研究与开发

深度卷积神经网络的柔性剪枝策略

陈靓[1,2]，钱亚冠[1,2]，何志强[1,2]，关晓惠[3]，王滨[4]，王星[4]

（1. 浙江科技学院理学院/大数据学院，浙江 杭州 310023；
2. 海康威视–浙江科技学院边缘智能安全联合实验室，浙江 杭州 310023；
3. 浙江水利水电学院信息工程与艺术设计学院，浙江 杭州 310023；
4. 浙江大学电气工程学院，浙江 杭州 310063）

摘　要：尽管深度卷积神经网络在多种应用中取得了极大的成功，但其结构的冗余性导致模型过大的存储容量和过高的计算代价，难以部署到资源受限的边缘设备中。网络剪枝是消除网络冗余的一种有效途径，为了找到在有限资源下最佳的神经网络模型架构，提出了一种高效的柔性剪枝策略。一方面，通过计算通道贡献量，兼顾通道缩放系数的分布情况；另一方面，通过对剪枝结果的合理估计及预先模拟，提高剪枝过程的效率。基于 VGG16 与 ResNet56 在 CIFAR-10 的实验结果表明，柔性剪枝策略分别降低了 71.3%和 54.3%的浮点运算量，而准确率仅分别下降 0.15 个百分点和 0.20 个百分点。

关键词：卷积神经网络；网络剪枝；缩放系数；通道贡献量

中图分类号：TP183

文献标志码：A

doi: 10.11959/j.issn.1000−0801.2022004

A flexible pruning on deep convolutional neural networks

CHEN Liang[1,2], QIAN Yaguan[1,2], HE Zhiqiang[1,2], GUAN Xiaohui[3], WANG Bin[4], WANG Xing[4]

1. School of Science/School of Big-data Science, Zhejiang University of Science and Technology, Hangzhou 310023, China
2. Hikvision-Zhejiang University of Science and Technology Edge Intelligence Security Lab, Hangzhou 310023, China
3. College of Information Engineering & Art Design, Zhejiang University of Water Resources and Electric Power, Hangzhou 310023, China
4. College of Electrical Engineering, Zhejiang University, Hangzhou 310063, China

Abstract: Despite the successful application of deep convolutional neural networks, due to the redundancy of its structure, the large memory requirements and the high computing cost lead it hard to be well deployed to the edge devices with limited resources. Network pruning is an effective way to eliminate network redundancy. An efficient flexible pruning strategy was proposed in the purpose of the best architecture under the limited resources. The contribution of channels was calculated considering the distribution of channel scaling factors. Estimating the pruning result and simulating in advance increase efficiency. Experimental results based on VGG16 and ResNet56 on CIFAR-10 show that the flexible pruning reduces FLOPs by 71.3% and 54.3%, respectively, while accuracy by only 0.15 percentage points and 0.20 percentage points compared to the benchmark model.

Key words: convolutional neural network, network pruning, scaling factor, channel contribution

---

收稿日期：2021−08−04；**修回日期：**2021−12−13

基金项目：国家重点研发计划项目（No.2018YFB2100400）；国家自然科学基金资助项目（No.61902082）

Foundation Items: The National Key Research and Development Program of China (No.2018YFB2100400), The National Natural Science Foundation of China (No.61902082)

0 引言

从最初的 8 层 AlexNet[1]，到上百层的 ResNet[2]，深度学习技术的发展得益于日益加深的网络结构，但同时受其制约。一个 152 层的 ResNet 有超过 6 000 万个参数，在推断分辨率为 224 dpi×224 dpi 的图像时需要超过 200 亿次浮点运算（floating point operation，FLOP）。尽管这在拥有大量高性能 GPU 的云平台上不成问题，但对于移动设备、可穿戴设备或物联网设备等资源受限的平台是无法承受的。然而，集中化的云端推理服务存在带宽资源消耗大、图像数据隐私泄密严重、时效性难以保证等问题[3]。因此，如何在边缘端部署深度神经网络模型是解决问题的关键。

为此学术界提出了很多压缩卷积神经网络的方法，包括低秩近似[4]、网络量化[5]和网络剪枝等，其中剪枝技术被证实是一种有效的方法。剪枝方法按照移除的粒度大小可以被分为两大类：权重剪枝和结构化剪枝。权重剪枝[6]也被称作非结构化剪枝，可以剪除网络中的任意连接权重。尽管它具有较高的剪枝率，但稀疏权重矩阵的存储和关联索引需要特定的计算环境，在实际中难以实现硬件加速[7-8]。结构化剪枝[9-12]直接减小权重矩阵的大小，同时保持完整矩阵的形式，因此它可以更好地兼容硬件进行加速，成为目前的主流研究方向。

早期的剪枝方法[12-14]通常根据权重重要性决定哪些通道被剪除，忽略权重重要性之间相互联系，从而容易造成过度剪枝或剪枝不足的问题，导致剪枝后模型性能不可复原地下降。最新的研究通常使用渐进的迭代剪枝方法[15]，或将网络结构搜索问题转化为优化问题[16]，逐步逼近最佳的网络架构。然而当这些方法应用于大剪枝率时，受到网络稀疏程度的影响，在实际剪枝过程中会带来额外的计算量。

为了能够更高效地剪枝得到最佳的模型结构，本文提出一种柔性剪枝策略。该方法以通道为单位剪枝，首先结合批归一化（batch normalization，BN）层的缩放系数值及其分布情况，计算通道贡献量作为衡量通道重要性的依据，并初步估算每层的剪枝比例。然后通过模拟剪枝调优剪枝策略，以此快速逼近最佳的网络架构。最后根据得到的架构对每一层分别剪枝，获得紧凑的模型。本文提出的剪枝方法，采用全局的结构化剪枝方式，过程中不需要对局部手动调参，剪枝后的模型无须特定的环境支持，是一种高效的模型压缩方法。

