楚鹏飞, 魏 巍. 基于注意力机制的通道剪枝方法研究[J]. 智能计算机与应用,2025,15(7):21-28. DOI:10.20169/j.issn.2095-2163.250703

基于注意力机制的通道剪枝方法研究

楚鹏飞[1], 魏 巍[2]
(1 贵州大学 电气工程学院, 贵阳 550025; 2 中国电建集团 贵州电力设计研究院有限公司, 贵阳 550025)

摘 要: 针对现有的剪枝算法出现剪枝率难以确定,对于不同任务和架构的适应性差等问题,提出了一种基于注意力机制的通道剪枝方法。算法融合了 Transformer 中的自注意力机制与 SENet 中的通道注意力机制,利用其激活后的注意力权重作为裁剪通道的重要程度指标,使网络能够有效的聚焦于关键的信息;通过设置相对应的阈值来对网络中冗余的通道数进行区分。所提算法在 CIFAR-10 数据集上对比原 VGG-16 网络中参数量下降 12.57(M),浮点运算数量降低 56%,与最新的通道剪枝方法相比,表现出了优越的性能。网络的浮点运算数量可以显著降低,在最小参数量的情况下,达到了最高的准确率。

关键词: 模型压缩; 通道剪枝; 注意力机制; 阈值
中图分类号: TP183 **文献标志码**: A **文章编号**: 2095-2163(2025)07-0021-08

Research on channel pruning method based on attention mechanism

CHU Pengfei[1], WEI Wei[2]
(1 School of Electrical Engineering, Guizhou University, Guiyang 550025, China;
2 China Power Construction Group, Guizhou Electric Power Design and Research Institute Co., Ltd, Guiyang 550025, China)

Abstract: On the basis of the existing pruning algorithms appearing the pruning rate is difficult to determine, poor adaptability for different tasks and architectures and other problems, a channel pruning method based on the attention mechanism is proposed. The algorithm integrates the self-attention mechanism in transformer and the channel attention mechanism in SENet, and utilizes the attention weight after its activation as an indicator of the importance of pruning channels, so that the network can effectively focus on the key information. Then a corresponding threshold is set to differentiate the number of redundant channels in the network. The proposed algorithm decreases the number of parameters by 12.57(M) and the number of FLOPs (floating point operations) by 56% on CIFAR-10 for the original VGG-16 network, which shows superior performance compared to the latest channel pruning methods, and the number of FLOPs in the network can be significantly reduced to achieve the highest accuracy with the minimum number of parameters.

Key words: model compression; channel pruning; attention mechanism; thresholds

0 引 言

深度神经网络(DNN)的高效架构设计,在许多监督学习任务中表现出卓越的性能,其中包括计算机视觉[1]、语音识别[2]、自然语言处理[3]等等。在过去的几年里,总的趋势是 DNN 越来越深、越来越宽,形成了大量的最终参数。但其代价是由于参数量与计算量大幅度增加,所需的配置要求越来越高,使其很难集成到一些移动应用程序中,例如智能手机或可穿戴设备。所以,在资源有限的一些移动设备上,部署 DNN 一直是一个挑战。

近几年,多种深度神经网络压缩技术被提出,其中主要包括低秩分解、模型量化、知识蒸馏、网络结构搜索(NAS)和网络剪枝等。低秩分解是把每一层的权重矩阵都被低秩的矩阵所取代;模型量化通过减少权重矩阵表示所需的比特数,来减少深度神经网络的内存和计算量;知识蒸馏将一个名为 Teacher 大型 DNN 模型的泛化能力转化给一个名为 Student

基金项目: 中国电建集团贵州电力设计研究院有限公司科技项目(GZEDKJ-2023-02)。
作者简介: 楚鹏飞(1997—),男,硕士,主要研究方向:深度学习,计算机视觉。
通信作者: 魏 巍(1984—),男,博士研究生,高级工程师,硕士生导师,主要研究方向:能源及电力系统规划,嵌入式技术,现场总线。Email:264885860@qq.com。
收稿日期: 2023-08-31

的紧凑模型;NAS 主要利用神经网络中的模型参数和超参数等信息,自动搜索最优的网络结构,获得更高的性能表现,但由于搜索空间巨大,其比网络剪枝方法需要更多的计算预算。网络剪枝方法主要是去除深度神经网络(DNN)组件,如权重参数和滤波器,对 DNN 的整体性能几乎没有影响。

