Neural Computing and Applications (2022) 34:21119–21138
https://doi.org/10.1007/s00521-022-07594-7

ORIGINAL ARTICLE

Global balanced iterative pruning for efficient convolutional neural networks

Jingfei Chang[1] · Yang Lu[1,2,3] · Ping Xue[1] · Yiqun Xu[1] · Zhen Wei[1,2,3]

Received: 14 November 2021 / Accepted: 29 June 2022 / Published online: 27 July 2022

Abstract

With the increase of structure complexity, convolutional neural networks (CNNs) take a fair amount of computation cost. Meanwhile, existing research reveals the salient parameter redundancy in CNNs. The current pruning methods can compress CNNs with little performance drop, but when the pruning ratio increases, the accuracy loss is more serious and the compressing rates of parameters and floating-point operations (FLOPs) are unbalanced. Moreover, the existing iterative pruning methods are difficult to accurately identify and delete unimportant parameters due to the accuracy drop during pruning. We propose a novel global balanced iterative pruning method (GBIP) for CNNs. Firstly, a global equilibrium pruning strategy based on feature distribution is proposed. Then the intermediate and output features of original network are applied to guide the fine-tuning of pruned network. Moreover, we design a shallow fully-connected network to allow the output of two networks to play an adversarial game, thereby it can quickly recover the pruned accuracy among iterative pruning intervals. We conduct extensive experiments on the image classification tasks CIFAR-10, CIFAR-100, and ILSVRC-2012 to verify our pruning method can achieve efficient compression for CNNs even without accuracy loss. On the ILSVRC-2012, when removing 36.78% parameters and 45.55% FLOPs of ResNet-18, the Top-1 accuracy drop are only 0.66%. Our method is superior to some state-of-the-art pruning schemes in terms of compressing rate and accuracy. Moreover, we further demonstrate that GBIP has good generalization on the object detection task PASCAL VOC.

Keywords Deep learning · Convolutional neural network · Network pruning · Image classification · Object detection

1 Introduction

Since the emergence of deep neural networks (DNNs) [1], due to the less labeled data, poor hardware storage and computing power, it has not been able to completely release the performance. As the number of labeled datasets keeps springing up, as well as the development of high-performance hardware such as GPU and TPU, DNNs have achieved great success in the fields of scientific research and engineering. As the main component, the CNNs achieve excellent performance in extracting image features combined by virtue of the parameter sharing and translation invariance characteristics. At present, CNNs has received extensive attention in computer vision tasks such as image classification, object detection, semantic segmentation, style transfer, and super-resolution images, moreover its performance is significantly better than the traditional methods. However, as image and video tasks

✉ Yang Lu
luyang@hfut.edu.cn

✉ Zhen Wei
weizhen@gocom.cn

Jingfei Chang
cjfhfut@mail.hfut.edu.cn

Ping Xue
xueping@mail.hfut.edu.cn

Yiqun Xu
yiqunxu@mail.hfut.edu.cn

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China

[2] Anhui Mine IOT and Security Monitoring Technology Key Laboratory, Hefei 230088, China

[3] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Hefei University of Technology, Ministry of Education, Hefei 230009, China

become more and more complex, the scale and classes of CNNs are gradually increasing. Although this can achieve better accuracy, it also extends the cost of hardware and computing power for network deployment, which limits the application of high-performance CNNs on resource-constrained devices. On the other hand, some works [2–4] have shown that existing CNNs have a certain degree of parameter redundancy, which provides background support and a theoretical basis for network compression.

The existing CNNs compressing methods mainly consist of optimizing the calculation methods of convolution and designing network compression algorithms. In a nutshell, the network compressing is committed to reducing the number of parameters and FLOPs as more as possible in the case of guaranteeing network performance. Mainstream algorithms include network pruning, quantification, low-rank decomposition, and knowledge distillation. Among them, the network pruning method based on parameter importance is more convenient and effective. However, existing pruning methods of this type tend to vary widely in the compressing rates for parameters and FLOPs. Moreover, during iterative pruning, the performance loss after each pruning leads to a gradual decrease in the accuracy of later compressing operations. In response to the above issues, we consider designing a strategy in the phase of balanced pruning to efficiently optimize the continuously compressed network. In this way, it can quickly restore the accuracy to perform the iterative pruning in a few training epochs, and it can ensure that the network after pruning has almost no performance loss. The motivation of our method is twofold. Firstly, Komodakis and Zagoruyko [5] demonstrates that feature maps of the large network can pay more accurate attention to the object than the small network through extensive experiments. Secondly, Lin et al. [6] introduces GANs to optimize the network compressing, but it adds a mask for pruning, which increases the cost of the network pruning, and a separate optimization for this parameter is required.

In this paper, we propose a global balanced iterative pruning method. Firstly, we design a global balanced pruning scheme which eliminates the unnecessary parameters via analyzing the magnitude distribution of channels. Considering that simple magnitude pruning across different layers or within the same layer may lead to unbalanced pruning rates of parameters and FLOPs, we do not perform magnitude analysis across different layers. Then, we introduce an efficient performance recovery policy, which define the original network as the teacher and the pruned network as the student. And using the intermediate feature maps and the output features of the teacher to transfer the information learned by the original network to the student during fine-tuning. Moreover, we construct a shallow neural network as a platform, making the output features of the two networks conduct an adversarial game. The above strategies act on the compact network after every pruning step. And iterative pruning is carried out during the training phase. To do so, the pruned network can recover accuracy by training a few epochs after each pruning operation, which can provide more accurate guidance for the judgment of the importance of parameters in the next pruning, and shorten the whole pruning phase. The final compact network is retrained to restore the experimental accuracy.

To demonstrate the effectiveness of our GBIP, we prune VGGNet [7], ResNet [8] and GoogLeNet [9] on the image classification datasets CIFAR-10, CIFAR-100 [10] and ILSVRC-2012 [11]. Moreover, we further perform experiments on the SSD [12] on the object detection dataset PASCAL VOC [13]. The results manifest that without harming overall performance it is possible to compress and accelerate the CNNs using the proposed pruning method in this paper. On the CIFAR-10, when removing 97.22% of the parameters and 96.57% of the FLOPs of the VGG-16, the classification accuracy can still reach 90.29%. In addition, when the compression rate of SSD exceeds 50%, the performance loss is still less than 1.00%.

The proposed GBIP can be applied to many convolutional networks in image classification tasks, and it also shows good generalization in object detection. The existing network compressing method can combine with our efficient performance recovery strategy to increase the accuracy of the compressed network. What's more, because no sparseness was introduced, GBIP does not require the assistance of additional sparse matrix operations and acceleration libraries. And the entire pruning process can be achieved only by controlling one parameter, which notably reduces labor intervention and can perform automatic compression and acceleration. If adopting the larger CNN as the teacher network, the efficiency of pruning and the performance of the compressed network can be further improved.

The main contributions of our work are as follows:

- This paper proposes a global balanced pruning scheme for convolutional channels. We analyze the magnitude distribution of intermediate feature maps to eliminate the unimportant parameters and connections.
- We design an efficient performance recovery method. The abundant knowledge learned in the training process of the original network is applied to guide the compact network to quickly recover the accuracy in the iterative pruning interval.

- We demonstrate the effectiveness of the method on CIFAR-10, CIFAR-100, ILSVRC-2012 with extensive experiments. Moreover, the results on the object detection dataset PASCAL VOC further verifies our GBIP has superior generalization in CNNs compressing and accelerating. The ablation analysis manifests that the adjustment of hyperparameter can stably control the pruning rate.

