重,则第$i$层网络权重的信息熵定义如下:

$$H\left(\boldsymbol{W}_i\right)=-\sum_{k=1}^{B}p_k^i\log_2 p_k^i \tag{1}$$

其中,第i层网络权重$\boldsymbol{W}_i$范围被等分成B份,落在第k份区间的可能性用p_k^i表示,$H\left(\boldsymbol{W}_i\right)$越小,说明这一层权重的变化越小,信息量越少,可压缩的比率越大. 大量优秀的工作表明,将信息熵引入神经网络衡量网络权重的变化是可取的. Meng等[24]提出基于网络卷积层信息熵的方法提升网络精度;Luo等[25]逐个计算过滤器的信息熵,迭代剪枝信息熵小的过滤器;Li等[26]利用核稀疏性与熵来解释卷积神经网络压缩的可行性.

在获取网络预训练模型的过程中,发现网络相同层信息熵大小在迭代过程中大小趋于稳定,且不同层信息熵大小比例关系渐进平稳. 以网络ResNet-56在CIFAR-100上为例,使用初始ResNet-56网络,直接在CIFAR-100上训练100轮,每轮保存一个模型参数,分别计算各残差块中第一层的信息熵,每十轮计算一次层信息熵均值与标准差,如图4所示,图中shortcut_i表示ResNet-56第i个残差块中的第一层,横坐标表示迭代轮数,纵坐标表示层信息熵大小,五角星点表示shortcut_i每十轮层信息熵平均值,竖线表示该点处的标准差. ResNet-56中有27个残差块,为了更好地反应整个网络的信息熵分布情况,每间隔3个残差块选取一次进行分析.

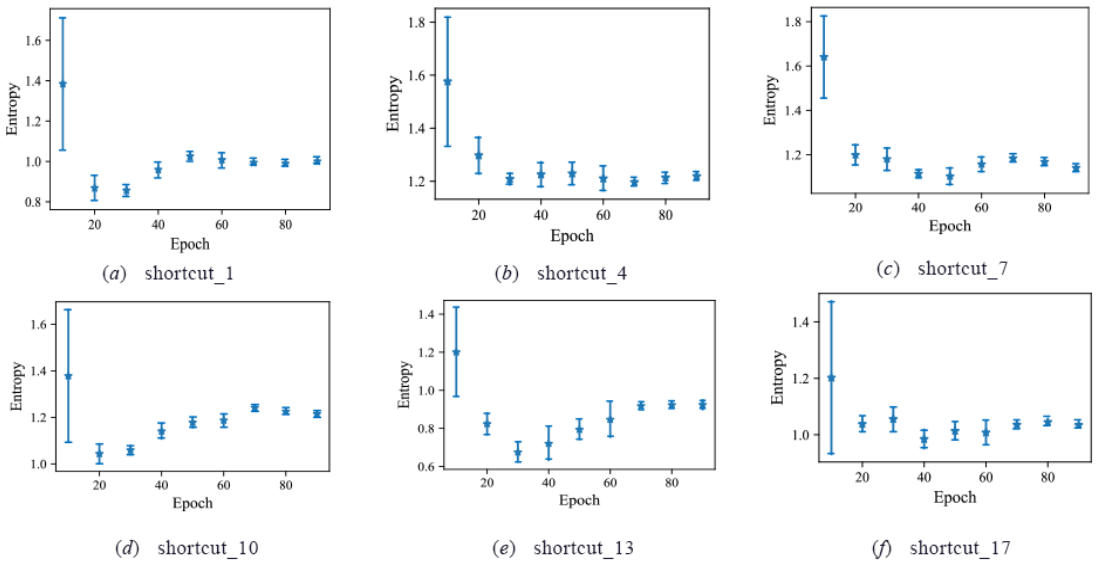

图4　ResNet-56在CIFAR-100上层信息熵变化过程

从图4中可以看出,当网络迭代次数较少时,各层信息熵在迭代过程中变化较大,而随着网络迭代次数的增加,层信息熵标准差逐渐趋于0,模型层信息熵逐渐趋于稳定. 也就是说,网络需经过多轮迭代才能构建稳定的层信息熵. 迭代剪枝方案在迭代更新网络结构时通常只训练1轮[21]或2轮[22],在网络尚未构建好稳定的层信息熵关系的情况下,迭代更新网络结构通常会导致“次优方案迭代次优方案”. 若训练多轮后再利用优化方法进行更新网络结构来避免这个问题,则需要耗费大量的计算资源. 为此,本文考虑层信息熵关系的鲁棒性以及网络构建层信息熵的能力,提出一种基于自适应层信息熵的模型压缩方法,利用层信息熵建立模型剪枝过程中剪枝与重训练的联系,这是一种不需要迭代剪枝的端到端结构化剪枝方案.