在以上的模型压缩方法中,网络剪枝因其出色的压缩性能在之后的研究工作中得到广泛的探索。网络剪枝方法可以分为多种分类,如果从剪枝的粒度上看,其主要分为两类,即结构化剪枝[4]和非结构化剪枝[5]。结构化剪枝是指在网络的特定层或结构上进行剪枝,例如对整个滤波器、通道或层进行剪枝。这样的剪枝不会改变网络的整体结构,只是减少了某些部分的参数。非结构化剪枝是指对网络的单个参数进行剪枝,而不考虑层或结构。这种剪枝方法可以在粒度更细的级别上减少参数,但可能导致部分神经元被完全剪枝,对模型性能和稳定性造成较大的影响,剪枝后的模型通常需要进行更多的微调才能恢复性能,增加了训练的复杂性。相比之下,只改变网络中滤波器和特征通道数量的结构化剪枝更适合被广泛地使用。

随着研究的深入,包含输入数据和权重信息的特征图将成为当前剪枝技术的主要基础,指导剪枝过程以实现更好的性能。Peng 等[6]利用 Hessian 矩阵实现了近似通道剪枝。Lin[7]发现秩高的特征映射所对应的权值包含更重要的信息,因此在剪枝过程中需要保留。Molchanov 等[8]提出了一种基于泰勒展开的方法,计算剪枝前后网络损失函数的变化,然后对引起较小变化的滤波器进行剪枝。上述方法的有效性很大程度上依赖于人工设计的剪枝标准,但人工的参与很容易使其陷入次优解决方案。而本文滤波器权重阈值设定是基于注意力模块激活值,能从全局角度协同过滤相似矩阵中学习冗余信息,优化剪枝过程。为了获得更好的剪枝效果,也有很多基于数据驱动工作。如,Luo[9]将滤波器剪枝建模为一个优化问题,并表明应该根据从下一层而不是当前层计算的统计数据来修剪滤波器,然后通过贪婪算法来解决优化问题。赵丽君等[10]先对网络进行正则化,然后分别采用权重折半剪枝算法,组合剪枝算法来对网络进行剪枝,取得了不错的效果。Tang 等[11]提出了一种新的范式,通过将所有输入实例的多种信息嵌入到修剪网络的空间中,动态地去除冗余滤波器。随着开发人员越来越关注深度神经网络在资源受限端设备上的推理效率,许多工作开始探索效率或资源约束来指导模型修剪。Ning 等[12]提出了一种有效的端到端预算剪枝流 DSA,其可以在连续空间中分配稀疏性,并通过基于梯度的优化找到分层剪枝比。

尽管现有的通道剪枝方法可以有效地压缩模型,但也存在一些不足之出。首先是剪枝率确定困难,过高的剪枝率可能导致信息损失和性能下降,而过低的剪枝率则无法实现有效的模型压缩;其次是不同层剪枝率不一致,不同卷积层的重要性和冗余程度可能不同,因此采用相同的剪枝率对所有层进行剪枝可能会导致一些层被过度剪枝,而其他层则被保留过多。在此基础上,本文提出了一种基于注意力机制的通道剪枝算法。在通道剪枝即减少输入通道的设计中,融合了 Transformer 的自注意力机制与 SENet 模块的通道注意力。通过融合注意力模块,把激活值作为通道的重要程度参数,裁剪掉卷积层中权重占比较小的通道,从而实现网络自适应剪枝方法。最终的实验结果表明,该方法能够有效地删除网络中各层冗余通道,从而在保持准确性的同时,显著减少了参数量和浮点运算次数。

1 通道剪枝

1.1 通道剪枝的定义

如图 1 所示,假设第 L 层的输入为 Z_l,通道数、高、宽分别记为 n_l、h_l、w_l,输出为 Z_{l+1},则每个滤波器 $A \in \mathbb{R}^{n_l \times k \times k}$ 和输入的特征图计算,得到输出的一个通道。通道剪枝可以理解为对于每个 $n_l \times k_l \times k_l$ 的滤波器剪枝得到 $n_l' \times k_l \times k_l$ 的滤波器 $n_l' \leqslant n_l$。这种剪枝实际上是减掉了之前输入层的一些通道。通道剪枝和滤波器剪枝的主要区别在于通道剪枝的输出通道数不变,而滤波器剪枝输出通道数减少。此外,通道剪枝还会让该层的所有滤波器也需要剪掉相应的通道,这样进行卷积的时候才能进行相应的匹配。通道剪枝不仅会影响本层通道数,还会对上一层产生影响,由于进行卷积的时候,每一组卷积核对应这一层的一个输出通道,也就是下一层的输入通道,自然就需要上一层中也需要剪掉相应的卷积核,这样会大量减少模型中冗余的参数。