2 Related works

At present, convolutional network compression has received widespread attention from both academia and industry. Many methods with significant effects such as pruning, quantification, low-rank decomposition, and knowledge distillation have emerged. The related works of our method are presented as follows:

2.1 Network pruning

Network pruning is to remove the relatively redundant weights or filters according to the importance of parameters in the CNN to compress and accelerate the network based on ensuring the accuracy of the task. The key of pruning is to determine the evaluation criteria of the importance of parameters and then design an effective pruning strategy. Some existing methods are based on the magnitude of parameters. From 2016, pruning compression for deep neural networks begins to receive wide attention from both academia and industry. In recent years, magnitude pruning, as one of the efficient methods, still exists in a large amount of works. Han et al. [2] is typical of unstructured pruning method which using the value of weight to measure the redundancy of connections and setting neurons smaller than a threshold directly to zero. Li et al. [14] deletes the filters with the smaller L1 norm. The impact of pruning different layers on the accuracy drop needs to be analyzed before each pruning. The sensitive layers will be pruned with a smaller pruning rate or directly skipped without pruning. Moreover, the specific pruning rate is manually set for all pruned layers. Polyak and Wolf [15] applies the variance of channel activation to measure their contribution. The pruning of the entire network is done sequentially on the network layers: from lower layers to the top ones, pruning is followed by fine-tuning, which is followed by pruning of the next layer. The pruning in this paper is carried out independently at each layer simultaneously. He et al. [16] prunes networks according to the L2 norm of filters. Molchanov et al. [17] uses the change of accuracy loss after deleting a parameter as the importance evaluation criterion for the parameter. The pruning is constructed as an optimization problem which is approximated using a first-order Taylor expansion. Ultimately, the parameter importance is transformed into the product of the activation and its gradient. In contrast, our method uses the maximum regularization of the L1 norm of the activation as the evaluation criterion of the parameter importance. Moreover, it uses the traditional fine-tuning method to recover the accuracy in iterative pruning intervals. Liu et al. [18] uses the scaling factor of the batch normalization layers as the evaluation standard of parameter importance. Lin et al. [19] utilizes the rank of the feature map matrix to judge how much information it contains. Li et al. [20] slims CNN through the diversity and similarity of feature maps. Tang et al. [21] fits the input complexity and feature similarity to the pruned network space to dynamically discard redundant filters. Wu et al. [22] uses the product of filter sparsity and feature dispersion to measure their importance. Chin et al. [23] learns different parameter pairs for all layers and performs affine transformation on L2 form of filters in the layer to get their importance ranking. Finally, the less important filters are removed according to the preset amount of floating point operations. Some pruning strategies are based on the impact of deleted parameters on performance drop. For example, Yu et al. [24] measures the importance of pruning neurons by minimizing the reconstruction error of the second-to-last layer in front of the final classification layer. Lee et al. [25] introduces connection sensitivity to evaluate the importance of structure, and the pruning is implemented in the parameter initialization stage before training. Guo et al. [26] samples the channel pruning as a Markov process that is optimized using standard regularization loss and model parameters or FLOPs budget regularization. You et al. [27] multiplies the intermediate feature map with a scale factor, and then estimates the accuracy loss caused by the scale factor set to zero to determine the importance of the relevant filters. Guo et al. [28] reconstructs the cropped feature and observe its impact on the classification loss to carry out layer-by-layer channel pruning. Other pruning approaches combine existing advanced algorithms to compress the network, such as [29] using reinforcement learning to search for better pruning strategies. Liu et al. [30] combines meta-learning to find compact networks with better performance. Lin et al. [31] formulates the search of optimal pruned structure as an optimization problem and integrate the ABC algorithm to solve it in an automatic manner. Ding et al. [32] generates a global network pruning strategy using long short-term memory. Compared to aforementioned works, the proposed method iteratively prunes the unneccessary channels and connections based on the importance magnitude distribution of feature maps with almost no decrease in accuracy. Moreover, our approach is able to compress the number of parameters and FLOPs in a balanced way.

2.2 Knowledge transfer

Knowledge transfer utilizes a pre-trained high-performance teacher network to guide a smaller student network, thereby improving the experimental accuracy of the small network. Ba and Caruana [33] uses the input of the final softmax layer to represent the knowledge learned by the teacher network to supervise the training of the student network. Hinton et al. [34] introduces temperature T to the output of the softmax layer and then trains together as a soft label with the real target. The two methods above only consider the information contained in the output, which is relatively limited. The FitNet proposed in [35] applies not only the output of the teacher network but also its intermediate feature to jointly optimize the training process of the student network. This can train a deep and narrow student network while enhancing its generalization ability. Komodakis and Zagoruyko [5] proposes attention transfer that using the attention maps in the teacher network to deliver the information of the teacher network's attention to the student network and improve the performance. Different from the aforementioned methods, we mainly introduce the information representation of feature maps and output features from the original network to guide the pruned network to quickly eliminate the accuracy loss, thus making each pruning operation of iterative compressing more accurate.

3 Proposed method

3.1 The global balanced iterative pruning framework

Figure 1 shows the overall framework of our pruning method. Firstly, the labeled training images are input into the original network. We analyze the magnitude distribution of feature maps in each pruning layer and remove the unnecessary channels. Then we select three-pair feature maps and the output from original network and pruned one to transfer attention and knowledge. It can be seen from the figure that for the same input sample, the attention maps generated by the pruned student network have obviously weaker interest in the classification object than the original teacher network. Afterwards, a shallow neural network is conduct to make the two output features play a game, which further improves the accuracy of the pruned network. Finally, after few epochs for accuracy recovery the next pruning phase is performed. In this scenario, we achieve iterative pruning to compress and accelerate the original convolutional network.

3.2 Global balanced pruning strategy

Given a convolutional neural network with L layers, we refer to $C = (C_1, C_2, \ldots, C_L)$ as the original network structure,

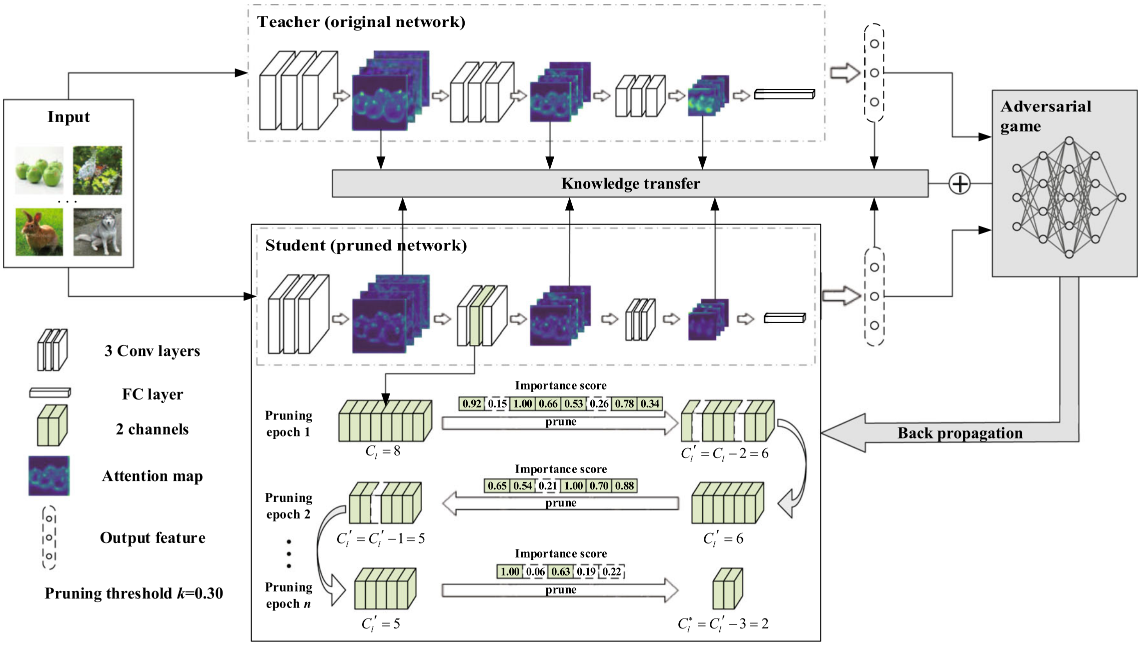


Fig. 1 The global balanced iterative pruning (GBIP) framework. (This figure is best viewed in color and zoomed in) (Color figure online)

where C_l is the number of channels in the lth layer. $W \in \mathbb{R}^{C_{\mathrm{out}} \times C_{\mathrm{in}} \times K \times K}$ is the weight of the filter, where C_{out} is the number of output channels, C_{in} is the number of input channels, and $K \times K$ is the size of the filter. The feature map generated in the lth layer is $M \in \mathbb{R}^{C_l \times w \times h}$, where w and h are width and height of the feature map. With the input sample x, the output produced by the original network is defined as $f_{\mathrm{T}}(x, W_{\mathrm{T}})$, and the output of the pruned network is defined as $f_{\mathrm{S}}(x, W_{\mathrm{S}})$. $G(f(x, W), W_{\mathrm{G}})$ is the adversarial platform, where $f(x, W)$ is the output of the teacher or the student network.