1 相关工作

剪枝作为网络压缩的有效方法之一，通过裁剪网络中不重要的权重，在不影响性能的情况下，提升网络运算速度，减少存储容量。早在 1989 年，LECUN 等[17]就指出神经网络中存在大量冗余，删除网络中不重要的权重能够获得更好的泛化能力，消耗更少的训练和推理时间。文献[6]通过修剪网络中的不重要连接，减少网络所需要的参数，减少内存和计算消耗。尽管权重剪枝在降低网络参数上有显著的效果，但会导致模型的非结构化，很难实现硬件加速[7-8]。

为兼容现有的硬件加速，近期的剪枝工作[9-12]集中在结构化剪枝上，通过修剪整个通道或者卷积核以保持结构的规则性，并提出了多种重要性度量标准。文献[9]使用卷积核的 L1 范数作为重要性度量，计算卷积核中所有权值的绝对值之和，从而规则地删除数值较小的卷积核避免稀疏连接。文献[10]通过计算激活层后神经元中数值为 0 的比例衡量神经元的重要性。文献[11]根据删除一个通道后对下一层激活值的影响衡量该通道重要性，寻找这一层输入的最优子集代替原来的输入，得到尽可能相似的输出，剪除子集外的通道。文献[12]提出网络瘦身（network slimming，NS）方法，用 BN 层的缩放系数作为通道重要性的评价

标准，根据全局阈值剪除缩放系数小于阈值的通道。

NS 方法[12]是一种经典的自动化剪枝方法[18]，可以根据整体剪枝率自动地生成剪枝后的网络架构，无须人为设计每层的剪枝比例。文献[19]指出，NS 方法中基于预定义的全局阈值（optimal thresholding，OT）的设计忽略了不同层之间的变化和权重分布，并提出一种最优阈值设定策略，根据不同层的权重分布设计各层的最优阈值。文献[20]同样基于 NS 方法，提出了一种极化正则化器（polarization regularizer，PR），通过调整稀疏化训练的策略，对不同重要性的通道进行区分。

常规的剪枝方法是一个不可逆过程，被剪除的通道或者卷积核不参与后续训练过程，也被称为“剪枝”。文献[21]和文献[15]提出“剪枝”法，在训练期间将被剪除的卷积核参数重置为零，并在后续训练阶段持续更新，以此降低过度剪枝风险。文献[22]在此基础上提出 ASRFP 方法，渐进地修剪卷积核，使得训练和剪枝过程更加稳定。文献[23]提出 BN-SFP 方法，根据 BN 层缩放系数进行软剪枝，实现性能的提升。

最新的研究工作通过自动搜索最优子网络结构实现网络剪枝，文献[24]预先训练大型 PruningNet 预测剪枝模型的权重，通过一种进化搜索方法搜索性能良好的修剪网络。文献[25]提出了近似的 Oracle 过滤器剪枝，用二分搜索的方式确定每层的剪枝数。文献[26]基于人工蜂群算法，搜索最优的剪枝架构。

相比于以往的剪枝方法通过去除冗余或者结构搜索的方式，渐进地逼近目标压缩模型，柔性方法的优越性体现在，通过分析模型的权值分布和对目标压缩模型的快速模拟，以少量计算成本获得高性能的压缩模型架构。

2　方法描述

神经网络的前向传播是一个连续过程，剪除其中一部分权重必然对后续的计算产生影响。原先的 NS 方法中基于预定义全局阈值的设计忽略了权重的关联性。另外，网络架构的改变会对权重的分布产生影响，即当对神经网络模型进行迭代剪枝时，每轮迭代后的模型由于架构发生改变，权重分布也会相应发生变化。当网络架构越接近于最佳的剪枝后架构时，权重分布也越接近。据此本文提出柔性剪枝策略，结合 BN 层缩放系数的分布情况，构建一个新的通道重要性评价指标——通道贡献量。并提出一种通道等效重组方法，对剪枝后的网络架构进行模拟，获得接近于最佳网络架构的权重分布。

柔性剪枝算法的流程如图 1 所示，整个流程可以分为获取架构和获取参数两个部分。通过计算通道贡献量、通道等效重组和适配调整缩放系数 3 个步骤获取压缩模型的架构，即剪枝后模型各层剩余的通道数。然后以此进行硬剪枝，微调获取压缩模型参数。其中，通道贡献量的计算包括阈值函数平滑和位序相关性加权两步，本文将在下面详细地阐述实施细节。需要指出的是，本文提出的柔性剪枝在模拟训练过程中提供了更多的网络架构调整机会，有效地避免一次性剪枝带来的过度剪枝问题。

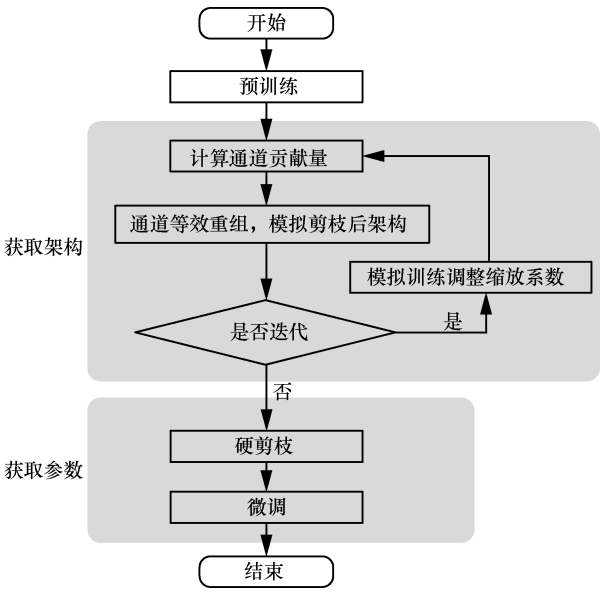


图 1　柔性剪枝的流程

2.1 阈值函数平滑

深度神经网络[2,27]广泛采用BN[14]加速训练过程的收敛，促进神经网络稳定训练，用z^{in}、z^{out}分别表示BN层的输入和输出：

$$z^{\text{out}}=\gamma\frac{z^{\text{in}}-\hat{\mu}}{\sqrt{\hat{\sigma}^2+\varepsilon}}+\beta \tag{1}$$