3.2 获取压缩模型

为了更好地衡量网络各层之间的信息量大小关系,将网络各层信息熵放在一起进行定量分析,利用层信息熵之间的大小关系确定网络各层过滤器个数的保留率,其过程如图5所示. 首先,根据式(1)得到网络各卷积层的信息熵,然后,分别找到其中最小的信息熵SE和最大的信息熵BE,定义模型原始层信息熵区间OHS=[SE, BE]. 确定各卷积层过滤器个数最大保留率$\alpha_{\max}\in\{10\%,20\%,\cdots,100\%\}$,图5取$\alpha_{\max}=80\%$为例. 为了更好地体现每层权重信息熵与选取的$\alpha_{\max}$之间的关系,对模型原始层信息熵区间OHS进行扩充,定义模型层信息熵区间HS为

$$\mathrm{HS}=\left[\mathrm{SE}-(\mathrm{BE}-\mathrm{SE})/\left(10\times\alpha_{\max}\right),\ \mathrm{BE}+(\mathrm{BE}-\mathrm{SE})/\left(10\times\alpha_{\max}\right)\right] \tag{2}$$

将模型层信息熵区间HS分成$10\times\alpha$等份，每份区间用HS_α表示，则网络各层过滤器保留率表示为

$$\alpha_i=\left\lceil H\left(\boldsymbol{W}_i\right)\in \mathrm{HS}_\alpha\right\rceil \tag{3}$$

其中$\lceil*\rceil$表示返回所取区间上端点处的保留率，$\alpha_i\in\left\{10\%,20\%,\cdots,\alpha_{\max}\right\}$表示第$i$层过滤器个数保留率，则压缩后的网络各层过滤器个数可以表示为$\boldsymbol{C}'=(\alpha_1c_1,\alpha_2c_2,\cdots,\alpha_Lc_L)$. α_i的最小值即为各卷积层过滤器个数的最小保留率，文中默认层最小保留率为10%，关于最小保留率的分析详见4.3.2节.

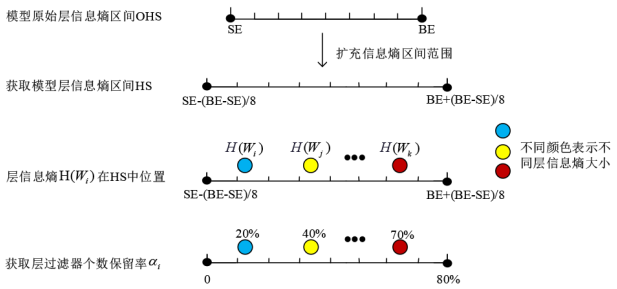

图5 网络各层过滤器保留率获取过程

为了更好地阐释整个压缩过程，以常规连续卷积为例，剪枝过程如图6所示，由基线网络求取各卷积层的信息熵，然后利用式(1)、式(2)、式(3)求取各卷积层过滤器个数的保留率，当某层过滤器个数与层保留率相乘的结果为小数时，采用向上取整的方法得到该层过滤器最终保留的个数. 剪枝后的网络，第i层特征图的个数等于第$(i-1)$层过滤器的个数，第$(i-1)$层每个过滤器的通道数等于第$(i-1)$层特征图的个数. 需要注意的是，剪枝后的网络结构是根据各卷积层过滤器的保留率直接得到，并不是从基线网络中继承某结构或某结构参数.

针对具有短连接的网络，本文采用相同的压缩方式，以ResNet和MobileNetV2为例，两者的短连接块均采用“$1\times1\rightarrow3\times3\rightarrow1\times1$”卷积，同时通过短连接将输出与输入相加的模式，区别在于ResNet采用先降维再升维，属于“沙漏型”结构，MobileNetV2采用先升维再降维，属于“纺锤形”结构. 具有短连接块的最后一层不参与压缩，短连接上的卷积不参与压缩，如图7所示，图中c_i和c_i'分别表示剪枝前后网络中过滤器的个数，$c_i'=c_i\times\alpha_i$；$c_2\times c_1\times1\times1$表示卷积层有$c_2$个过滤器，每个过滤器有$c_1$个卷积核，每个卷积核的尺寸为$1\times1$.

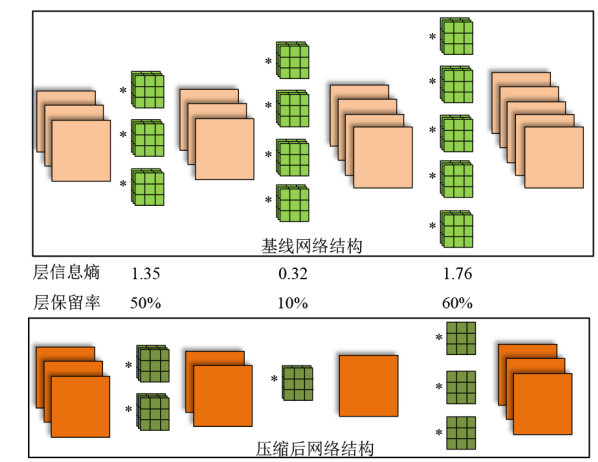

图6 压缩网络结构的过程

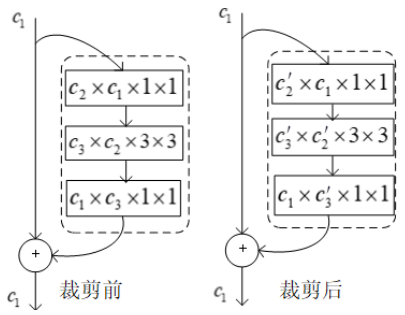

图7 具有短连接块的网络压缩方法

3.3 自适应联合嫁接

模型压缩的本质是压缩网络，提升精度. 在3.2节，利用层信息熵之间的大小关系获取了压缩后的网络结构，本节的重点是提升压缩后的网络精度. Meng等[24]首次提出基于信息熵自适应嫁接卷积层参数的方法提升网络精度，但没有考虑到BN层对网络剪枝的影响. 受文献[27]的启发，本文综合考虑压缩过程的层信息熵指标，通过对卷积层与BN层建立多网络自适应联合嫁接，对剪枝后的网络并行重训练. 利用层信息熵之间的大小关系，每轮更新时，将并行训练的网络相同卷积层与BN层进行自适应联合嫁接，提升压缩后的网络精度.