1.2 自注意力机制

2017 年 Vaswani 等[13]提出了 Transformer 模型,其中包含了自注意力机制。此模型一经提出便广泛应用于自然语言处理任务,如机器翻译、文本生成和问答系统等。自注意力机制的核心思想是根据输入

序列不同位置之间的相对关系，为每个位置分配权重，以表示其与其他位置的关联程度。相比 RNN 在处理长序列时容易产生梯度消失或梯度爆炸的问题，Transformer 在处理长序列数据时可以保持更好的正确性和有效性，因此被成功地应用于图像处理领域。在图像处理中，自注意力机制被用来捕捉图像中不同位置之间的关联，从而实现更有效的特征提取和图像理解。

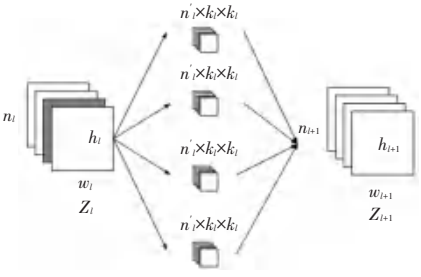

图 1　通道剪枝

Fig. 1　Channel Pruning

1.3　通道注意力机制

通道注意力机制主要通过学习不同通道之间的关联性，来自适应性地调整通道的权重，从而增强有用特征的表示，减少冗余信息的干扰。其主要思想是不同通道在处理特定任务时，可能有不同的重要性，因此应该自动地确定每个通道的权重，以使网络更加关注有用的特征。

通道注意力机制在许多视觉任务中表现出色，如图像分类、目标检测、分割等。其可以帮助网络更好地理解不同通道之间的关联，提升特征表示的质量，从而在各种任务中取得更好的性能。2018 年，Hu 等[14] 提出 SENet，其可以有针对性地提取输入数据中最具有代表性的特征，并且动态地调整不同通道之间的信息关联，从而取得更好的分类性能。

1.4　改进的通道剪枝算法

针对现存大多数通道剪枝算法仅考虑固定剪枝率对网络中各层进行剪枝，而没有考虑不同层的重要性不同的问题，本文在通道剪枝上融合了 Transforme 中的自注意力机制，以及 SENet 中的通道注意力机制，对通道的重要性进行全局比较，并改善模型的泛化能力，提高裁剪精度。

如图 2 所示，假设在第 L 层要剪枝的特征图为 Z_l，输入通道数、高、宽分别为 n_l、h_l、w_l，Z_l 经过全局最大池化（GMP）得到 K，经过两次全局平均池化（GAP）得到 Q，V，其大小都为 $1\times1\times n_l$，其分别表示为：

$$K(x)=\max x_l(i,j) \tag{1}$$

$$Q(x)=V(x)=\frac{1}{w_l\times h_l}\sum_{i=1}^{w_l}\sum_{j=1}^{h_l}x_l(i,j) \tag{2}$$

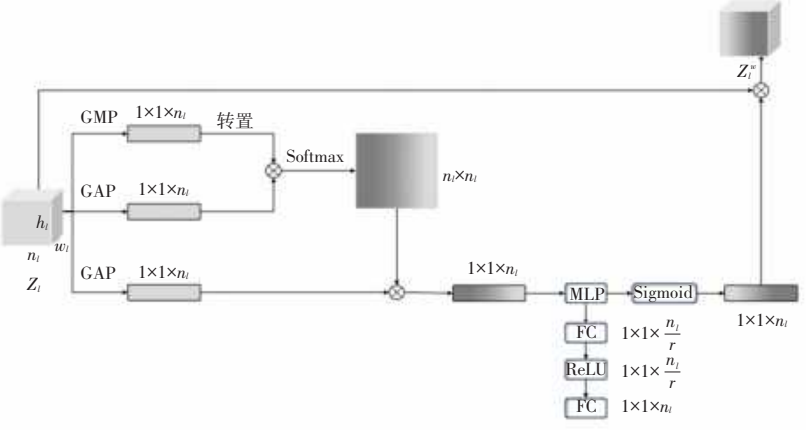

图 2　改进的通道剪枝算法

Fig. 2　Improved channel pruning algorithm

对 $K(x)$ 和 $Q(x)$ 进行 Softmax 函数计算，得到空间注意力权重图 $f(x_i,x_j)$。此举在于融合 GMP 和 GAP 的操作过程，加强特征融合。