Here, we propose a global balanced pruning strategy of the convolutional neural network. To reduce the complexity of the network pruning, an effective pruning method based on the magnitude of parameters should be designed. Filters with smaller weights tends to produce relatively weak activation feature maps compared to other filters at the same layer. Therefore, we can preferentially remove such filters to reduce network redundancy during network pruning. Most of the existing methods delete unimportant parameters directly based on the L1, L2 norm, or other magnitude of the filters and feature maps. However, this depends on a relatively uniform distribution of feature map's magnitude. Otherwise, when the pruning threshold is unreasonable, it will cause enormous differences in the pruning rate of layers, which will seriously affect the final performance of the network. Here, we analyze the L1 norm for the feature maps of VGG-16 and Resnet-56 on the CIFAR10 and Resnet-18 on the ILSVRC-2012. Specifically, we first calculate all the L1 norm of the feature maps in the layers to be pruned and then perform maximum regularization on the feature maps in each layer using Eq. 1 to obtain the importance score m_l^c of each feature map.

$$m_l^c = \|M_l^c\|_1 / \max\left\{\|M_l^1\|_1, \|M_l^2\|_1, \ldots, \|M_l^{C_l}\|_1\right\} \tag{1}$$

where $c \in \{1, 2, \ldots, C_l\}$ is the index of the every feature map in the lth layer. $\|\boldsymbol{\cdot}\|_1$ refers to the L1 norm. We visualize the results obtained as Fig. 2.

It can be seen from the figure that for CIFAR-10, the importance scores of VGG-16 are generally concentrated between 0 and 0.5. While the importance distribution of Resnet-56 is relatively uniform, but the importance scores in the first few layers are almost between 0 and 0.5. On ILSVRC-2012, the importance of the features for Resnet-18 at each layer is significantly different. The importance scores of the sixth layer are almost between 0 and 0.4, while those of the eighth layer mainly vary from 0.4 to 1. Therefore, directly setting the threshold based on the L1 norm cannot achieve ideal compression for all layers in the network. If the layer-wise pruning rate is preset, when the compressing rate of all layers is kept the same, the final pruning rates of parameters and FLOPs can indeed be exactly equal. However, when the compressing rate of each layer is immense, the performance of the pruned network is also poor. When different pruning rates are preset for different layers, firstly, a balanced compression rate is not always obtained, i.e., the difference between the pruning rates of parameters and FLOPs is large. The two pruning rates of some methods can even differ by nearly 40%. Therefore, we set pruning factor k to perform on the mean value of the importance scores of the feature maps to determine the final pruning threshold m_l^p of the lth layer.

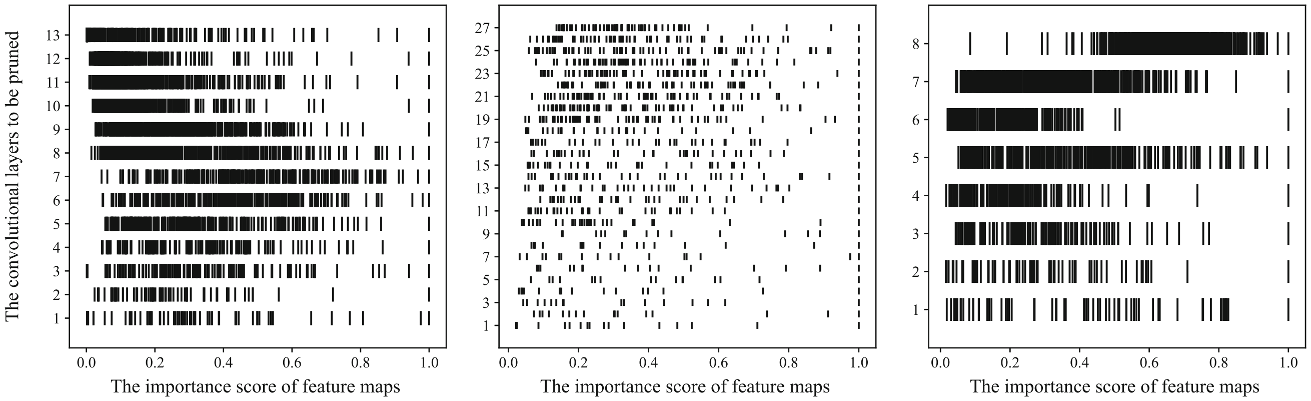
$$m_l^p = k \cdot \frac{1}{C_l} \sum_{c=1}^{C_l} m_l^c \tag{2}$$

where $k \in (0, 1)$ is the pruning threshold factor used to control the network pruning rate, and it is also the only variable parameter in our proposed method. The number of parameters and FLOPs of CNNs are calculated as follows:

$$\mathrm{Params} = C_{\mathrm{out}} \times (C_{\mathrm{in}} \times K \times K + 1) \tag{3}$$

$$\mathrm{FLOPs} = 2 \times w \times h \times C_{\mathrm{out}} \times (C_{\mathrm{in}} \times K \times K + 1) \tag{4}$$

Pruning channels is equivalent to reducing the number of C_{out}. The dimension $w \times h$ of the feature maps in front



(a) VGG-16 on CIFAR-10 **(b)** Resnet-56 on CIFAR-10 **(c)** Resnet-18 on ILSVRC-2012

Fig. 2 The importance score distribution density of the feature map in the pruned layers

layers is larger than that in last layers. If the channels pruning rate of the front layers is considerably larger, the compression ratio of the FLOPs can be much higher than that of the parameters. Using the method in this paper, channels in each layer can achieve relatively balanced compressing. In this case, the clipping rate of parameters and FLOPs will be closer, which is more conducive to the balanced compression and acceleration of the convolutional network. Moreover, the compressing amplitude of each layer can be adaptively adjusted via k to achieve global equilibrium pruning under different degrees. Redundant parameters in the CNNs can be deleted at each pruning step via the above pruning strategy, and the problem of unbalanced compressing among layers will not occur. After pruning, the performance of the network will not suffer a great loss.

3.3 Performance recovery scheme

3.3.1 Knowledge transfer

The intermediate feature map of CNNs is the concrete or abstract representation extracted by the filters from the input images, which shows the objects that the network pays attention to when treating specific tasks. For image classification, the feature maps will highlight the target to be classified and weaken the background and irrelevant objects to obtain a more reliable classification result. Therefore, whether the feature map can precisely pay attention to the goal and how strong the attention is are especially important to the performance of the network. It can be seen from Fig. 1 that the feature maps of the pruned network pay less attention to the target, which will seriously affect the correctness of pruning and the accuracy of the compressed network. Because of this, we introduce knowledge transfer in the pruning process.