以剪枝后并行训练2个网络的第一层为例，这2个网络具有相同的结构，迭代相同的轮数，为了增加网络中学习到的信息多样性，可设置初始学习率不一致，学习率衰减方式不一致，采用随机初始化网络参数等方法. 卷积层与BN层的联合嫁接方法如图8所示，每轮训练完成后，将$N_1_l_1$与$N_1_b_1$的参数分别嫁接到网络$N_2_l_1$和$N_2_b_1$中，两者的嫁接过程完全一致，以$N_2_l_1$为例，嫁接过程的权重更新方式为

$$\boldsymbol{W}_1^{N_2}=\beta\boldsymbol{W}_1^{N_2}+(1-\beta)\boldsymbol{W}_1^{N_1},\ 0<\beta<1 \tag{4}$$

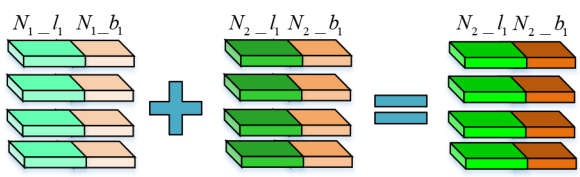

图8 联合嫁接示意图

自适应嫁接系数的大小主要取决于嫁接网络对应层信息熵的大小. 以图8中卷积层为例,当$H\left(\boldsymbol{W}_1^{N_2}\right)=H\left(\boldsymbol{W}_1^{N_1}\right)$时,嫁接后$\boldsymbol{W}_1^{N_2}$应是嫁接前$\boldsymbol{W}_1^{N_2}$与$\boldsymbol{W}_1^{N_1}$的平均值,即$\beta=0.5$. 当$H\left(\boldsymbol{W}_1^{N_2}\right)>H\left(\boldsymbol{W}_1^{N_1}\right)$时,$\beta>0.5$,$H\left(\boldsymbol{W}_1^{N_2}\right)<H\left(\boldsymbol{W}_1^{N_1}\right)$时,$\beta<0.5$,且自适应嫁接系数的大小应具有中心对称的性质. 同样的网络结构即使在同一数据集上训练时,由于网络初始化参数不同,以及采用随机梯度下降优化带来的随机性,网络训练一轮后,对应层信息熵的大小可能会存在较大差异. 但即使在这种情况下,嫁接后的网络也应该包含原网络自身权重信息,因此,β的取值范围应进一步缩小. 据此,自适应嫁接系数β定义为

$$\beta=\mathrm{A}\times\left(\arctan\left(\mathrm{c}\times\left(H\left(\boldsymbol{W}_i^{N_2}\right)-H\left(\boldsymbol{W}_i^{N_1}\right)\right)\right)\right)+0.5 \quad (5)$$

其中,A和c是常数. 图9所示为根据式(5)所设计的β随$H\left(\boldsymbol{W}_i^{N_2}\right)-H\left(\boldsymbol{W}_i^{N_1}\right)$的变化曲线图,可以看出,该$\beta$定义满足上述条件.

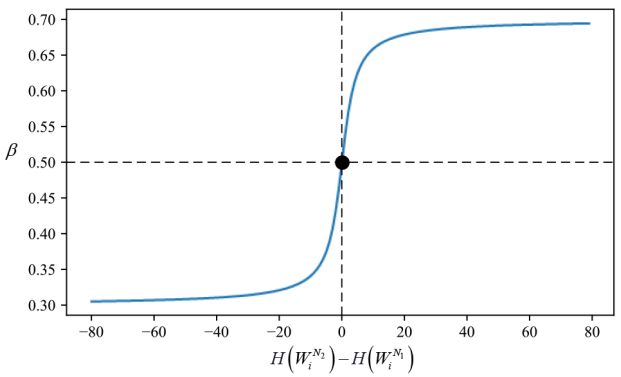

图9　自适应嫁接系数β变化关系图

嫁接后的网络可以从不同的网络权重中获取更多的信息,达到提升网络精度的目的. 需要注意的是,当只并行嫁接2个网络时采用2个网络互相嫁接的策略,使用多网络嫁接时采用顺序成闭环的嫁接方式,如图10所示,箭头方向表示网络嫁接方向.

4　实验分析

为了验证本文方法ALE的有效性,针对当下主流网络VGGNet,ResNet以及轻量级网络MobileNetV2,在3个标准分类数据集CIFAR-10[28],CIFAR-100[28]和SVHN[29]上,以及2个行人重识别数据集Market1501[30]和DukeMTMC-ReID(Duke)[31]上进行了实验. CIFAR-10与CIFAR-100均包含60 000张大小为32×32的三通道彩色图像,其中CIFAR-10包括10个类别,训练集50 000张图片,测试集10 000张图片;CIFAR-100包括100个类别,每个类包含600张图片,每类各有500张训练图片和100张测试图片. SVHN是一个真实的街景门牌号数据集,包括73 257张训练图片和26 032张测试图片,每张图片包含阿拉伯数字"0~9"十个类别. Market1501包括1 501个行人、32 668个检测到的行人矩形框,训练集有751个人、12 936张图像,测试集有750个人、19 732张图像. Duke数据集是一个大规模标记的多目标多摄像头行人跟踪高清视频数据集,具有7 000多个单摄像机轨迹和2 700多个独立人物.