$$f(x_i,x_j)=\frac{\exp(l_{ij})}{\sum_{i=1}^{N}\exp(l_{ij})} \tag{3}$$

式中：$l_{ij}=K(x_i)^{\mathrm{T}}Q(x_j)$，$f(x_i,x_j)$ 在图像中表示在通过 GAP 操作下的第 j 个像素点时，网络对 GMP 操作下的第 i 个像素点的注意程度。然后再与 $V(x)$ 相乘得到 y_i：

$$y_i=f(x_i,x_j)V(x_j) \tag{4}$$

y_i 为带有 GMP 与 GAP 融合特征的通道注意力权重图，随后将 y_i 放进 SENET 模块来融合通道注意力机制。首先通过一个 MLP（多层感知机），MLP 由

一个降维比为 r 的降维层、一个 ReLU 激活函数、一个增维比为 r 增维层组成,让其特征向量大小由 $1\times1\times n_l/r$ 转化为 $1\times1\times n_l$,然后通过 Sigmoid 激活函数将输出值映射到 0~1 之间,得到每个通道的注意力权重:

$$w_l = \sigma[w_2\delta(w_1 y_i)] \tag{5}$$

其中,σ 代表 Sigmoid 激活函数;δ 代表 ReLU 激活函数;w_1 和 w_2 则代表全连接层中对应的权重值;w_l 代表第 L 层通过融合注意力模块后的滤波器激活值。最后将带有注意力机制的权重 w_l 与原输入 Z_l 相乘:

$$Z_l^w = w_l Z_l \tag{6}$$

得到一个带有注意力权重的特征图 Z_l^w。权重越大就表示其对应通道对整个网络模型的贡献程度越大,应该保留。然后,利用注意力模块的权重来评价特征图中相应通道的显著性,本文设置阈值的方式为求得各权重的平均值,设置阈值 T:

$$T = \frac{\theta}{n_l}\sum_{i=1}^{n_l} F_l^{(i)}(\cdot) \tag{7}$$

其中,$F_l(\cdot)$ 表示注意力模块的激活输出;i 代表 L 层中第 i 个通道;θ 表示裁剪的比例参数。

实验中,剪枝通道的数量可由 θ 值的设定进行控制。还可针对不同卷积层剪枝敏感程度设定不同阈值达到不同剪枝比例,降低剪枝后模型的精度损失。注意力权重低于设定阈值的通道和所有相关的参数都被一起删除,以提高模型速度和降低储存消耗。

1.5 不同卷积神经网络的剪枝策略

不同类型的卷积神经网络在剪枝过程中需要采用不同的策略,以保持其结构的稳定性。本文分别对 VGG 网络与 ResNet 网络进行了剪枝方法设计。如图 3 所示,针对 VGG 这种普通逐层卷积神经网络,选择直接对卷积层进行剪枝,但需保证每层裁剪后的通道数与上下层相应的输出和输入保持一致。ResNet 网络分为两种类型,一种是基础残差块,另一种则是瓶颈残差块。基础残差块的网络,剪枝方法和裁剪 VGG 类似,裁剪中间层的同时保持上下通道一致。瓶颈残差块的 ResNet 网络由 3 层卷积层组成,其中第一层和第三层都是 1×1 的卷积层,中间一层是 3×3 的卷积层。裁剪时需考虑到 3 层卷积核尺寸的不同,选择裁剪中间层的输入和输出通道数,并相应调整第一层和第三层的输入和输出通道数。