We select three layers with the different dimensions of feature maps and integrate the feature maps in the same layer to form an attention map to guide pruned networks to focus on classification objectives. Specifically, for the C_l intermediate feature maps of the lth layer, the attention map is constructed using the Eq. 5:

$$A^l\left(M^{ab}\right) = \frac{1}{C_l}\sum_{i=1}^{C_l}\left(M_i^{ab}\right)^2 \tag{5}$$

where M^{ab} is the pixel of the attention map in the lth layer with $a \in \{0, 1, \ldots, w-1\}$ and $b \in \{0, 1, \ldots, h-1\}$. The attention map produced in this way can get the attention area of the input sample, and on the other hand, it can also represent the amount of information learned in the layer. Then as shown in Eq. 6, after regularizing the three pairs of attention maps of two networks, we use the L2 norm of their difference to construct the attention transfer loss $\mathcal{L}_{\mathrm{AT}}$ of the student network.

$$\mathcal{L}_{\mathrm{AT}} = \sum_{l=1}^{3}\left\|\frac{A_S^l}{\left\|A_S^l\right\|_2} - \frac{A_T^l}{\left\|A_T^l\right\|_2}\right\|_2 \tag{6}$$

where A_S^l is the attention map in the lth layer of the student network, and A_T^l is the corresponding attention map of the teacher network. The interest of the student can be made as close as possible to that of the teacher in the inference process through the $\mathcal{L}_{\mathrm{AT}}$ loss. In the experiments, we prune VGGNet, ResNet, and GoogLeNet. The specific implementation positions for extracting attention maps in these three networks are plotted in Fig. 3.

When dealing with image classification tasks, the output of the convolutional network is the probabilities of each category, so we apply the output features of the original network to guide the compressed network. In this way, we can perform more accurate pruning and improve the performance of the pruned network at the same time. Regarding the outputs of the teacher and the student network $f_{\mathrm{T}}(x)$ and $f_{\mathrm{S}}(x)$, this paper introduces temperature T_{emp} which draws on the idea of [34] to smooth the two outputs as shown in Eqs. 7 and 8. Hence the classification probability of the student for each category can be as similar to the teacher network as possible to avoid the probability of all incorrect classification tends to zero, and result in a smoother category probability distribution.

$$p(x) = F_{\mathrm{soft\,max}}\left(f_{\mathrm{S}}(x)/T_{\mathrm{emp}}\right) \tag{7}$$

$$q(x) = F_{\mathrm{soft\,max}}\left(f_{\mathrm{T}}(x)/T_{\mathrm{emp}}\right) \tag{8}$$

where $F_{\mathrm{soft\,max}}(\centerdot)$ refers to softmax function. Then the KL divergence of $p(x)$ and $q(x)$ is calculated according to Eq. 9, and the accuracy of the student can be improved by reducing the divergence during training.

$$D_{KL}(p \parallel q) = \sum_{i=1}^{n}\left[p(x)\log(p(x)) - p(x)\log(q(x))\right] \tag{9}$$

At the same time, to better correct the output of the student network, the cross-entropy loss $L_{\mathrm{CE}}(W_{\mathrm{S}})$ between the output features of the student and the real labels is added to the above divergence, which is regarded as the output features transfer loss $L_{\mathrm{OT}}(W_{\mathrm{S}})$.

$$\mathcal{L}_{\mathrm{OT}}(W_{\mathrm{S}}) = \alpha \cdot \mathcal{L}_{\mathrm{KL}}(W_{\mathrm{S}}) + (1-\alpha)\mathcal{L}_{\mathrm{CE}}(W_{\mathrm{S}}) \tag{10}$$

where α is the weight between the two losses of KL divergence and cross entropy. And $L_{\mathrm{KL}}(W_{\mathrm{S}})$ is formulated in Eq. 11. In order to make the effect of these two losses roughly under the same magnitude, we multiply D_{KL} by T_{emp}^2.

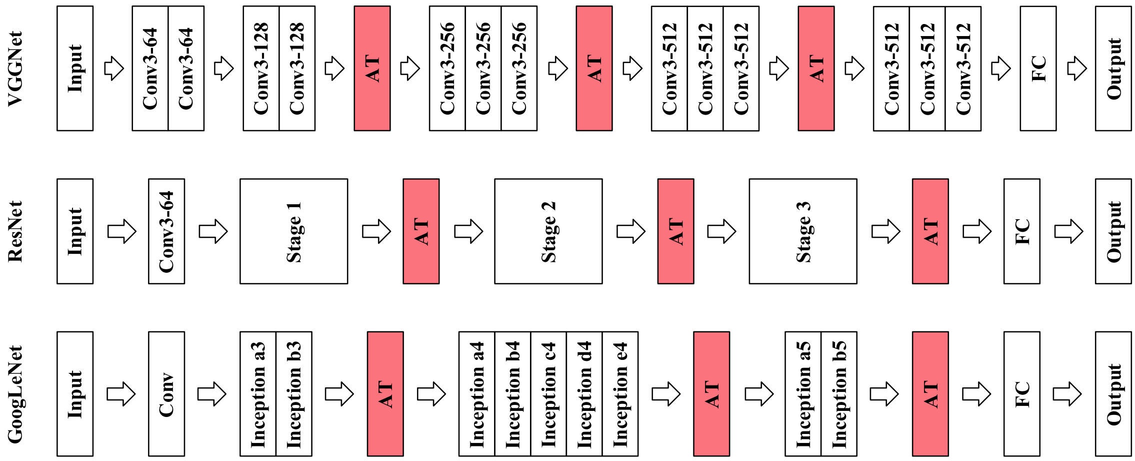


Fig. 3 The positions of VGGNet, ResNet, and GoogLeNet for extracting attention maps

$$\mathcal{L}_{\mathrm{KL}}(W_{\mathrm{S}}) = T_{\mathrm{emp}}^{2} \cdot D_{\mathrm{KL}}(p \parallel q) \tag{11}$$

Different from the existing knowledge transfer method with fixed structure of student network, we continue to compress the network. In fact, knowledge transfer is only an auxiliary strategy of iterative pruning in our method, which aims to quickly mitigate the accuracy loss from each pruning step. To fully accelerate the optimizing for pruned network from training process and results, we apply the intermediate feature maps and the output simultaneously to guide the fine-tuning of the compact network. Through the above methods, the intermediate and the final classification information learned by the original network can be thoroughly transmitted to the pruned student network. In this scenario, we can not only ensure that the student accurately finds the unimportant parameters for the corresponding task but also restore the network's performance through a few training epochs after each pruning step to achieve iterative pruning during the training process.

3.3.2 Adversarial game

The above-mentioned knowledge transfer has achieved the effective delivery of semantic information from the unpruned network to the pruned network. On this basis, we found that introducing an adversarial game strategy can further improve the final output performance of the student and the recovery speed of accuracy in the iterative pruning. Hence, this paper constructs a shallow neural network as the adversarial platform and makes the outputs of the student and the teacher network play an adversarial game on it. In this way, the output of the student network may closer approach that of the teacher network. Therefore, the adversarial game loss of the student network is defined as follows:

$$L_{\mathrm{AG}}(W_{\mathrm{S}}) = E_{f_{\mathrm{S}}(x) \sim p_{\mathrm{S}}(x)}[\log(1 - G(f_{\mathrm{S}}(x, W_{\mathrm{S}}), W_{\mathrm{G}}))], \tag{12}$$

where $p_{\mathrm{S}}(x)$ represents the feature distribution of the student network. Combined with the knowledge transfer loss in the previous section, the training loss of the student network in the proposed pruning method consists of the following three parts:

$$\mathcal{L}_{\mathrm{S}}(W_{\mathrm{S}}) = \mathcal{L}_{\mathrm{AG}}(W_{\mathrm{S}}) + \mathcal{L}_{\mathrm{AT}}(W_{\mathrm{S}}) + \mathcal{L}_{\mathrm{OT}}(W_{\mathrm{S}}). \tag{13}$$