其中，$\hat{\mu}=E[z^{\text{in}}]$，$\hat{\sigma}^2=\text{Var}[z^{\text{in}}]$，$\gamma$和$\beta$分别是在BN层中学习到的缩放系数和偏置项，$\varepsilon$是数据平衡的小正值。BN层的缩放系数$\gamma$直观地反映输入的改变对当前通道输出的影响程度，一定程度上体现该通道在整个分类任务中发挥的作用。基于NS方法，柔性剪枝使用缩放系数作为通道重要性的重要依据，在训练过程中使用L1正则化对缩放系数γ进行稀疏化训练[12]。

由于缩放系数局限于反映单个通道的重要性，因此在缩放系数的基础上提出通道贡献量，以便于扩展到对整个层的度量。通道贡献量的设计初衷是用于抑制通道的输出，实现模拟剪枝。NS方法根据全局阈值，剪除缩放系数小于全局阈值的通道，保留缩放系数大于全局阈值的通道。其剪枝过程可以视为将通道输出乘以一个关于缩放系数的阈值函数，类似于阶跃函数，即缩放系数小于阈值的通道输出乘以0，缩放系数大于阈值的通道输出乘以1。然而这会造成被剪除部分信息不可恢复的损失，并且无法体现不同通道的重要性差异，因此为便于对剪枝后架构的模拟，将其改造成平滑度可控的函数。

首先，根据通道缩放系数的大小，设计一个平滑函数：

$$S(\gamma)=1/\left(1+\frac{P}{1-P}\exp\left(\frac{\gamma_{\text{C}}-\gamma}{u/10}\right)\right) \tag{2}$$

其中，γ表示缩放系数；P表示给定的剪枝率；γ_{C}表示给定剪枝率P下的全局阈值；平滑系数u用控制通道重要性的差异程度。（1）当$u=0$时，$S(\gamma)=0$或1，此时等价于硬剪枝，将通道极端地区分为“有用”和“无用”两种类型；（2）当逐步增大平滑系数u，不同通道的重要性差异逐渐缩小；（3）当$u\to\infty$时，所有通道的重要性完全相同。图2展示了函数的阈值$\gamma_C=0.35$，剪枝率$P=80\%$，平滑系数分别为$u=0,0.5,1,2,\infty$时的平滑函数。由此可见，利用平滑函数可更灵活地控制通道重要性的分布。为避免缩放系数为0的通道被误认为具有贡献量，即使得$S(0)=0$，因此进行归一化处理：$(S(\gamma)-S(0))/(1-S(0))$。

2.2 位序相关性加权

在CIFAR10上训练VGG16模型中全局缩放系数的分布如图3所示，通道缩放系数呈现多峰分布，缩放系数在某些区间（±0.01）的数量高度集中，使得在数值上非常接近的缩放系数在排序后的位序上间隔很大。这种现象在函数平滑后仍然存在，如图3中的两组缩放系数(x_1,x_2)和$S(\gamma)$ (x_3,x_4)，虽然$(x_2-x_1)\approx(x_4-x_3)$，但在位序距离上，$(x_1,x_2)\gg(x_3,x_4)$。利用平滑函数的连续光滑特性，提出通道贡献量替代缩放系数，充分体现分布密集的缩放系数的排序信息，克服由于稀疏化训练导致初始网络中的缩放系数分布过于集中的现象。同时，把通道贡献量作为新的剪枝度量，还需要与整体剪枝率相关联。当剪枝率为P

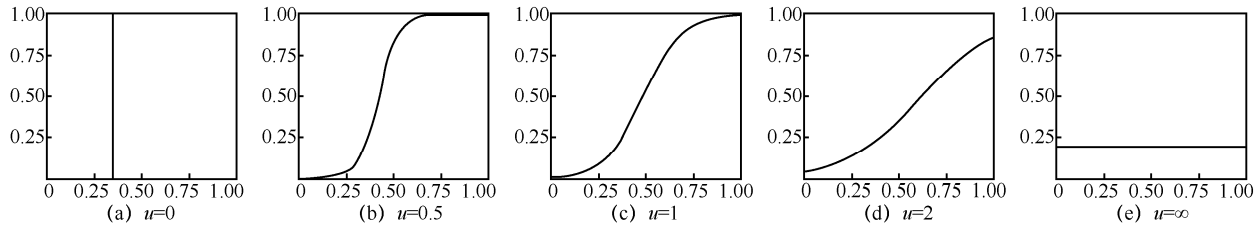


图2 不同平滑系数下的平滑函数

时，剪枝后的剩余通道数为 $C=N(1-P)$，假设完整保留通道的贡献量为 1，那么网络总贡献量则为 $M_{\text{sum}}=C\times 1$。柔性剪枝策略要求满足 $\sum_{k=1}^{N}M_k(\gamma)=C$。因此，综合考虑剪枝率、缩放系数的位序分布，提出位序相关性加权方法计算通道贡献量 $M_k(\gamma)$。将所有通道的平滑函数值按降序排列：$S_1\geqslant\cdots\geqslant S_k\geqslant\cdots\geqslant S_N$，其中，$k$ 为通道索引。通道 k 的贡献量表示为：

$$M_k=\frac{C}{r+1}\left(S_k+\sum_{l=1}^{r}\frac{r+1-l}{r+1}(S_{k-l}+S_{k+l})\right)/\sum_{k=1}^{N}S_k \quad (3)$$