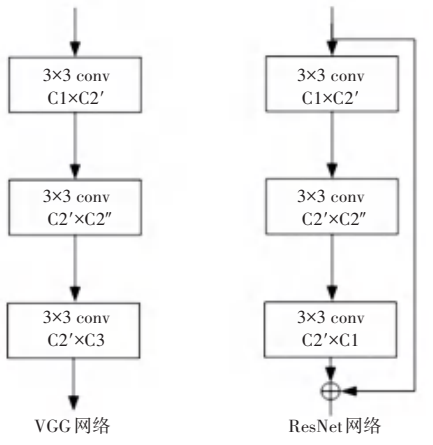

图 3 VGG、Resnet 网络剪枝策略

Fig. 3 VGG, Resnet network pruning strategy diagram

2 实验分析

为了检验本文所提算法的有效性,本文在 Linux 操作系统下,基于 Pytorch 框架选用 VGG-16 与 ResNet56,在 CIFAR-10/100 数据集上进行实验,并与现有的一些剪枝方法进行性能对比。

2.1 数据集与参数设置

网络模型压缩常用的分类数据集有 MNIST、CIFAR-10/100 与 ImageNet。由于 MNIST 内容相对单一,仅包含手写数字,不适合测试模型在复杂场景中的泛化能力,ImageNet 需要较大的计算资源和存储来处理和训练模型,所以实验选用 CIFAR-10/100 数据集用于本文所提算法的评估。CIFAR-10 数据集包含飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船与卡车等 10 个类别,每个类别有 6 000 张 32×32 像素的彩色图像,总计 60 000 张图像。CIFAR-10 数据集是一个中等规模的数据集,适用于测试模型在多类别分类任务上的性能。CIFAR-100 数据集是在 CIFAR-10 的基础上扩展而来,包含 100 个类别,每个类别有 600 张图像,其中包含 20 个精细类别和 80 个粗糙类别。每个精细类别包含 5 个粗糙类别,因此粗糙类别表示更广泛的类别,而精细类别表示更具体的类别。CIFAR-100 数据集的类别更加丰富和多样,对模型的泛化性能提出了更高的要求,因为模型需要能够区分更多不同的类别。

为了帮助模型更好的泛化,提升在不同场景的性能,本文对 CIFAR-10 和 CIFAR-100 数据集采用随机裁剪、随机翻转、颜色抖动等方法进行数据增强。本文采用的优化器为适应性矩估计法(Adaptive Moment Estimation, Adam)。初始学习率

设为 0.1，并以总训练周期的一半和 75%时缩小十倍学习率更新训练。Epoch 设置为 160，BatchSize 大小设为 64，当权重衰减为 1×10^{-4}，动量变为 0.9。融合注意力模块中的降维因子与增维因子 r 设置为 16，并将 BN 层加在卷积层的后面即激活层之前，激活输出阈值中的参数 θ 设为 1.2。

2.2　性能指标

对于进行双通道滤波器剪枝后的网络，本文选用模型准确率（*Accuracy*）、模型参数量（*Parameters*）和浮点运算次数（*FLOPs*）来评估剪枝算法性能。

1）*Accuracy*

准确率即分类正确的样本数除以总样本数，准确率越高，模型分类越好。对于 CIFAR-10、CIFAR-100 分类数据集均采用 Top-1 的分类预测准确度作为网络模型的性能指标。计算公式如下：

$$Accuracy = \frac{TP + TN}{FN + FP + TN + TP} \tag{8}$$

式中：*FN* 表示实际为正样本被判定为负样本的个数，*FP* 表示实际为负样本被判定正样本的个数，*TN* 表示实际是负样本被判定负样本的个数，*TP* 表示实际是正样本被判定正样本的个数。

2）*Parameters*

参数量就是模型的大小，每层卷积操作参数量为：

$$Params_{\text{conv}} = C_{\text{out}} \times (k_w \times k_h \times C_{\text{in}} + 1) \tag{9}$$

其中，C_{out} 表示输出通道数；C_{in} 表示输入通道数；k_w 表示卷积核宽；k_h 表示卷积核高。

3）*FLOPs*

FLOPs 表示网络模型进行一次前向推理浮点运算的总操作数，常用于度量网络模型的计算复杂度。一般浮点运算量越小，网络模型推理速度越快。卷积层的浮点运算量计算公式为：

$$FLOPs_{\text{conv}} = [(C_{\text{in}} \times k_w \times k_h) + (C_{\text{in}} \times k_w \times k_h - 1) + 1] \times C_{\text{out}} \times W \times H \tag{10}$$

其中，H 和 W 表示输入特征图的高和宽。

2.3　实验结果与分析

实验仅对卷积层进行通道剪枝，全连接层会自动匹配前后卷积层的数量。

2.3.1　CIFAR-10 数据集实验结果分析

本次实验基于 CIFAR-10 数据集，在 VGG-16、ResNet56 两种典型的神经网络模型中验证所提出方法的有效性。VGG-16 网络在压缩前后的网络通道数对比结果见表 1。