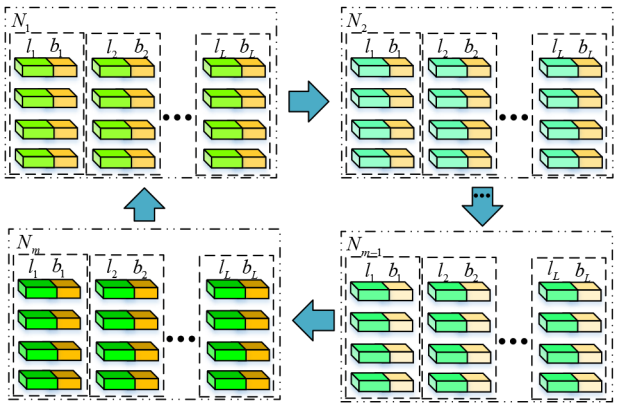

图10　多网络联合嫁接示意图

4.1　实验设置

实验环境为PyTorch-1.2.0框架,Python3.6,2张NVIDIA GTX1080TI 11GB显卡.

具体实验设置信息为得到剪枝后的网络模型,采用顺序成闭环的多网络嫁接方式嫁接6个网络进行训练,随机初始化各个网络权重,采用随机梯度下降优化算法,余弦退火学习率衰减方式,计算自适应嫁接系数β公式中,常数A = 0.4/π,c = 500. 在标准分类数据集上,初始学习率均为0.1,批大小为64,采用交叉熵损失,训练200轮,在行人重识别数据集上,初始学习率均为0.035,批大小为32,采用softmax损失,训练120轮.

本文中提供3个算法评价指标:准确率(Accuracy)用来衡量网络的任务表现;计算量(Floating-point Operations per second,FLOPs),表示网络前向传播时需要进行的加法操作和乘法操作的次数,用来衡量网络前向推理速度;参数量(parameters)用来衡量网络复杂度.

为了更好地验证ALE方法的优越性,本文引入近年来模型压缩领域其他方法进行对比,如PF[16],SSS[32],HRank[33],GAL[21],LEGR[34],NISP[35],Hinge[36],KSE[26],FPGM[19],ABC[22]和C-SGD[4].

4.2　方法比较及分析

在CIFAR-10数据集上,表2~表5分别提供了网络ResNet-56,ResNet-110,VGGNet和MobileNetV2的实验结果. "M"表示百万,ALE-$\alpha_{\max}$中的$\alpha_{\max}$表示网络每层过滤器个数的最大保留率.

由表2可以看到，ALE在计算量压缩率为36.2%的情况下，准确率相比于基线网络提升了1个百分点．相比于GAL-0.6和NISP，ALE方法在压缩率大致相同时，准确率提升了约1个百分点．相比于HRank2和KSE，在计算量压缩率提升了10个百分点的情况下，准确率依然提升了0.4个百分点．PF算法通过实验找出网络模型的“敏感层”，即裁剪该层对网络输出影响较大，然后在裁剪过程中对这些层不进行裁剪，表中PF-A表示“敏感层”为[16，20，38，54]，PF-B表示“敏感层”为[16，18，20，34，38，54]．表2中LEGR1与LEGR2为同一篇论文LEGR中提供的2组不同压缩率下的结果，HRank1与HRank2为同一篇论文HRank中提供的2组不同压缩率下的结果，SSS为复现的结果．

表2 ResNet-**56**在CIFAR-**10**上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	93.26%	125.49M(0.0%)	0.85M(0.0%)
PF-A[16]	93.10%	112.00M(10.7%)	0.77M(9.4%)
PF-B[16]	93.06%	90.90M(27.6%)	0.73M(14.1%)
SSS[32]	93.39%	89.35M(28.8%)	0.59M(30.6%)
HRank1[33]	93.52%	88.72M(29.3%)	0.71M(16.8%)
LEGR1[34]	94.10%	87.80M(30.0%)	
NISP[35]	93.01%	81.00M(35.5%)	0.49M(42.4%)
ALE-100%	94.26%	80.10M(36.2%)	0.54M(32.9%)
GAL-0.6[21]	93.38%	78.30M(37.6%)	0.75M(11.8%)
HRank2[33]	93.17%	62.72M(50.0%)	0.49M(42.4%)
Hinge[36]	93.69%	62.72M(50.0%)	0.44M(51.27%)
KSE[26]	93.23%	60.00M(50.0%)	0.43M(49.4%)
FPGM[19]	93.26%	59.40M(52.6%)	
LEGR2[34]	93.70%	58.90M(53.1%)	
ABC-70%[22]	93.23%	58.54M(54.1%)	0.39M(54.2%)
ALE-60%	93.64%	49.53M(60.5%)	0.36M(57.6%)
C-SGD[4]	93.31%	49.13M(60.85%)	

由表3可以看到，ALE在计算量压缩率为52.4%的情况下，准确率相比于基线网络提升了1.42个百分点，据我们所知，此为目前模型压缩方法中最佳表现．相比于FPGM，在计算量压缩率大致相同的情况下，ALE方法准确率提升了1.18个百分点．相比于HRank，在计算量压缩率提高了5个百分点的情况下，准确率提升了1个百分点．

由表4可以看到，相比于HRank与LEGR，ALE在压缩率和准确率方面均取得了更好的表现．相比于SSS，ALE方法在大幅提高压缩率的同时，准确率仅下降不到0.2个百分点．相比于GAL，当GAL超参数为0.1时(GAL-0. 1)，ALE方法在大幅提高压缩率的同时，准确率下降不到1个百分点；当GAL超参数为0.2时(GAL-0.2)，ALE-80%在参数压缩率提高了6个百分点的情况下，精度依然提高了近1个百分点．