The network for adversarial game needs to be continuously trained to distinguish whether the input is from the teacher or the pruned network. For the output features from the teacher network, it should produce a positive response, while for the output features generated by the student network, the network should treat it as the pseudo sample. To be specified, the loss of the adversarial platform during training is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{G}}(W_{\mathrm{G}}) = {} & E_{f_{\mathrm{T}}(x) \sim p_{\mathrm{T}}(x)}[\log(1 - G(f_{\mathrm{T}}(x, W_{\mathrm{T}}), W_{\mathrm{G}}))] \\ & + E_{f_{\mathrm{S}}(x) \sim p_{\mathrm{S}}(x)}[\log(G(f_{\mathrm{S}}(x, W_{\mathrm{S}}), W_{\mathrm{G}}))],\end{aligned} \tag{14}$$

where $p_{\mathrm{T}}(x)$ represents the feature distribution of the teacher network. The adversarial platform and the pruned network are alternately optimized in each training epoch to accelerate the performance improvement of the student network. In addition, we integrate the knowledge transfer, so that the accuracy of the compact network can be regained only after a few training epochs and then the iterative pruning will be conducted. Accordingly, the entire network pruning process becomes more compact and accurate. Moreover, our method can significantly improve the accuracy of the network after pruning. Extensive experiments have shown that even in the case of a considerable compressing rate, the performance of the pruned network after retraining can still reach or exceed that of the original network. The subsequent experiments in this paper also entirely demonstrate the effectiveness and accuracy of our channel pruning method in network compressing and accelerating. Algorithm 1 shows the pseudocode of the global balanced iterative pruning method. Given a pretrained original convolutional network, a compact model S^*_{pruned} can be obtained after pruning with the GBIP scheme. Finally, we retrain the pruned model from scratch to restore the accuracy of the experiment.

```
**Algorithm 1** Global Balanced Iterative Pruning Algorithm (GBIP)
Input: Training set $X_{train} = \{x_1, x_2, \cdots, x_n\}$ with $n$ samples, original model $T_{model}$, pruned model $S_{model} = \{C_1, C_2, \cdots, C_L\}$ with weight $W_S$, adversarial platform $G_{model}$ with weight $W_G$, learning rate $\eta$, Num epochs $N$, epochs of pruning intervals $s_p$, pruning threshold factor $k$
Output: Pruned compact structure $S^*_{pruned} = \{C^*_1, C^*_2, \cdots, C^*_L\}$ with weight $W^*_S$
Initialize $S_{model}$ with pretrained weight $W_S$;
for $epoch = 0$ to $N$ do
    < Prune the $S_{model}$ >
    if $epoch \% s_p == 0$ then
        for $l = 1$ to $L$ do
            $j = 0$;
            for $c = 1$ to $C_l$ do
                Calculate the important score $m^c_l$ via Eq.1;
                Calculate the pruning threshold $m^p_l$ of the $l$-th layer via Eq.2;
                if $m^c_l < m^p_l$ then
                    Prune the $c$-th channel and the filters related;
                    $j = j + 1$;
                end if
            end for
            $C^{'}_l = C_l - j$;
        end for
        Pruned student structure $S^{'}_{pruned} = \{C^{'}_1, C^{'}_2, \cdots, C^{'}_L\}$;
    end if
    $S_{model} = S^{'}_{pruned}$;
    < Fix $S_{model}$ and update $G_{model}$ >
    Sample output $f_S(x)$ of $S_{model}$;
    Sample output $f_T(x)$ of $T_{model}$;
    Update weight $W_G$ of adversarial platform $G_{model}$ via Eq.14;
    < Fix $G_{model}$ and update $S_{model}$ >
    Update weight $W_S$ of student model $S_{model}$ via Eq.13;
end for
$S^*_{pruned} = S_{model}$.
```

For most of the existing network pruning methods, the compressed network inherits the weights and bias from the original network to restore the performance as much as possible through fine-tuning. However, when the network pruning rate is remarkable, the accuracy recovery after fine-tuning is not obvious, and the actual performance of the compact network cannot be greatly manifested. Liu et al. [36] makes a surprising observation in structured network pruning that fine-tuning a pruned model only gives comparable or worse performance than training that model with randomly initialized weights. And the experiment results reveal that the pruned architecture itself, rather than a set of inherited important weights, is more crucial to the efficiency in the final model. Our results of pruning VGG-16 on the CIFAR-10 further verify the observation. In order to fully demonstrate the performance of the compact network, we retrain the pruned network from scratch via the performance recovery method in our experiments. Specifically, we keep the number of FLOPs consistent before and after pruning. The number of training epochs of the original network is multiplied by the accelerating rate of FLOPs as the retraining epochs of the compressed network. Finally, we compare the accuracy of the pruned network with the original network to draw a conclusion.

3.4 Pruning strategy for different CNNs

Since different CNNs have different network structures, the specific pruning implementation details should also change accordingly. We perform experiments on VGGNet, ResNet, and GoogLeNet. Among them, VGGNet is a common layer-by-layer convolutional network and does not include unusual architecture. Therefore, all layers can be directly pruned without affecting the integrity of the final network structure. ResNet contains customized residual modules, so arbitrarily compressing each layer will destroy the dimension matching of the channels. The basic residual block is composed of two convolutional layers. We only discard the output channels in the first layer, and the input channels in the second layer will also change accordingly. By doing so, the overall dimension of the ResNet is still matched and can be trained correctly after pruning. GoogLeNet is a more complex convolutional network with multiple Inception V3 modules, each of which contains four branches. We cut the branches containing two and three convolutional layers to conduct the compressing and accelerating. The specific structure and pruning scheme of the Inception V3 module is plotted in Fig. 4.

4 Experiments

We demonstrate the effectiveness of the proposed method by pruning VGGNet, ResNet, and GoogLeNet on the CIFAR-10, CIFAR-100, and ILSVRC-2012. Moreover, we compress SSD via GBIP on the PASCAL VOC to analyze its generalization on object detection. All experiments are implemented with Pytorch on NVIDIA TITAN X GPUs. For fairly comparing with the existing pruning methods, the network pre-training and parameter settings use the method presented in [8]. Specifically, the pre-training epochs of CNNs on the CIFAR are 160, while on the ILSVRC-2012 are 90. The learning rate is initially set to 0.1 and then decreased by a factor of 10 on half and three-quarter epochs. Stochastic gradient descent (SGD) with momentum is used for backpropagation, and the momentum is 0.9 with a weight decay of 1e−4. In the retraining stage, we adjust the learning rate with the cosine annealing adjustment strategy. The parameter settings during the

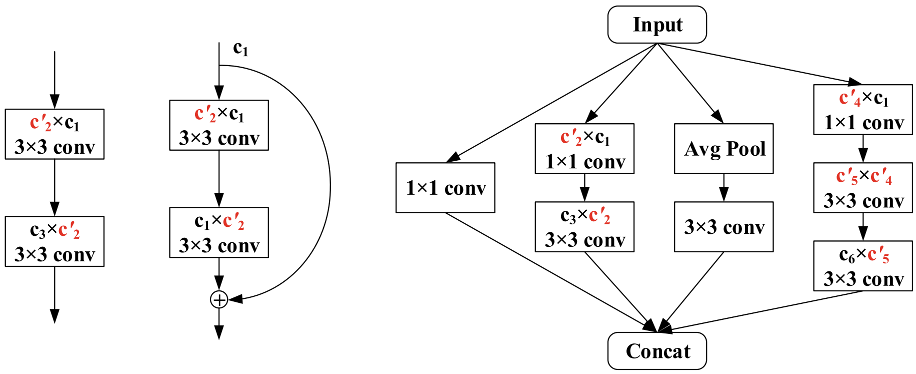


Fig. 4 Illustration of pruning VGGNet, ResNet and GoogLeNet. The black font indicates the number of original channels, and the red font indicates that after pruning (Color figure online)

iterative pruning are as follows. The weight in output transfer is $\alpha = 0.3$. On the CIFAR, the total epochs of training are $N = 30$, and the pruning interval period is $s_p = 10$. On the ILSVRC-2012, the training epochs for pruning are $N = 20$, and the pruning interval is $s_p = 10$. The pruning threshold factor k is the only parameter that is changed for pruning. In addition, we draw on the neural network composed of three fully-connected layers with the neurons of 128-256-128 in [6] as the adversarial platform in the adversarial game.

In this section, we compare the proposed method with the existing pruning schemes, among which Li et al. [14], SFP [16], DCP [37], FPGM [38], EDP [39], CNN-FCF [40], CCP [41], Taylor-FO-BN [42], HRank [19], ManiDP [21], NPPM [43] are the state-of-the-art methods. Due to the difference in experimental equipment and environment, the results obtained by different papers also have several differences. In order to make a fair comparison as much as possible, we also mainly compare the decrease of accuracy after pruning according to current methods. The results of these competing methods are reported according to the original article.