其中，r 表示在加权过程中使用了当前通道位序前后 r 项的值，体现了序列中的相邻通道对于当前通道贡献量的影响程度。

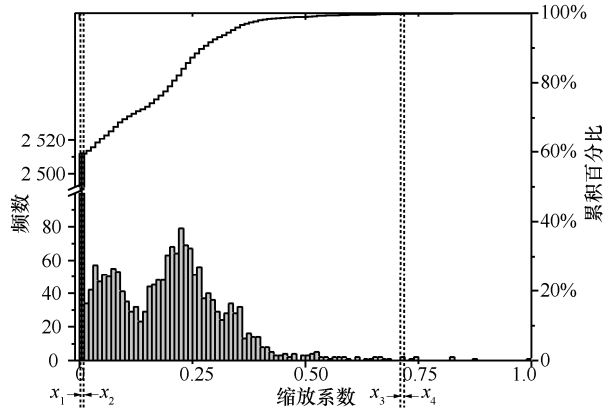


图 3　全局缩放系数的分布

2.3　通道等效重组

文献[15]提到，直接删除大量包含训练集信息的通道，会导致严重的信息丢失，从而不可避免地降低剪枝后网络性能。为了避免过度剪枝，在正式剪枝前进行模拟剪枝。通过将通道输出乘上贡献量作为下一层的输入，起到抑制通道的作用，以此模拟剪枝后网络的输出，此处贡献量视为抑制通道的程度。具体而言，贡献量在[0,1]区间取值，当通道贡献量 $M_k(\gamma)=1$ 时，该通道不受抑制，等价于被完全保留；当通道贡献量 $M_k(\gamma)=0$ 时，该通道被完全抑制，等价于被剪除；而贡献量 $0<M_k(\gamma)<1$ 时，表示通道输出被部分抑制。通过通道抑制，在保证各通道完整的前提下，利用贡献量等效原则进行通道重组，达到模拟剪枝的效果。

通道等效重组的过程示意图如图 4 所示。假设 A、B 和 C 通道的贡献量分别为 $M_{\text{A}}=1$、$M_{\text{B}}=M_{\text{C}}=M_{\text{D}}=0.33$，利用贡献量对一个 4 通道进行抑制。在阈值剪枝策略下，除了 A 通道会被保留外，B、C 和 D 通道可能都会被剪除，导致过度剪枝。但采用柔性剪枝策略后，A 通道没有受到抑制，被完整地保留；而被抑制后的 B、C、D 通道输出的总和可等效于一个通道 E 的输出。由此，便可以将 4 通道被抑制后的输出等效于 2 通道的输出，在保证剪枝后网络性能和剪枝率的同时，避免过度剪枝。对于神经网络中的第 i 层，其通道分别对应贡献量 $M_j^{(i)}$，对一整层进行通道等效重组后，该层通道数近似于层通道贡献量的总和，表示为：

$$C^{(i)}=\left\lceil\sum_{j=1}^{N}M_j^{(i)}\right\rceil \quad (4)$$

其中，$\lceil\cdot\rceil$ 表示向上取整操作。这是由于通道贡献量的计算结果为实数，而通道数量为整数，使用向上取整操作还能够避免出现网络断裂的极端情况。此时模拟剪枝后的网络架构可表示为 $[C^{(1)},C^{(2)},C^{(3)},\cdots,C^{(L)}]$，$L$ 为总层数。给定剪枝率

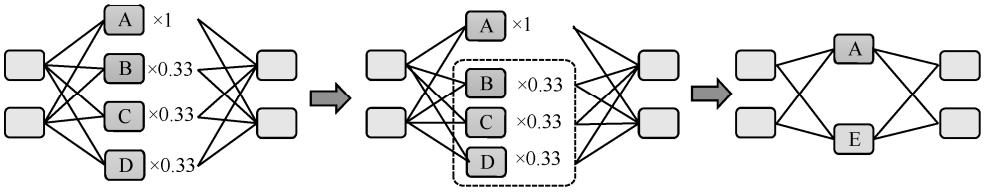


图 4　通道等效重组的过程示意图

P，此时总通道数近似地满足整体的剪枝率，即 $P \approx 1-\sum_{i=1}^{N} C^{(i)} / N$。

使用通道贡献量对初始模型进行等效重组的过程，可以视作一种模拟剪枝方法，此时虽然保留了模型所有信息的完整，但是各层的输出近似于被剪枝之后的输出。

2.4 缩放系数的适配调整

由于不同的网络架构会影响缩放系数的分布情况。因此，在模拟剪枝后再训练得到新的缩放系数，相比于之前的缩放系数，与模拟剪枝的架构更加适配。在实验的计算过程中，使用新得到的 BN 层缩放系数γ与通道被抑制程度的乘积作为此时该通道的缩放系数：

$$\hat{\gamma}_j^{(i)} = M_j^{(i)} \times \gamma_j^{(i)} \tag{5}$$

然后，使用新得到的通道缩放系数$\hat{\gamma}_j^{(i)}$更新通道贡献量$M_j^{(i)}$，再根据式（4）得到新一轮的网络架构$[C^{(1)}, C^{(2)}, C^{(3)}, \cdots, C^{(L)}]$。通过根据 BN 层缩放系数变动情况，进一步调整剪枝策略，使得新一轮模拟剪枝的架构更接近于最佳架构。首次迭代过程中，若剪枝率较大，则重置 BN 层的参数，回调 BN 层学习率并在训练过程中逐渐衰减。这里需要强调的是，等效重组过程是一种剪枝模拟，并不是进行实际的硬剪枝，而等效重组得到的网络架构则是对剪枝的一种模拟。正因为是一种模拟，才可以在保留其信息不丢失的情况下，通过训练对其不断地更新调整，使得网络架构不断优化。

2.5 微调

通过上述步骤，能够获取目标剪枝率下压缩模型的架构。根据各层的剩余通道数，分别进行硬剪枝，根据此时各通道通过式（5）得到的缩放系数大小，剪除数值较低的部分通道。由于柔性剪枝是一步式的剪枝方法，因此最后的训练过程根据剪枝率的大小，提供两种方案。当剪枝率较小时，对剪枝后的模型微调。当剪枝率较大时，考虑使用压缩模型架构直接从头训练（train from scratch）。