表3 ResNet-**110**在CIFAR-**10**上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-110	93.50%	252.89M(0.0%)	1.72M(0.0%)
GAL-0.1[21]	93.59%	205.7M(18.7%)	1.65M(4.07%)
NISP[35]	93.32%	143.35M(43.3%)	
GAL-0.5[21]	92.74%	130.20M(48.5%)	0.95M(44.8%)
FPGM[19]	93.74%	121.00M(52.2%)	
ALE-80%	94.92%	120.29M(52.4%)	0.89M(48.3%)
HRank[33]	93.36%	105.70M(58.2%)	0.70M(59.3%)
ALE-60%	94.36%	92.68M(63.4%)	0.69M(59.9%)
ABC-60%[22]	93.58%	89.87M(64.5%)	0.56M(67.4%)

表4 VGGNet在CIFAR-**10**上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
VGGNet	93.96%	313.74M(0.0%)	14.95M(0.0%)
SSS[32]	93.02%	183.13M(42.3%)	3.93M(73.7%)
GAL-0.1[21]	93.42%	171.89M(45.2%)	2.67M(82.2%)
HRank1[33]	92.34%	108.61M(65.4%)	2.64M(82.3%)
HRank2[33]	91.23%	73.70M(76.5%)	1.78M(88.2%)
ALE-90%	92.85%	71.29M(77.28%)	2.87M(80.8%)
LEGR[34]	92.40%	70.30M(77.6%)	---
GAL-0.2[21]	91.89%	65.85M(79.0%)	3.35M(77.6%)
ALE-80%	92.81%	60.23M(80.88%)	2.46M(83.6%)

为了进一步证明方法的鲁棒性，在轻量型网络MobileNetV2上进行了验证，如表5所示，可以发现，计算量压缩55.2%时，模型准确率反而提升了1.29个百分点，计算量压缩率超过64%时，该方法依旧可以获得比基线网络更好的准确率．轻量型网络中，ALE方法依旧可以获得良好的效果．

表5 MobileNetV**2**在CIFAR-**10**上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
MobileNetV2	92.25%	91.15M(0.0%)	2.26M(0.0%)
ALE-90%	93.54%	40.86M(55.2%)	0.94M(58.4%)
ALE-70%	92.79%	32.66M(64.2%)	0.83M(63.3%)

在CIFAR-100数据集上，网络ResNet-56的实验结果如表6所示，相比于PF-A与PF-B，ALE在计算量与参数量的压缩率均更高的情况下，准确率更高．

在SVHN数据集上，网络ResNet-56与ResNet-110的实验结果如表7所示，为了更好地比较，对GAL方法

表6 ResNet-**56**在CIFAR-**100**上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	71.92%	125.49M(0.0%)	0.85M(0.0%)
PF-A[16]	70.42%	112.44M(10.4%)	0.77M(9.4%)
PF-B[16]	69.95%	90.85M(27.6%)	0.73M(13.7%)
ALE-90%	70.91%	64.77M(48.4%)	0.53M(37.7%)
ALE-80%	70.72%	57.97M(53.8%)	0.47M(44.7%)

在SVHN上进行了实验,可以看到,ResNet-56在计算量与参数量压缩率均约为60%时,准确率相比于基线网络提升了0.14%. 相比于GAL,ALE方法在计算量与参数量压缩率对比为41.4% vs. 60.0%, 34.1% vs. 58.8%情况下,准确率高了0.7%. ResNet-110在准确率一致的情况下,ALE方法的计算量与参数量的压缩率均更高.

表7　ResNet-56和ResNet-110在SVHN上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	96.38%	125.49M(0.0%)	0.85M(0.0%)
GAL-0.6[21]	95.82%	73.58M(41.4%)	0.56M(34.1%)
ALE-60%	96.52%	50.17M(60.0%)	0.35M(58.8%)
ResNet-110	96.36%	252.89M(0.0%)	1.72M(0.0%)
GAL-0.06[21]	96.63%	120.77M(52.2%)	0.83M(51.7%)
ALE-60%	96.65%	98.50M(61.1%)	0.71M(58.7%)

为了更好地验证ALE方法的有效性和可扩展性,本文增加了在行人重识别数据集Market1501和Duke上的实验,网络ResNet-50的实验结果如表8和表9所示,可以看到,在2个数据集上,计算量与参数量在均压缩超过一半时,mAP指标与Rank1指标均只是稍有下降,验证了ALE方法不仅适用于分类任务,同样适用于其他任务.

表8　ResNet-50在Market1501上结果

模型	mAP	Rank1	计算量(压缩率)	参数量(压缩率)
ResNet-50	65.1%	84.1%	4087.14M (0.0%)	24.99M (0.0%)
ALE-90%	62.2%	82.1%	1577.95M (61.4%)	8.28M (66.9%)
ALE-80%	61.5%	81.7%	1419.49M (65.3%)	7.48M (70.1%)

表9　ResNet-50在Duke上结果

模型	mAP	Rank1	计算量(压缩率)	参数量(压缩率)
ResNet-50	52.3%	72.1%	4087.14M (0.0%)	24.89M (0.0%)
ALE-90%	51.7%	71.1%	1639.11M (59.9%)	7.41M (70.2%)
ALE-80%	51.0%	70.5%	1508.50M (63.1%)	7.22M (71.0%)

4.3　消融实验

4.3.1　ALE参数分析

ALE中通过超参数α_{max}调整网络压缩率,选取ResNet-56在CIFAR-10上的实验结果来分析最大保留率α_{max}对网络性能的影响. 为了更好地说明α_{max}的影响,针对同一预训练模型,相同压缩方法(除α_{max}选取不同,其他均一致)进行实验. 结果如图11所示,不难发现,随着最大保留率α_{max}的增加,模型准确率逐渐上升,计算量与参数量的压缩率逐渐降低,网络在不同压缩率情况下,任务精度均取得了良好的表现,说明ALE方法具有良好的鲁棒性. 本文中,均综合考虑模型准确率与计算量的压缩率来选取结果,例如,选取ALE-60%作为实验结果呈现在表2中.