4.1 Results comparison on CIFAR-10

We first prune VGG-16 on the CIFAR-10, and the results are shown in Table 1. It can be seen from the table that when $k = 0.3$, our method reduces up to 47.86% of the parameters and 44.39% of the FLOPs for VGG-16, however, the accuracy of the network is even improved by 0.54% compared with the baseline. When the network compression ratio exceeds 80%, the compact network still has a performance improvement of 0.17%. Although the parameter compression ratio of ABCPruner [31] is 5.65% higher than that of our method, the pruning rate of FLOPs is lower than that of this paper and the final accuracy after pruning is also smaller (− 0.06% vs. − 0.17%). As the VGG-16 continues to be compressed, the accuracy of the network is gradually declining. When discarding 97.22% of the parameters and 96.57% of the FLOPs with $k = 0.7$, the accuracy of the final network still reaches 90.29%.

The experimental results show that for the CIFAR-10, the VGG-16 does have a certain degree of parameter redundancy. Compressing the network can reduce the impact of overfitting and improve the accuracy of the network. At the same time, the effectiveness of the pruning method proposed in this paper is preliminarily verified. Then, we continue to cut ResNet-56, and the experimental results are tabulated in Table 2. When $k = 0.4$, the parameters and FLOPs of ResNet-56 are reduced by 41.18% and 47.81%, respectively. At this time, the accuracy of the network after pruning is increased by 0.67%. And when $k = 0.5$, the pruning rate has exceeded 50.00%, but the network still has a performance improvement of 0.36%, which is significantly better than the compared algorithms. Although the final accuracy improvement of SRR-GR [49] is 0.01% higher than that of ours, the compression rate of its FLOPs is relatively low by 9.55% (53.80% vs. 63.35%). The accuracy of ResNet-56 only drops by 0.38% when deleting 70.37% of the parameters and 73.41% of the FLOPs. In this case, the network parameters are only 0.21M. In addition, it can be found that when $k = 0.4$, the classification accuracy of ResNet-56 is 94.09%, which is 0.32% higher than that of VGG-16 when $k = 0.5$, however, the parameters are only about 1/5 of VGG-16. It also confirms from the side that the residual module can effectively improve the performance of CNNs in image classification tasks.

Then, we compress ResNet-110. From Table 3, it can be concluded that the baseline accuracy of ResNet-110 on the CIFAR-10 is 93.53%. When 20.81% of the parameters and 22.95% of FLOPs are discarded, the accuracy increased by 0.95%. HRank [19] compresses parameters and FLOPs by 41.20% and 39.40%, respectively, which is about 20% lower than our method when $k = 0.5$, and the performance drop is also 0.02% worse. The performance of the compressed network is still improved by 0.52% compared to the original network

Table 1 Performance comparison of VGG-16 on CIFAR-10

Method	Base Acc/%	Pruned Acc/%	Acc.drop/%	Parameters/M	Parameters.drop/%	FLOPs/M	FLOPs.drop/%
Baseline	93.60	–	–	14.73	–	314.59	–
GBIP (k = 0.3)	**93.60**	**94.14**	**− 0.54**	**7.68**	**47.86**	**174.94**	**44.39**
EPFS-F-0.001 [3]	93.50	93.61	− 0.11	6.49	56.70	206.00	34.30
Li et al. [14]	93.25	93.40	− 0.15	5.40	64.00	206.00	34.20
Liu et al. [36]	93.63	93.78	− 0.15	5.40	64.00	206.00	34.20
Zhao et al. [44]	93.25	93.18	0.07	3.92	73.34	190.00	39.10
GAL-0.05 [6]	93.96	93.77	0.19	3.36	77.60	189.49	39.60
GAL-0.1 [6]	93.96	93.42	0.54	2.67	82.20	171.89	45.20
ABCPruner [31]	93.02	93.08	− 0.06	1.67	88.68	82.81	73.68
GBIP (k = 0.5)	**93.60**	**93.77**	**− 0.17**	**2.50**	**83.03**	**60.53**	**80.76**
GBIP (k = 0.7)	**93.60**	**90.29**	**3.31**	**0.41**	**97.22**	**10.79**	**96.57**

Acc.drop is the accuracy drop of the pruned network, so a negative number means the compressed model has better performance than the baseline. A smaller number of Acc.drop is better. Parameters.drop and FLOPs.drop are the pruned percentage of the parameters and FLOPs, respectively

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

even when 71.10% of the parameters and 72.82% of the FLOPs are eliminated. At this time, the network scale is similar to that of CNN-FCF [40], but the final accuracy loss is 1.14% lower (– 0.52 vs. 0.62). This experiment shows that ResNet-110 has obvious parameter redundancy on the CIFAR-10, which leads to overfitting during the training process, resulting in lower accuracy of the original network. And our global balanced iterative pruning method can achieve better accuracy recovery in the case of accurately compressing the ResNet-110. It also manifests that the compressing rate of parameters and FLOPs and the accuracy drop using our pruning method are significantly better than all comparative methods.

In order to further demonstrate the applicability of GBIP to various convolutional networks, we continue to prune GoogLeNet. The experimental results are depicted in Table 4. Due to the Inception module, GoogLeNet increases the width, therefore the baseline accuracy on the CIFAR-10 reaches 94.72%, which is ahead of VGGNet and ResNet. When $k = 0.4$, after deleting 33.87% of the parameters and 37.95% of the FLOPs, the accuracy of the network increased by 0.52%. When the parameters and FLOPs are compressed to about 50%, the performance of the compact network is increased by 0.41%. Even if the parameters and the FLOPs are removed by 65.64% and 69.34% respectively, the classification accuracy of the network is still improved by 0.34%. It attests that GoogLeNet is also redundant on the CIFAR-10. Using the iterative pruning method in this paper can effectively eliminate unimportant parameters and improve the experimental performance of GoogLeNet.

4.2 Results comparison on CIFAR-100

We continue to prune VGG-19 and ResNet-56 on the CIFAR-100, and the experimental results are reported in Tables 5 and 6, respectively. CIFAR-100 has the same total number of training and test images as CIFAR-10, but the category has increased from 10 to 100. As the training data for each class of images decreases, the performance of the convolutional neural network also drops significantly. It can be seen from Table 5 that the baseline accuracy of VGG-19 on CIFAR-100 is 73.58%. When pruning 75.66% of the parameters and 68.54% of the FLOPs, the accuracy of the retrained compact network is increased by 0.48%. Even when $k = 0.5$, the performance is only declined by 1.76% when parameters and FLOPs are compressed by 85.17% and 89.82%, respectively. And it is significantly better than Slimming [18], Liu et al. [36] and GReg-2 [51] in terms of network compression ratio and performance recovery. This manifests that our GBIP is also applicable to datasets with relatively few training samples. Table 6 shows that the baseline accuracy of ResNet-56 is 71.36%. Because parameters and FLOPs of ResNet-56 are significantly less than VGG-19, the redundancy of ResNet-56 is also smaller. However, when the number of FLOPs is discarded by 48.71%, there is still a 0.52% improvement in performance. When $k = 0.5$, we remove 68.27% of the FLOPs with 0.18% accuracy drop that is still significantly superior to the comparison method.