3 实验与分析

本文实验代码采用 Pytorch 编程框架，所使用的 GPU 为 NVIDIA RTX 2080Ti。本节将结合实验验证深度卷积神经网络柔性剪枝策略的实际性能。

3.1 实验数据与评价指标

实验采用 CIFAR-10/100[28]数据集，其被广泛应用于评价网络剪枝方法。CIFAR-10/100 数据集都包括 50 000 个训练数据和 10 000 个测试数据，每张图像的大小是 32 dpi×32 dpi，分别有 10 个和 100 个类。按照文献[12]的方法对 CIFAR-10/100 进行预处理图像。

本文选取经典的VGG[27]和ResNet[2]作为剪枝的预训练网络。这两个网络分别作为无跨层连接和有跨层连接的代表，能够最直观地反映出网络压缩对于一般网络模型的效果。需要指出的是，当对 ResNet 上的跨层连接进行剪枝时，通过剪枝得到一个钱包型（wallet）[29]的结构会更有利于剪枝后的网络架构。因此，本文设计一种相对保守的剪枝策略：对于跨层连接的通道，以每次跨层连接前后的 BN 层缩放系数中的最大值作为它的重要性度量，再计算得到跨层连接所需要的通道数。相比于 NS 剪枝完全不对跨层的通道进行裁剪，通过这一策略能够在最大程度保证剪枝后准确率不会骤降的同时，对跨层连接的通道进行裁剪。

实验主要通过剪枝后模型的准确率、浮点运算量和参数量3个评价指标与其他剪枝方法对比。准确率指的是分类神经网络在测试数据集上的 Top-1 分类准确率。浮点运算量用于衡量模型的计算复杂度，浮点运算量越低说明模型实际运算所需的计算量越少，模型加速的效果越好。参数量表示模型占用的内存大小量，参数量的减少可以直接体现模型压缩的效果。

3.2　参数设定

正常训练：使用 SGD 作为优化器，设置动量为 0.9，权重衰减参数为 0.000 1，初始学习率为 0.1。训练 CIFAR-10/100 时，Batch size 设置为 64。共训练 160 个轮次，在第 80（50%）和第 120（75%）个训练轮次处，学习率分别下降为当前学习率的 $\frac{1}{10}$。在本文的实验中，对网络参数使用了 kaiming 初始化，VGG、ResNet 中 BN 层的初始通道缩放分别为 0.5、1，初始偏置为 0。

稀疏训练：与 NS[12]一样，通过对 BN 层缩放系数施加 L1 惩罚项进行稀疏化训练。在 CIFAR 数据集上，对于 VGG16 选择 10^{-4} 作为稀疏率，ResNet56 则选择 10^{-5}。所有其他设置与正常训练保持一致。

柔性剪枝：在本文的实验，通道贡献量计算过程中，平滑系数设置为 $u=1$，超参数 $r=C$。测试中柔性剪枝的迭代次数为 1 次。剪枝过程中，总共训练 80 轮次。

适配调整：若剪枝率小于 30%，使用原网络的学习率微调 80 个轮次。首次迭代时，若剪枝率大于 30%，则初始化 BN 层的系数和学习率，而后续迭代过程中，不再初始化 BN 层系数。训练 80 个轮次，并在第 10（25%）和第 20（50%）训练轮次处，BN 层的学习率下降为 $\frac{1}{10}$。

微调：若剪枝率小于 30%，微调剪枝后的压缩模型 40 个轮次，学习率恒定为 0.001。若剪枝率大于 30%，综合考虑微调训练的结果以及从头训练的结构作为最终的模型准确率，其中，由于考虑到小模型训练起来较为困难，使用文献[18]中提出的同等计算预算的方法进行从头训练，在正常训练的基础上，增加总训练轮次，而学习率初始设置不变，依旧在 50%和 75%下降为当前的 $\frac{1}{10}$。

3.3　CIFAR10/100 上 VGG16 的剪枝实验

在 CIFAR10/100 数据集上修剪 VGG16 的比较结果见表 1。为了保证数据的准确性，直接使用文献[19-20, 25]中的实验数据与柔性剪枝方法进行对比。由于 NS[12]最先提出使用 BN 层的缩放系数作为衡量通道重要性的度量，OT[19]、PR[20]是基于 NS 的最新改进工作，AOFP 降低剪枝过程的运算成本，这些都与柔性剪枝有一定关联，因此后续实验主要与这些工作进行比较。首先修剪 VGG16 模型使其减少与 PR 方法相当的浮点运算

表 1　在 CIFAR10/100 数据集上修剪 VGG16 的比较结果

数据集	剪枝方法	准确率	准确率下降	浮点运算量/10^6	浮点运算量减少
CIFAR-10	基线	93.88%	—	314	—
	NS[12]	93.62%	0.26%	155	51%
	OT[19]	93.87%	0.01%	163	48%
	PR[20]	93.92%	−0.04%	144	54%
	AOFP[25]	94.03%	−0.15%	186	41%
	本文	94.04%	−0.16%	140	55%
CIFAR-100	基线	73.83%	—	314	—
	NS[12]	74.20%	−0.37%	195	38%
	OT[19]	73.99%	0.21%	197	37%
	PR[20]	74.25%	−0.42%	179	43%
	本文	74.33%	−0.50%	174	45%

量，实验结果显示在 CIFAR10/100 数据集上，柔性剪枝方法在浮点运算量较基线模型分别降低 55%和 45%时，准确率为 94.04%和 74.33%，分别比 PR 提高 0.12%和 0.08%，比 OT 提高 0.17%和 0.71%。其中在 CIFAR10 数据集上，柔性剪枝比 AOFP 多减少 14%的浮点运算量，但准确度基本一致。