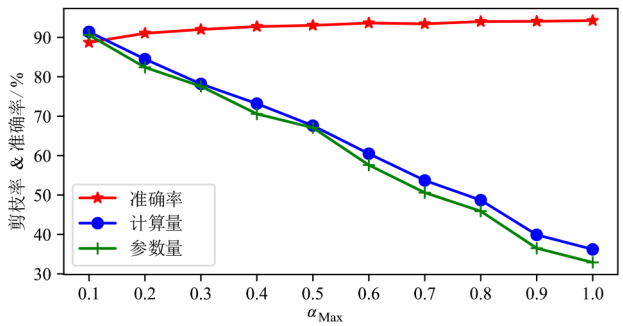

图11　ResNet-56在CIFAR-10上最大保留率α_{max}对网络性能的影响

相同层网络信息熵越大,说明该层学习到的信息越丰富. 为了更好地验证网络压缩前后层信息熵大小关系的变化,选取MobileNetV2在CIFAR-10上最大保留率为90%的剪枝重训练模型,计算被压缩的卷积层信息熵,并与压缩前的基线模型进行对比,如图12所示,黄色折线表示剪枝后的模型,蓝色线表示基线模型,可以发现,相比于剪枝前的网络,剪枝后的网络层信息熵普遍增加,说明网络确实裁减了冗余的过滤器.

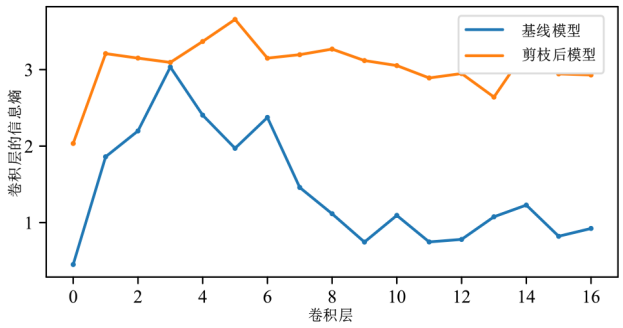

图12　MobileNetV2在CIFAR-10上压缩前后层信息熵关系比较

本节针对不同网络在不同数据集上进行了实验,对比了剪枝前后网络信息熵的变化情况,验证了ALE方法的有效性,通过对最大保留率α_{max}的分析,验证了ALE方法的鲁棒性.

4.3.2　ALE各模块必要性分析

本文提出了一种基于层信息熵的自适应联合嫁接方法,为了证明此方法相比于剪枝后直接训练以及直接嫁接卷积层的优越性,在数据集CIFAR-10与SVHN上针对不同网络进行了验证,实验设置除嫁接部分不

同外，其余均一致．实验结果如图13所示，其中，纵坐标表示错误率，横坐标“VGG-C10”表示在CIFAR-10上VGGNet的实验结果．可以发现，在被压缩的网络结构一致的情况下，采用联合嫁接的方式在所有任务上均获得了最好的准确率表现．

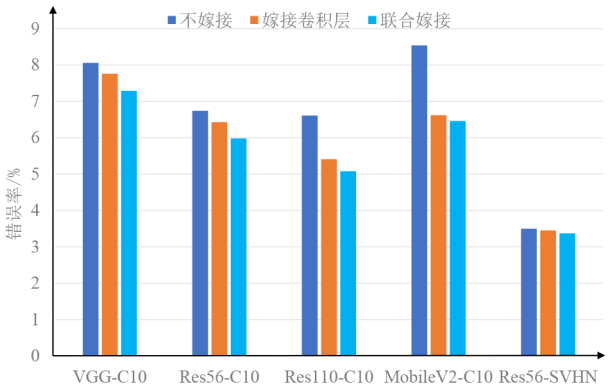

图13 嫁接对比实验结果

本文在剪枝前需要加载网络预训练权重，为了更好地说明不同轮数的预训练权重对ALE剪枝方法的影响，在信息熵变化较大的阶段，每隔5个epoch选取一次权重，针对信息熵变化较小的阶段，每隔30个epoch选取一次权重，以ResNet-56在CIFAR-100上为例，选取第5，10，15，20，30，60，90，120，150轮的权重进行实验，采用的最大保留率均为80%，其中采用第120轮权重与采用精度最高的预训练轮数权重（122轮）剪枝后的网络结构完全一致，实验结果如表10所示，表中ALE-90%-122表示最大保留率为90%，采用第122轮预训练权重．从表10中可以看出，采用前30轮预训练权重，ALE方法表现相对较差，30轮之后的预训练权重，ALE方法均能有较好的表现，甚至使用第90轮的预训练权重能获得相对于选取最高精度的预训练权重剪枝后表现更好．从总体效果以及实用性角度，本文使用统一的选取指标，即以最高精度的预训练轮数的权重进行剪枝．

ALE算法默认的层最小保留率为10%，本文中所有的结果均是建立在此基础上的．为了更好地验证最小保留率为10%的合理性，本文对不同的网络在CIFAR-10上进行了实验，实验结果如表11所示，表中20%-ALE-60%表示最小保留率为20%，最大保留率为60%，在其他实验参数设置均一致的情况下，发现随着层最小保留率的增大，网络剪枝量变小，但网络精度并没有明显提升，综合考虑网络精度与剪枝量2个指标，本文不将层最小保留率作为超参数．