Table 2 Performance comparison of ResNet-56 on CIFAR-10

Method	Baseline Acc/%	Pruned Acc/%	Acc.drop/%	Parameters.drop/%	FLOPs.drop/%
SFP [16]	93.59	93.89	− 0.30	–	14.70
Li et al. [14]	93.04	93.06	− 0.02	13.70	27.60
Liu et al. [36]	93.14	93.05	0.09	13.70	27.60
LSTM-SEP-2 [45]	93.04	93.85	− 0.81	27.91	47.52
HRank [19]	93.26	93.52	− 0.26	29.30	16.80
GBIP (k = 0.4)	**93.42**	**94.09**	**− 0.67**	**41.18**	**47.81**
SFP [16]	93.59	93.78	− 0.19	–	41.10
HRank [19]	93.26	93.17	0.09	50.00	42.40
CNN-FCF [40]	93.14	93.38	− 0.24	43.09	42.78
NISP [24]	–	–	0.03	42.60	43.61
He et al. [46]	93.59	93.72	− 0.13	–	47.10
He et al. [47]	92.80	91.80	1.00	–	50.00
AMC [29]	92.80	91.90	0.90	–	50.00
DCP [37]	93.80	93.59	0.21	–	50.00
DMC [48]	93.62	93.69	− 0.07	–	50.00
NPPM [43]	93.04	93.40	− 0.36	–	50.00
SFP [16]	93.59	93.35	0.24	–	52.60
FPGM [38]	93.59	92.89	0.70	–	52.60
CCP [41]	93.50	93.42	0.08	–	52.60
He et al. [46]	93.59	93.34	0.25	–	52.90
LEGR [23]	93.90	93.70	0.20	–	53.00
SRR-GR [49]	93.38	93.75	− 0.37	–	53.80
ABCPruner [31]	93.26	93.23	0.03	54.20	54.13
RL-MCTS [50]	93.20	93.56	− 0.36	–	55.00
EDP [39]	93.61	93.61	0	54.18	57.71
GReg-2 [51]	93.36	93.36	0	–	60.78
ManiDP [21]	93.70	93.64	0.06	–	62.40
GBIP (k = 0.5)	**93.42**	**93.78**	**− 0.36**	**55.24**	**63.35**
EPFS-C-0.6-0.05 [3]	93.26	92.53	0.73	67.10	55.00
FALF [52]	93.09	93.05	0.04	–	67.62
DAIS [53]	94.53	93.53	1.00	–	70.90
GBIP (k = 0.6)	**93.42**	**93.04**	**0.38**	**70.37**	**73.41**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

Experiments on the CIFAR datasets preliminarily verify the effectiveness and superior performance of the proposed method in image classification tasks. Our GBIP can achieve a certain degree of compression for the parameters and FLOPs of VGGNet, ResNet, and GoogLeNet almost without accuracy drop. It also fully indicates that in different tasks, the existing CNNs have certain parameter redundancy, and removing these unimportant parameters can achieve network compression and acceleration without affecting the performance of networks. In this way, the computational cost of the neural network will reduce remarkably.

4.3 Results comparison on ILSVRC-2012

To further assess the effectiveness of the proposed pruning method, we experiment on the large image classification dataset ILSVRC-2012 with 1000 categories which are difficult to precisely classify, and the parameters of the CNNs are less redundant, so pruning is more challenging. In this subsection, we select ResNet-18 and ResNet-50 for pruning, which can highlight the power of our method. The original Top-1 and Top-5 accuracy of ResNet-18 on ILSVRC-2012 are 70.02% and 89.23%. As we can see from Table 7 that when pruning less than 46.00% of the FLOPs via GBIP, the Top-1 and Top-5

Table 3 Performance comparison of ResNet-110 on CIFAR-10

Method	Baseline Acc/%	Pruned Acc/%	Acc.drop/%	Parameters/M	Parameters.drop/%	FLOPs/M	FLOPs.drop/%
Baseline	93.53	–	–	1.73	–	256.04	–
SFP [16]	93.68	93.83	− 0.15	–	–	216.00	14.60
Li et al. [14]	93.53	93.55	− 0.02	1.68	2.30	213.00	15.90
Liu et al. [36]	93.14	93.22	− 0.08	1.68	2.30	213.00	15.90
GBIP (*k* = 0.3)	**93.53**	**94.48**	**− 0.95**	**1.37**	**20.81**	**197.29**	**22.95**
SFP [16]	93.68	93.93	− 0.25	–	–	182.00	28.20
Li et al. [14]	93.53	93.30	0.20	1.16	32.40	155.00	38.60
Liu et al. [36]	93.14	93.60	− 0.46	1.16	32.40	155.00	38.60
HRank [19]	93.50	94.23	− 0.73	–	41.20	–	39.40
SFP [16]	93.68	93.86	− 0.18	–	–	150.00	40.80
CNN-FCF [40]	93.58	93.67	− 0.09	–	43.19	–	43.08
NISP [24]	–	–	0.18	–	43.25	–	43.78
GAL [6]	93.50	92.74	0.76	0.95	44.80	130.20	48.50
FPGM [38]	93.68	93.73	− 0.05	–	–	121.00	52.30
He et al. [46]	93.68	93.79	− 0.11	–	–	101.00	60.30
GBIP (*k* = 0.5)	**93.53**	**94.28**	**− 0.75**	**0.69**	**60.12**	**96.48**	**62.32**
ABCPruner [31]	93.50	93.58	− 0.08	0.56	67.41	89.87	65.04
HRank [19]	93.50	92.65	0.85	–	68.60	–	68.70
CNN-FCF [40]	93.58	92.96	0.62	–	69.51	–	70.81
GBIP (*k* = 0.7)	**93.53**	**94.05**	**− 0.52**	**0.50**	**71.10**	**69.58**	**72.82**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

Table 4 Performance comparison of GoogLeNet on CIFAR-10

Method	Baseline Acc/%	Pruned Acc/%	Acc.drop/%	Parameters/M	Parameters.drop/%	FLOPs/G	FLOPs.drop/%
Baseline	94.72	–	–	6.17	–	1.53	–
GBIP (*k* = 0.4)	**94.72**	**95.25**	**− 0.52**	**4.08**	**33.87**	**0.95**	**37.95**
GBIP (*k* = 0.5)	**94.72**	**95.13**	**− 0.41**	**3.19**	**48.30**	**0.74**	**52.04**
GAL-0.5 [6]	95.05	94.56	0.49	3.12	49.30	0.94	38.20
ABCPruner [31]	95.05	94.84	0.21	2.46	60.14	0.51	66.56
GBIP (*k* = 0.7)	**94.72**	**95.06**	**− 0.34**	**2.12**	**65.64**	**0.47**	**69.34**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

Table 5 Performance comparison of VGG-19 on CIFAR-100

Method	Baseline Acc/%	Pruned Acc/%	Acc.drop/%	Parameters/M	Parameters.drop/%	FLOPs/M	FLOPs.drop/%
Baseline	73.58	–	–	20.09	–	399.52	–
Slimming [18]	73.26	73.48	− 0.22	5.00	75.10	251.00	37.10
Liu et al. [36]	72.63	73.08	− 0.45	5.00	75.10	251.00	37.10
GBIP (*k* = 0.4)	**73.58**	**74.06**	**− 0.48**	**4.89**	**75.66**	**125.67**	**68.54**
GReg-2 [51]	74.02	67.75	6.27	–	–	–	88.69
GBIP (*k* = 0.5)	**73.58**	**71.82**	**1.76**	**2.98**	**85.17**	**40.67**	**89.82**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

accuracy loss is smaller than that of other methods. Although the parameter compressing rate of ABCPruner [31] is 6.77% higher than that of ours, and the FLOPs pruning rate is 0.67% lower, the performance drop after

Table 6 Performance comparison of ResNet-56 on CIFAR-100

Method	Baseline Acc/%	Pruned Acc/%	Acc.drop/%	FLOPs/M	FLOPs.drop/%
Baseline	71.36	–	–	127.09	–
GBIP (*k* = 0.3)	**71.36**	**73.57**	**− 2.21**	**92.35**	**27.33**
GBIP (*k* = 0.4)	**71.36**	**71.88**	**− 0.52**	**65.18**	**48.71**
He et al. [46]	71.41	70.83	0.58	60.80	51.60
SFP [16]	71.40	68.70	2.61	59.40	52.60
FPGM [38]	71.41	69.66	1.75	59.40	52.60
GBIP (*k* = 0.5)	**71.36**	**71.18**	**0.18**	**40.33**	**68.27**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