表 1　VGG-16 在 CIFAR-10 数据集上裁剪前后各层通道数对比

Table 1　Comparison table of the number of channels in each layer of VGG-16 before and after cropping on the CIFAR-10 dataset

卷积层	原始通道数	剪枝后通道数	剪枝率/%
Conv1_1	64	33	48
Conv1_2	64	34	46
Conv2_1	128	65	49
Conv2_2	128	61	52
Conv3_1	256	118	53
Conv3_2	256	121	52
Conv3_3	256	139	47
Conv4_1	512	281	45
Conv4_2	512	361	29
Conv4_3	512	369	27
Conv5_1	512	307	40
Conv5_2	512	373	27
Conv5_3	512	410	19

从表中可以观察到，在前 6 层卷积层的剪枝率基本相似，随着卷积层数的加深，裁剪的通道数逐渐增多。表明在 VGG 网络中，较浅的卷积层在特征提取方面发挥着更为关键的作用，因此裁剪较少的特征图可使其输出的通道数增多，从而传达更多的特征信息。相反，深层卷积层因通道数增多且尺寸减小，其特征信息更为集中，而通道的冗余性也随之增加。此外，数据还呈现同一阶段的剪枝率通常逐渐减小，后面层保留的通道数普遍多于前面层。

由表 2 呈现的比对结果可知，从准确率来看 NS 算法最高达到了 93.11%，但文本算法与原始 VGG-16 准确率相差无几。从减少的参数量来说，本文算法减少的参数量无疑是最低的，比原始算法减少了近 12.5 M，比 NS 算法少了 0.18 M。从浮点运算次数来看，本文算法比原始算法与 NS 算法降低了近 56.3%与 10.8%。

表 2　本文所提方法与其他方法在 VGG-16 上的性能比较

Table 2　Performance comparison of the proposed method with other methods on VGG16

方法	Top-1 *Accuracy*/ %	*Parameters*/ M	*FLOPs*/ M
VGG-16	93.02	14.73	314.69
文献[7]	92.03	2.56	145.61
文献[15]	93.02	3.95	183.11
文献[16]	92.03	3.38	189.76
文献[17]	93.11	2.34	154.19
本文方法	92.94	2.16	137.39

综上所述,本文算法虽然在准确率上比 NS 稍低一筹,但参数量与浮点运算次数都优于其他算法,实现了 VGG-16 网络的高效压缩。

在 CIFAR-10 数据集上,对 ResNet56 网络进行剪枝后各个卷积层的剪枝率如图 4 所示。此网络模型共有 56 层,在某层中可以出现保留全部的通道数,也可出现裁剪全部的通道数,集中反映了不同层次对于整个网络模型的重要程度,并且 ResNet56 网络的冗余通道主要分布在较深的层中。

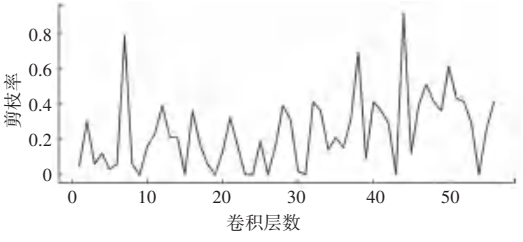

图 4 ResNet56 在 CIFAR-10 数据集上的各层剪枝率情况

Fig. 4 ResNet56 pruning rates by layer on the CIFAR-10 datasets

表 3 则展示了本文所提算法与其他剪枝算法在 CIFAR-10 数据集上对 ResNet56 的性能比较。文献[7]中算法在参数量与浮点运算次数减少量上虽与本文算法相近,但准确率却比本文算法减少了 0.84%。从表中可以看出,虽然以上所有算法在准确率上都没超过原始 ResNet56 模型,但文献[18]、文献[19]以及本文算法等皆非常接近原算法。从减少的参数量看出,本文算法较原始算法减少 0.6 M,对比以上准确率高的算法皆是大幅度降低参数量。综上,本文所提方法在最小的浮点运算次数和参数量的情况下达到了最高的准确率。

表 3 本文所提方法与其他方法在 ResNet56 上性能比较

Table 3 Performance comparison of the proposed method with other methods on ResNet56