为了更好地说明自适应联合嫁接的优越性，本文在数据集CIFAR-10上对网络ResNet-110进行了不同嫁接网络个数的实验，实验结果如表12所示．由实验发现，自适应联合嫁接方法有着极大的优越性，增加网络嫁接的个数，准确率基本在逐渐增加，增加幅度在变缓，实际应用时可以综合考虑硬件资源以及准确率需求来选择网络嫁接个数．

表10 ResNet-56在CIFAR-100上结果

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	71.92%	125.49M(0.0%)	0.85M(0.0%)
ALE-80%-10	69.31%	69.93M(44.3%)	0.52M(38.8%)
ALE-80%-15	70.06%	68.31M(45.6%)	0.53M(37.7%)
ALE-90%-122	70.91%	64.77M(48.4%)	0.53M(37.7%)
ALE-80%-60	70.48%	63.74M(49.2%)	0.48M(43.5%)
ALE-80%-20	70.26%	61.97M(50.6%)	0.50M(41.2%)
ALE-80%-90	70.93%	61.67M(50.9%)	0.49M(42.4%)
ALE-80%-30	70.19%	61.50M(51.0%)	0.47M(44.7%)
ALE-80%-122	70.72%	57.97M(53.8%)	0.47M(44.7%)
ALE-80%-150	70.65%	55.35M(55.9%)	0.44M(48.2%)
ALE-80%-5	69.17%	50.32M(59.9%)	0.36M(57.6%)

表11 层最小保留率的影响

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	93.26%	125.49M(0.0%)	0.85M(0.0%)
ALE-60%	93.64%	49.53M(60.5%)	0.36M(57.6%)
20%-ALE-60%	93.50%	52.80M(57.9%)	0.37M(56.5%)
ResNet-110	93.50%	252.89M(0.0%)	1.72M(0.0%)
ALE-80%	94.92%	120.29M(52.4%)	0.89M(48.3%)
20%-ALE-80%	94.55%	131.50M(48.0%)	0.96M(44.2%)
MobileNetV2	92.25%	91.15M(0.0%)	2.26M(0.0%)
ALE-90%	93.54%	40.86M(55.2%)	0.94M(58.4%)
20%- ALE-90%	94.11%	46.20M(49.3%)	1.08M(52.2%)

表12 嫁接网络个数的影响

模型	嫁接个数	准确率
ResNet-110		93.50%
ALE-80%	1	93.39%
	2	93.86%
	3	94.55%
	4	94.21%
	5	94.71%
	6	94.92%

本文3.2节在进行网络剪枝时对信息熵区间进行了扩充，为验证扩充信息熵区间的必要性，对是否对其进行扩充在数据集CIFAR-10上进行了实验对比，其结果如表13所示．从表中可以看到，无论是否进行扩充，本文方法均能完成剪枝，且取得很好的效果，但通过对信息熵进行扩充，能在保持压缩率的同时，得到更高的识别准确率．

表 13　是否扩充信息熵实验对比

模型	准确率	计算量(压缩率)	参数量(压缩率)
ResNet-56	93.26%	125.49M(0.0%)	0.85M(0.0%)
ALE-100%(扩)	94.26%	80.10M(36.2%)	0.54M(32.9%)
ALE-100%	94.11%	81.47M(35.1%)	0.59M(30.6%)
ALE-60%(扩)	93.64%	49.53M(60.5%)	0.36M(57.6%)
ALE-60%	93.37%	49.38M(60.7%)	0.36M(57.6%)
ResNet-110	93.50%	252.89M(0.0%)	1.72M(0.0%)
ALE-80%(扩)	94.92%	120.29M(52.4%)	0.89M(48.3%)
ALE-80%	94.81%	120.88M(52.2%)	0.91M(47.1%)
MobileNetV2	92.25%	91.15M(0.0%)	2.26M(0.0%)
ALE-90%(扩)	93.54%	40.86M(55.2%)	0.94M(58.4%)
ALE-90%	93.42%	38.90M(57.3%)	0.89M(60.6%)

5　总结

本文提出一种基于自适应层信息熵的模型压缩方法,利用层信息熵之间的关联性确定各层过滤器的保留率,得到被压缩的网络结构后,通过基于层信息熵的自适应联合嫁接方法来恢复网络精度. 本方法不仅在常见的主流网络 VGGNet 和 ResNet 中取得良好的表现,在轻量型网络 MobileNetV2 中也取得了很好的效果,此外,此方法具有可扩展性,实验证明,ALE 方法不仅在分类领域有效,在行人重识别领域同样获得了良好的表现. 未来将结合迁移学习,综合考虑任务准确率和压缩率指标,努力实现在不同数据集上剪枝出一种具有相同网络结构的模型,从而减少模型压缩过程中需要耗费的计算资源.

参考文献

[1] ZHU F, ZHU L, YANG Y. Sim-real joint reinforcement transfer for 3d indoor navigation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California: IEEE, 2019: 11388-11397.

[2] 权宇, 李志欣, 张灿龙, 等. 融合深度扩张网络和轻量化网络的目标检测模型[J]. 电子学报, 2020, 48(2): 390-397.
QUAN Y, LI Z X, ZHANG C L, et al. Fusing deep dilated convolutions network and light-weight network for object detection[J]. Acta Electronica Sinica, 2020, 48(2): 390-397. (in Chinese)

[3] 周涛, 霍兵强, 陆惠玲, 等. 残差神经网络及其在医学图像处理中的应用研究[J]. 电子学报, 2020, 48(7): 1436-1447.
ZHOU T, HUO B Q, LU H L, et al. Research on residual neural network and its application on medical image processing[J]. Acta Electronica Sinica, 2020, 48(7): 1436-1447. (in Chinese)

[4] DING X, DING G, GUO Y, et al. Centripetal sgd for pruning very deep convolutional networks with complicated structure[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California: IEEE, 2019: 4943-4953.