大剪枝率下 CIFAR10 数据集上修剪 VGG16 的比较结果见表 2。为公平地比较压缩模型网络架构的性能，分别使用文献[18]提出的两种训练方法 Scratch-E、Scratch-B 对各网络架构进行从头训练。其中，Scratch-E 表示使用与预训练相同的 160 个训练轮次，Scratch-B 则使用相同计算预算，如剪枝后的网络比初始模式减少了一半的浮点运算量，则用两倍的训练轮次。表 2 中依次展示通道剪枝率为 74%、76%、80%和 86%时柔性剪枝后压缩模型的网络架构。与 OT 相比，当剪枝率相同时，柔性剪枝方法（剪枝率 74%）得到的架构准确率更高，但浮点运算量更大。而当浮点运算量相当时，柔性剪枝方法（剪枝率 76%）与 OT[19]的结果准确率非常接近。从架构上看，柔性剪枝方法（剪枝率 76%）与 OT 在前 6 层的通道数非常接近，每层的通道数差别都在 10 以内，而在后面几层，柔性剪枝方法则保留了更少的通道。RBP[30]是一种在贝叶斯框架下的层递归贝叶斯剪枝方法，与其相比，在相同的浮点运算量下，柔性剪枝方法（剪枝率 80%）在两种训练方法下的准确率比 RBP 分别提高了 0.49%和 0.53%。在相同的剪枝率下，柔性剪枝方法（剪枝率 86%）虽然在 Scratch-E 训练下的准确率比 RBP 降低了 0.51%，但在 Scratch-B 训练下的准确率提高了 0.17%。其主要原因是柔性剪枝方法（剪枝率 86%）得到的架构浮点运算量太低，只有 RBP 架构的一半。实验结果表明，在大剪枝率下，柔性剪枝方法得到的网络架构性能超过了一些当下最新的剪枝方法。

3.4 CIFAR10/100 上 ResNet56 的剪枝实验

在 CIFAR10/100 数据集上修剪 ResNet56 的比较结果见表 3。为了保证数据的准确性，使用文献[12,15,20-21,23,26]中的实验数据与柔性剪枝方法对比，其中，NS[12]、PR[20]、BN-SFP[23]是基于 BN 层缩放系数的剪枝方法，SFP[21]、BN-SFP、ASFP[15]是软剪枝方法，与柔性剪枝方法存在一定关联。从表 3 中可以看出，在相同浮点运算量下，柔性剪枝方法在 CIFAR10/100 数据集上分别比 NS[12]提高了 0.58%和 1.11%，而与 PR 的方法结果非常接近，仅提高了 0.02%和 0.08%。与软剪枝方法对比中，柔性剪枝具有一定优势，在 CIFAR10 数据集上的准确率分别比 SFP、BN-SFP、ASFP 提高了 0.25%、0.31%、0.48%。除此以外，柔性剪枝方法在浮点计算量减少 48%时，准确率达到

表 2 大剪枝率下 CIFAR10 数据集上修剪 VGG16 的比较结果

剪枝方法	剪枝率	网络架构	总通道数	浮点运算量/10^6	参数量/10^6	准确率（Scratch-E）	准确率（Scratch-B）
基线	0	64-64-128-128-256-256-256-512-512-512-512-512-512	4 224	314	20.04	93.88%	93.88%
NS[12]	70%	30-62-123-126-247-238-199-101-27-10-9-6-89	1 268	155	1.68	93.44%	93.91%
OT[19]	73%	26-59-114-120-206-172-128-98-56-38-27-32-57	1 133	113	1.16	93.28%	93.86%
本文	74%	32-64-115-128-228-198-118-65-25-12-19-22-74	1 100	140	1.26	93.47%	94.04%
本文	76%	31-64-107-128-214-180-103-56-21-10-16-19-59	1 017	116	1.05	93.30%	93.91%
RBP[30]	86%	50-63-123-108-104-57-23-14-9-8-6-7-11	583	9.0	0.39	92.43%	93.20%
本文	80%	26-62-91-121-181-152-86-41-13-7-10-11-61	862	90	0.78	92.92%	93.73%
本文	86%	19-47-65-92-126-103-52-23-6-3-4-5-47	592	47	0.37	91.92%	93.37%

表 3　在 CIFAR10/100 数据集上修剪 ResNet56 的比较结果

数据集	剪枝方法	准确率	准确率下降	浮点运算量/10^6	浮点运算量减少
CIFAR-10	基线	93.80%	—	127	—
	NS[12]	93.27%	0.53%	66	48%
	PR[20]	93.83%	−0.03%	67	47%
	SFP[21]	93.35%	0.45%	59	53%
	BN-SFP[23]	93.29%	0.51%	59	53%
	ASFP[15]	93.12%	0.68%	59	53%
	ABCPruner[26]	93.23%	0.57%	58	54%
	DCP[31]	93.81%	−0.01%	67	47%
	DeepHoyer[32]	93.54%	0.26%	67	47%
	LFPC[33]	93.72%	0.08%	67	47%
	本文	93.85%	−0.05%	66	48%
	本文	93.60%	0.20%	58	54%
CIFAR-100	基线	72.49%	—	127	—
	NS[12]	71.40%	1.09%	96	24%
	PR[20]	72.43%	0.06%	95	25%
	本文	72.51%	−0.02%	95	25%

93.85%，优于一些其他最新剪枝 ABCPruner[26]、DCP[31]、DeepHoyer[32]、LPEC[33]。

3.5　超参数分析

本节针对柔性剪枝方法式中存在的两个超参数，式（2）中的平滑系数 u 和式（3）中的 r，分别进行实验，实验通过柔性剪枝后的模型架构的浮点计算量与精度的变化情况，说明不同取值对于柔性剪枝性能的影响。首先固定 $r=C$，观察平滑系数 u 的取值对柔性剪枝性能的影响，如图 5 所示。如图 5（a）所示，当 u 取值为 0.1 和 0.5 时，准确率较为接近，而当 u 取值为 1 时，准确率最高。而随着 u 取值继续增大，准确率逐渐下降。接着固定平滑系数 u 为 1，观察 r 对剪枝性能的影响。如图 5（b）所示，随着 r 取值的逐渐增大，剪枝的性能也逐渐提高，当 r 取最大值 $r=C$ 时，剪枝后模型的性能最好。故在柔性剪枝中，平滑系数取值为 1，参数 r 取最大值为 C。通过超参数分析实验可知，平滑函数的设计过于陡峭或平缓都不利于剪枝性能，而在位序相关性上更多地考虑周边分布，有助于提升剪枝性能。