方法	Top-1 *Accuracy*/ %	*Parameters*/ M	*FLOPs*/ M
ResNet56	93.24	0.85	126.32
文献[7]	91.94	0.27	35.63
文献[16]	91.55	0.29	50.11
文献[18]	93.02	0.73	90.31
文献[19]	93.04	0.72	90.04
本文方法	92.78	0.25	34.24

2.3.2 CIFAR-100 数据集实验结果分析

为进一步验证本文提出方法的有效性,实验选用较大数据集 CIFAR-100 在 VGG-16 与 ResNet56 上进行了实验。

从表 4 中的数据可以观察到,在除了首层之外的前几层,剪枝通道的比例均较低,这表明第一层存在着大量的参数冗余,而在随后的几层中,对网络性能产生显著影响,因此需要较多地保留这些层。从 Conv4_1 层开始,剪枝通道比例显著上升,甚至在 Conv4_1 层达到了 60%。这表示该层的参数冗余相当明显,随后的剪枝通道比例逐渐下降。这也表明了网络深层的卷积层对整体性能逐渐发挥着更大的作用,需要保留更多通道以维持网络性能。进一步证明了本文提出的通道剪枝方法在不同情境下具备分辨网络通道重要性的能力,能够自适应调整各层内需要剪枝的通道数,以减少网络中的冗余通道数。

表 4 VGG-16 在 CIFAR-100 数据集上裁剪前后的各层通道数对比

Table 4 Comparison table of the number of channels in each layer of VGG-16 before and after cropping on the CIFAR-100 dataset

卷积层	原始通道数	剪枝后通道数	剪枝率/%
Conv1_1	64	41	35
Conv1_2	64	63	1.5
Conv2_1	128	120	6.2
Conv2_2	128	124	3.1
Conv3_1	256	198	22
Conv3_2	256	236	7.8
Conv3_3	256	256	0
Conv4_1	512	201	60
Conv4_2	512	239	53
Conv4_3	512	314	38
Conv5_1	512	361	29
Conv5_2	512	402	21
Conv5_3	512	396	22

图 5 中的折线图展示了在 CIFAR-100 数据集上对 ResNet56 网络进行剪枝后各个卷积层的剪枝率。从图中可以看出,在 CIFAR-100 数据集中,这些冗余通道则主要分布在较浅层和较深层。对比 CIFAR-10,ResNet56 网络在 CIFAR-100 数据集上的整体剪枝率较低,表明在 CIFAR-100 数据集上,ResNet56 网络的参数冗余相对较低。

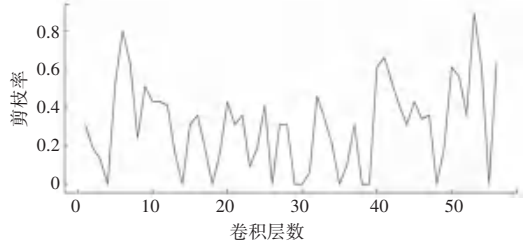

图 5 ResNet56 在 CIFAR-100 数据集上的各层剪枝率情况

Fig. 5 ResNet56 pruning rates by layer on the CIFAR-100 datasets

不同算法在 CIFAR-100 数据集下 VGG-16 与 ResNet56 的剪枝性能比较结果见表 5、表 6。在 VGG-16 对比中，对于准确率来讲，文献[17]中算法最接近原始准确率，但其浮点运算次数只下降 23.9%，浮点运算次数有小幅度减少，文献[20]中算法虽然浮点运算减少 42.99%，但其准确率也下降了 1.89%，文献[12]中算法准确率和浮点运算次数均保持较好的值。而本文算法在准确率只下降 0.77%的基础上，浮点运算次数却减少了 47.92%。

表 5　本文所提方法与其他方法在 VGG16 上的性能比较

Table 5　Performance comparison of the proposed method with other methods on VGG16

方法	Top-1 *Accuracy*/ %	*FLOPs*/ M
VGG-16	73.64	313.787
文献[17]	73.14	238.610
文献[19]	71.32	219.148
文献[20]	71.75	178.860
文献[21]	73.61	219.330
本文方法	72.87	163.410

在 ResNet56 中，文献[22]SFP、文献[23]FPGM 与本文算法浮点运算次数降低量相差无几，但本文算法准确率比前者高，而文献[24]与文献[25]算法虽与本文在准确率上几乎相同，但其浮点运算降低次数均比本文低，从整体来看，本文所提算法均优于以上算法。