[5] 饶川, 陈靓影, 徐如意, 等. 一种基于动态量化编码的深度神经网络压缩方法[J]. 自动化学报, 2019, 45(10):1960-1968.
RAO C, CHEN J Y, XU R Y, et al. A dynamic quantization coding based deep neural network compression method[J]. Acta Automatica Sinica, 2019, 45(10): 1960-1968. (in Chinese)

[6] ADRIANA R, NICOLAS B, SAMIRA E K, et al. Fitnets: Hints for thin deep nets[C]//International Conference on Learning Representations. California: OpenReview.net, 2015: 1-13.

[7] 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, 2015, 38(10): 1943-1955.

[8] WANG Y, XU C, XU C, et al. Beyond filters: Compact feature map for portable deep model[C]//International Conference on Machine Learning. Sydney: ACM, 2017: 3703-3711.

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

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

[11] CARREIRA-PERPINAN M A, IDELBAYEV Y. Learning-compression algorithms for neural net pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 8532-8541.

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

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

[14] LUO J H, WU J, LIN W. Thinet: A filter level pruning method for deep neural network compression[C]//Proceedings of the IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 5058-5066.

[15] WANG D, ZHOU L, ZHANG X, et al. Exploring Linear Relationship in Feature Map Subspace for Convnets Compression[EB/OL]. (2018-03-15) [2020-12-01]. https://arxiv.org/abs/1803.05729.

[16] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters

for efficient convnets[C]//International Conference on Learning Representations. Toulon: OpenReview.net, 2017: 1-13.
[17] YE J, LU X, LIN Z, et al. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers[C]//International Conference on Learning Representations. Vancouver: OpenReview.net, 2018: 1-11.
[18] ZHUO H, QIAN X, FU Y, et al. Scsp: Spectral Clustering Filter Pruning with Soft Self-Adaption Manners [EB/OL]. (2018-07-14) [2020-12-01]. https://arxiv. org/abs/1806.05320.
[19] HE Y, LIU P, WANG Z, et al. Filter pruning via geometric median for deep convolutional neural networks acceleration[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California: IEEE, 2019: 4340-4349.
[20] LIU Z, SUN M, ZHOU T, et al. Rethinking the value of network pruning[C]//International Conference on Learning Representations.NewOrleans:OpenReview.net,2019:1-21.
[21] LIN S, JI R, YAN C, et al. Towards optimal structured cnn pruning via generative adversarial learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California: IEEE, 2019: 2790-2799.
[22] LIN M, JI R, ZHANG Y, et al. Channel pruning via automatic structure search[C]//International Joint Conference on Artificial Intelligence. Yokohama: Morgan Kaufmann, 2020: 1-7.
[23] LIU Z, ZHANG X, SHEN Z, et al. Joint Multi-Dimension Pruning[EB/OL]. (2020-05-18)[2020-12-01]. https://arxiv.org/abs/2005.08931.
[24] MENG F, CHENG H, LI K, et al. Filter grafting for deep neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 6599-6607.
[25] LUO J H, WU J. An Entropy-Based Pruning Method for CNN Compression[EB/OL]. (2017-07-19) [2020-12-01]. https://arxiv.org/abs/1706.05791.
[26] LI Y, LIN S, ZHANG B, et al. Exploiting kernel sparsity and entropy for interpretable CNN compression[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. California: IEEE, 2019: 2800-2809.
[27] WANG L, XIONG Y, WANG Z, et al. Temporal segment networks: Towards good practices for deep action recognition[C]//Proceedings of the European Conference on Computer Vision, Amsterdam. Holland: Springer, 2016: 20-36.
[28] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[J]. Technical Report, 2009, 1 (1): 1-60.
[29] NETZER Y, WANG T, COATES A, et al. Reading digits in natural images with unsupervised feature learning[J]. NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011, 1(1): 1-9.
[30] ZHENG L, SHEN L, TIAN L, et al. Scalable person re-identification: A benchmark[C]//Proceedings of the IEEE International Conference on Computer Vision. Santiago: IEEE, 2015: 1116-1124.
[31] RISTANI E, SOLERA F, ZOU R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]//Proceedings of the European Conference on Computer Vision. Amsterdam: Springer, 2016: 17-35.
[32] HUANG Z, WANG N. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European Conference on Computer Vision. Munich: Springer, 2018: 304-320.
[33] LIN M, JI R, WANG Y, et al. HRank: Filter pruning using high-rank feature map[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 1529-1538.
[34] CHIN T W, DING R, ZHANG C, et al. Towards efficient model compression via learned global ranking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 1518-1528.
[35] YU R, LI A, CHEN C F, et al. Nisp: Pruning networks using neuron importance score propagation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 9194-9203.
[36] LI Y, GU S, MAYER C, et al. Group sparsity: The hinge between filter pruning and decomposition for network compression[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 8018-8027.

作者简介

魏钰轩 男,1997 年 10 月出生,江苏徐州人. 江南大学研究生. 主要研究方向为图像处理、模型压缩.

陈 莹(通讯作者) 女,1976 年 12 月出生,浙江丽水人. 江南大学教授,博士生导师. 主要研究方向为图像处理、信息融合、模式识别.
E-mail: chenying@jiangnan.edu.cn