3.6　迭代训练分析

为了进一步验证柔性剪枝策略的高效性，本节设计实验分析柔性剪枝方法所需的迭代次数以及每次迭代后再训练的轮次对剪枝效果的影响，如图 6 所示。首先，如图 6（a）所示，固定总训练轮次为 80 轮，分别测试 1 次、2 次、4 次迭代，每次迭代后再训练轮次相同。结果显示，迭代次数的增加不会对柔性剪枝的性能产生较大影响。其次，如图 6（b）所示，测试更多的总训练轮次是否导致更好的剪枝效果，实验中分别测试 80 次、160 次、240 次总训练轮次。结果显示，随着总训练轮次的增加，剪枝性能没有显著提升。通过迭代训练的实验可知，柔性剪枝方法的优越性在于，与传统的迭代剪枝方法不同，柔性剪枝的过程不需要重复“剪枝−再训练”的过程，仅需要消耗较少的运算成本，获得高性能的压缩模型架构。

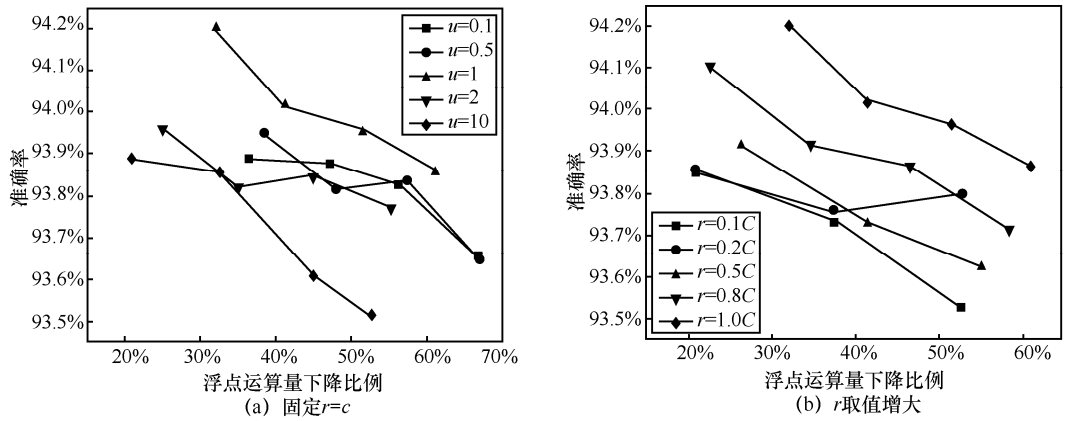


图 5　超参数对剪枝的影响

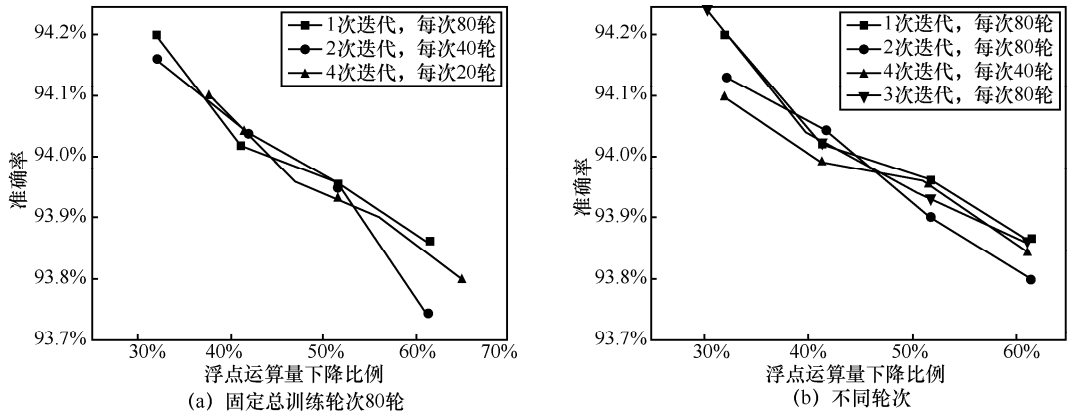


图 6　迭代训练对剪枝的影响

3.7　重构实验分析

柔性剪枝方法不仅适用于网络压缩，同时还可以在浮点运算量相同的条件下，对原网络重构，提高分类准确率。具体地，通过先将原网络架构等比例放大，再对其进行剪枝，从而获得与原网络浮点运算量相同的网络架构，并使得其中一些层的通道数多于原网络中该层的通道数。以此在不减少网络计算的同时调整其各层的通道数，获得更高的准确率。在 CIFAR10/100 数据集上重构 VGG16、ResNet56 的效果见表 4，对 VGG16 和

表 4　在 CIFAR10/100 数据集上重构 VGG16、ResNet56 的效果

数据集	模型	剪枝方法	准确率	准确率提升	浮点运算量/10^6	总通道数
CIFAR-10	VGG16	基线	93.88%	—	314	4 224
		本文（重构）	94.20%	0.32%	313	2 359
	ResNet56	基线	93.80%	—	127	2 032
		本文（重构）	94.04%	0.24%	126	2 181
CIFAR-100	VGG16	基线	73.83%	—	314	4 224
		本文（重构）	74.20%	0.37%	310	3 115
	ResNet56	基线	72.49%	—	127	2 032
		本文（重构）	72.72%	0.23%	114	2 141

ResNet56 进行重构后，再重新训练得到的准确率提升。结果显示，相比原始网络架构，在同等的浮点运算量下，柔性剪枝方法重构的 VGG16 准确率分别提升超过 0.32%和 0.37%，重构的 ResNet56 准确率分别提升超过 0.24%和 0.23%。通过重构实验说明，人工预定义的网络架构对于特定的分类任务往往不是最优的，通过剪枝的方法可以获得同等浮点运算量下准确率更高的架构。

4　结束语

本文提出了一种柔性剪枝策略，一方面，结合神经网络模型中的权重分布情况，设计了通道贡献量作为通道重要性评价指标，以层的视角宏观地设计整层的剪枝方案；另一方面，创造性地提出了一种模拟剪枝方案，使用较少的计算消耗获取高性能的网络架构。在公开数据集上的对比实验表明，柔性剪枝后模型架构的性能优于其他最新的剪枝方法，是一种非常高效的剪枝方法。在后续的工作中，将进一步测试在跨层连接上的剪枝策略，以及将柔性剪枝策略应用于其他最新的轻量级网络。

参考文献：

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[3] 唐博恒, 柴鑫刚. 基于云边协同的计算机视觉推理机制[J]. 电信科学, 2021, 37(5): 72-81.
TANG B H, CHAI X G. Cloud-edge collaboration based computer vision inference mechanism[J]. Telecommunications Science, 2021, 37(5): 72-81.