表 6　本文所提方法与其他方法在 ResNet56 上的性能比较

Table 6　Performance comparison of the proposed method with other methods on Resnet56

方　法	Top-1 *Accuracy*/ %	*FLOPs*/ M
ResNet56	71.41	172.27
文献[22]	68.79	59.14
文献[23]	69.65	59.52
文献[24]	70.82	61.35
文献[25]	70.63	67.47
本文方法	70.79	57.21

3　结束语

针对现有的神经网络参数大、网络过深，在硬件上部署困难等情况，本文提出了一种基于注意力机制的通道剪枝方法。在减少输入通道的数量上，本文融合了 Transformer 中的自注意力机制与 SENet 中的通道注意力机制，使得网络能够有效的聚焦于重要通道，从而区分出冗余通道。在 CIFAR-10 和 CIFAR-100 数据集上，对常见的卷积神经网络 VGG-16 和 ResNet56 进行实验，用于验证本章所提算法的有效性。实验结果表明，在不明显降低网络性能的前提下，本文算法能够有效地减少模型中的参数和通道数。

参考文献

[1] 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.

[2] GRAVES A, JAITLY N. Towards end-to-end speech recognition with recurrent neural networks[C]//Proceedings of International Conference on Machine Learning. IMLS, 2014: 1764-1772.

[3] JOZEFOWICZ R, VINYALS O, SCHUSTER M, et al. Exploring the limits of language modeling[J]. arXiv preprint arXiv, 1602.02410, 2016.

[4] ANWAR S, HWANG K, SUNG W. Structured pruning of deep convolutional neural networks[J]. ACM Journal on Emerging Technologies in Computing Systems, 2017, 13(3): 1-18.

[5] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. Fibers, 2015, 56(4):3-7.

[6] PENG H, WU J, CHEN S, et al. Collaborative channel pruning for deep networks[C]// Proceedings of International Conference on Machine Learning. IMLS, 2019: 5113-5122.

[7] 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. Piscataway, NJ: IEEE, 2020: 1529-1538.

[8] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient inference[J]. arXiv preprint arXiv, 1611.06440, 2016.

[9] 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. Piscataway, NJ: IEEE, 2017: 5058-5066.

[10] 赵丽君，周永军，汤小红，等. 无人驾驶深度学习模型组合剪枝算法[J]. 传感器与微系统，2021，40(3)：127-129.

[11] TANG Y, WANG Y, XU Y, et al. Manifold regularized dynamic network pruning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 5018-5028.

[12] NING X, ZHAO T, LI W, et al. DSA: More efficient budgeted pruning via differentiable sparsity allocation[C]// Proceedings of European Conference on Computer Vision. Cham: Springer, 2020: 592-607.

[13] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv, 1706.03762, 2017.

[14] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 7132-7141.

[15] HUANG Z, WANG N. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European Conference on Computer Vision (ECCV). Cham: Springer, 2018: 304-320.

[16] 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. Piscataway, NJ: IEEE, 2019: 2790-2799.

[17] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2017: 2736-2744.

[18] LIU Z, SUN M, ZHOU T, et al. Rethinking the value of network pruning[J]. arXiv preprint arXiv,1810.05270, 2018.

[19] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient convNets[C]// Proceedings of International Conference on Learning Representations. ICLR,2017:1-13.

[20] WANG W, FU C, GUO J, et al. Cop: Customized deep model compression via regularized correlation-based filter-level pruning [J]. arXiv preprint arXiv,1906.10337, 2019.

[21] HE Y, LIN J, LIU Z, et al. Amc: Automl for model compression and acceleration on mobile devices[C]//Proceedings of the European Conference on Computer Vision (ECCV). Cham: Springer, 2018: 784-800.

[22] HE Y, KANG G, DONG X, et al. Soft filter pruning for accelerating deep convolutional neural networks[J]. arXiv preprint arXiv,1808.06866, 2018.

[23] 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. Piscataway, NJ: IEEE, 2019: 4340-4349.

[24] DONG X, YANG Y. Network pruning via transformable architecture search[C]// Proceedings of Advances in Neural Information Processing Systems. 2019:10372-10383.

[25] LU X, HUANG H, DONG W, et al. Beyond network pruning: a joint search-and-training approach[C]//Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence. Piscataway, NJ: IEEE, 2021: 2583-2590.