[4] WANG Y, BIAN Z P, HOU J H, et al. Convolutional neural networks with dynamic regularization[EB]. 2019.

[5] COURBARIAUX M, BENGIO Y, DAVID J P. Binary Connect: training deep neural networks with binary weights during propagations[J]. CoRR, 2015.

[6] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[J]. CoRR, 2015.

[7] WEN W J, YANG F, SU Y F, et al. Learning low-rank structured sparsity in recurrent neural networks[C]//Proceedings of 2020 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2020: 1-4.

[8] HE Y H, ZHANG X Y, SUN J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 1398-1406.

[9] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[EB]. 2016.

[10] HU H Y, PENG R, TAI Y W, et al. Network trimming: a data-driven neuron pruning approach towards efficient deep architectures[EB]. 2016.

[11] LUO J H, WU J X, LIN W Y. ThiNet: a filter level pruning method for deep neural network compression[C]//Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 5068-5076.

[12] LIU Z, LI J G, SHEN Z Q, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2017: 2755-2763.

[13] YE J B, LU X, LIN Z, et al. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers[EB]. 2018.

[14] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift[J]. CoRR, 2015.

[15] HE Y, DONG X Y, KANG G L, et al. Asymptotic soft filter pruning for deep convolutional neural networks[J]. IEEE Transactions on Cybernetics, 2020, 50(8): 3594-3604.

[16] LIN M B, JI R R, ZHANG Y X, et al. Channel pruning via automatic structure search[C]//Proceedings of Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. California: International Joint Conferences on Artificial Intelligence Organization, 2020.

[17] LECUN Y, DENKER J S, Solla S A. Optimal brain damage[C]// Proceedings of the Advances in Neural Information Processing Systems. Berlin: Springer, 1989: 598-605.

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

[19] YE Y, YOU G M, FWU J K, et al. Channel pruning via optimal thresholding[M]//Communications in Computer and Information Science. Cham: Springer International Publishing, 2020: 508-516.

[20] ZHUANG T, ZHANG Z X, HUANG Y H, et al. Neuron-level structured pruning using polarization regularizer[C]// Advances in Neural Information Processing Systems. 2020: 1-13.

[21] RONG J T, YU X Y, ZHANG M Y, et al. Soft Taylor pruning for accelerating deep convolutional neural networks[C]// Pro-

ceedings of IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society. Piscataway: IEEE Press, 2020: 5343-5349.

[22] CAI L H, AN Z L, YANG C G, et al. Softer pruning, incremental regularization[C]//Proceedings of 2020 25th International Conference on Pattern Recognition (ICPR). Piscataway: IEEE Press, 2021: 224-230.

[23] XU X Z, CHEN Q M, XIE L, et al. Batch-normalization-based soft filter pruning for deep convolutional neural networks[C]//Proceedings of 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV). Piscataway: IEEE Press, 2020: 951-956.

[24] LIU Z C, MU H Y, ZHANG X Y, et al. MetaPruning: meta learning for automatic neural network channel pruning[C]// Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 3295-3304.

[25] DING Y X, ZHAO W G, WANG Z P, et al. Automaticlly learning featurs of android apps using CNN[C]//Proceedings of 2018 International Conference on Machine Learning and Cybernetics (ICMLC). Piscataway: IEEE Press, 2018: 331-336.

[26] LIN M B, JI R R, ZHANG Y X, et al. Channel pruning via automatic structure search[C]//Proceedings of Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. California: International Joint Conferences on Artificial Intelligence Organization, 2020: 673-679.

[27] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. CoRR, 2014.

[28] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[J]. Handbook of Systemic Autoimmune Diseases. 2009,1(4).

[29] LUO J H, WU J X. Neural network pruning with residual-connections and limited-data[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 1455-1464.

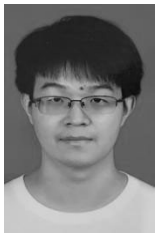
[30] ZHOU Y F, ZHANG Y, WANG Y F, et al. Accelerate CNN via recursive Bayesian pruning[C]//Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway: IEEE Press, 2019: 3305-3314.

[31] PENG H Y, WU J X, CHEN S F, et al. Collaborative channel pruning for deep networks[C]//Proceedings of the International Conference on Machine Learning. New York: ACM Press, 2019:5113-5122.

[32] YANG H R, WEN W, LI H. DeepHoyer: learning sparser neural network with differentiable scale-invariant sparsity measures[EB]. 2019.

[33] HE Y, DING Y H, LIU P, et al. Learning filter pruning criteria for deep convolutional neural networks acceleration[C]// Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 2006-2015.

[作者简介]



陈靓（1995− ），男，浙江科技学院硕士生，主要研究方向为网络剪枝。

钱亚冠（1976− ），男，博士，浙江科技学院副教授，主要研究方向为深度学习、人工智能安全、大数据处理。

何志强（1996− ），男，浙江科技学院硕士生，主要研究方向为网络剪枝。

关晓惠（1977− ），女，博士，浙江水利水电学院副教授，主要研究方向为数字图像处理与模式识别。

王滨（1978− ），男，博士，浙江大学研究员，主要研究方向为人工智能安全、物联网安全、密码学。

王星（1985− ），男，博士，浙江大学在站博士后，主要研究方向为机器学习与物联网安全。