Table 7 Performance comparison of ResNet-18 on ILSVRC-2012

Method	Pruned Top-1 Acc/%	Top-1 Acc.drop/%	Pruned Top-5 Acc/%	Top-5 Acc.drop/%	Parameters.drop/%	FLOPs.drop/%
MIL [54]	66.33	3.43	86.94	2.14	–	33.30
DSA [55]	68.61	1.11	88.35	0.72	–	40.00
SFP [16]	67.10	3.18	87.78	1.85	–	41.80
FPGM [38]	68.41	1.35	88.48	0.60	–	41.80
EPFS-F-0.05 [3]	67.81	1.94	88.37	0.87	34.60	42.10
PFP [56]	65.65	4.11	86.75	2.33	–	43.00
DAIS [53]	67.56	2.20	87.90	1.18	–	43.30
GBIP (*k* = 0.5)	**69.36**	**0.66**	**88.71**	**0.52**	**36.78**	**45.55**
ABCPruner [31]	67.28	2.38	87.67	1.41	43.55	44.88
FBS [57]	68.17	1.59	88.22	0.86	–	49.50
ManiDP [21]	68.88	0.88	88.76	0.32	–	51.00
GBIP (*k* = 0.6)	**69.20**	**0.82**	**88.60**	**0.63**	**46.32**	**51.65**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

pruning is significantly greater. The Top-1 accuracy in [31] loses 2.38%, while the performance only drops 0.66% via our method, and its Top-5 accuracy also decreases 0.89% higher than GBIP. FBS [57] pruning 3.95% FLOPs higher than that of GBIP, and its Top-1 and Top-5 accuracy loss is also higher than ours by 0.93% (1.59% vs. 0.66%) and 0.34% (0.86% and 0.52%) respectively. When $k = 0.5$, the cutting rate of FLOPs using GBIP is 5.45% lower than that of ManiDP [21], and the Top-5 accuracy drop is 0.20% higher (0.52% vs. 0.32%), but the Top-1 accuracy loss is 0.22% lower (0.66% vs. 0.88%). When $k = 0.6$, GBIP deletes 46.32% of the parameters and 51.65% of the FLOPs. In this scenario, the compression degree is significantly higher than the comparative pruning algorithms, and the accuracy loss is also higher. To the best of our knowledge, this is because the number of remaining parameters is too little to adequately extract the target information in the images during the learning process with the continuous compression of the network, and it results in the decrease of the final classification performance.

After that, we continue to conduct the pruning experiments on the ResNet-50. Table 8 depicts the performance comparison of pruning ResNet-50. The original Top-1 and Top-5 accuracy of ResNet-50 are 75.94% and 92.93% respectively. When pruning 55.36% parameters and 63.34% FLOPs, the Top-1 accuracy only decreases by 0.47% and the Top-5 accuracy drop is even less. Although the Top-1 accuracy loss of CNN-FCF [40] is the same as ours and their Top-5 accuracy loss is 0.04% less, the compressing rates of parameters and FLOPs of our GBIP are respectively 12.95% (55.36% vs. 42.41%) and 17.29% (63.34% vs. 46.05%) higher than CNN-FCF.

All the experiments for image classification reveal that our global balanced iterative pruning method can achieve a similar degree of compression rate on the parameters and FLOPs of convolutional networks. For simple tasks, after using GBIP for network pruning, overfitting is eliminated, and the performance of the compact network can maintain or even exceed the accuracy of the original network after retraining. The complex classification task requires more

Table 8 Performance comparison of ResNet-50 on ILSVRC-2012

Method	Pruned Top-1 Acc/ %	Top-1 Acc.drop/ %	Pruned Top-5 Acc/ %	Top-5 Acc.drop/ %	Parameters.drop/ %	FLOPs.drop/ %
ThiNet [58]	74.03	1.27	92.11	0.09	33.72	36.79
SFP [16]	74.61	1.54	92.06	0.81	–	41.80
HRank [19]	74.98	1.17	92.33	0.54	36.67	43.77
LSTM [45]	75.00	1.12	92.67	0.33	37.56	–
NISP [24]	–	0.89	–	–	43.82	44.01
Taylor-FO-BN [59]	74.50	1.68	–	–	44.53	45.00
CNN-FCF [40]	75.68	0.47	92.68	0.19	42.41	46.05
EDP [39]	75.34	0.56	92.43	0.34	43.90	52.60
FPGM [38]	74.83	1.32	92.32	0.55	–	53.50
CCP [41]	75.21	0.94	92.42	0.45	–	54.10
GAL [6]	71.80	4.35	90.82	2.05	24.27	55.00
RL-MCTS [50]	76.46	0.88	92.83	0.34	–	55.00
SRR-GR [49]	75.11	1.02	92.35	0.51	–	55.10
DAIS [53]	74.45	1.70	92.21	0.66	–	55.30
ABCPruner [31]	73.52	2.49	91.51	1.45	56.01	56.61
GReg-2 [51]	74.93	1.20	–	–	–	60.94
HRank [19]	71.98	4.17	91.01	1.86	46.00	62.10
GBIP (k = 0.4)	**75.47**	**0.47**	**92.70**	**0.23**	**55.36**	**63.34**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

parameters to extract the semantic information in the image. There are almost no redundant parameters in tiny convolutional networks, therefore pruning will be accompanied by a decrease in accuracy. However, our GBIP can still control the performance loss in a smaller range. It indicates that the iterative channel pruning method proposed in this paper can effectively remove unimportant parameters in the CNNs and reduce network redundancy in the sense that it also has regularization on the network training.

4.4 Pruning SSD on PASCAL VOC

The existing network pruning algorithms are almost totally for single-target image classification tasks with obvious targets and rarely involve other more complex tasks. In the real world, the scenarios of object detection are more extensive and the requirements for low storage and real-time are higher. However, in the context of uncertain conditions such as occlusion, size, and light changes, these tasks often need more complicated models. Therefore, compressing the model for object detection while maintaining accuracy faces salient challenges. To show off the generalization of the proposed method, we prune the SSD on the PASCAL VOC object detection dataset. The backbone of the SSD adopts the VGG-16 trained on the CIFAR-100. Here, we compare the parameters and FLOPs compressing rate and Mean Average Precision (mAP) loss. The results are depicted in Table 8. When $k = 0.3$, the pruning rates of the parameters and FLOPs are 47.66% and 30.25%, respectively. Compared with the mAP of the baseline of 76.10%, the detection accuracy of the compact SSD only decreases by 0.50%. While pruning 57.66% of the parameters and 54.06% of the FLOPs in the SSD with $k = 0.4$, the mAP drops by 0.90%.

To visually display the results of the pruned SSD in the object detection, we select five pictures in PASCAL VOC to visualize the experiments in Table 9. And the results are depicted in Fig. 5. The first line is the original images, and the second line is the detection result obtained using the baseline SSD, while the last two lines are the pruned results via GBIP with $k = 0.3$ and $k = 0.4$. It can be found from the figure that the compressed SSD can still correctly detect the object in the images, despite the position and size of the detection frame may alter slightly within an acceptable range. Moreover, the confidence of some targets will also fluctuate to a certain extent. For example, the baseline confidence of the tvmonitor in figure (d) is 0.97, but when $k = 0.3$ and $k = 0.4$, they are 0.92 and 0.95, respectively. The confidence of some targets in the other pictures is also different. We conjecture this is due to the detection accuracy of some categories has been improved after pruning,

Table 9 The results of pruning SSD on PASCAL VOC

Method	mAP/%	mAP.drop/%	Parameters/M	Parameters.drop/%	FLOPs/G	FLOPs.drop/%
Baseline	76.10	–	26.29	–	11.34	–
Keeffe et al. [60]	75.04	1.06	18.74	28.72	8.36	26.28
GBIP (*k* = 0.3)	**75.60**	**0.50**	**13.76**	**47.66**	**7.91**	**30.25**
Li et al. [14]	74.91	1.19	13.25	49.60	6.53	42.42
GBIP (*k* = 0.4)	**75.20**	**0.90**	**11.13**	**57.66**	**5.21**	**54.06**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

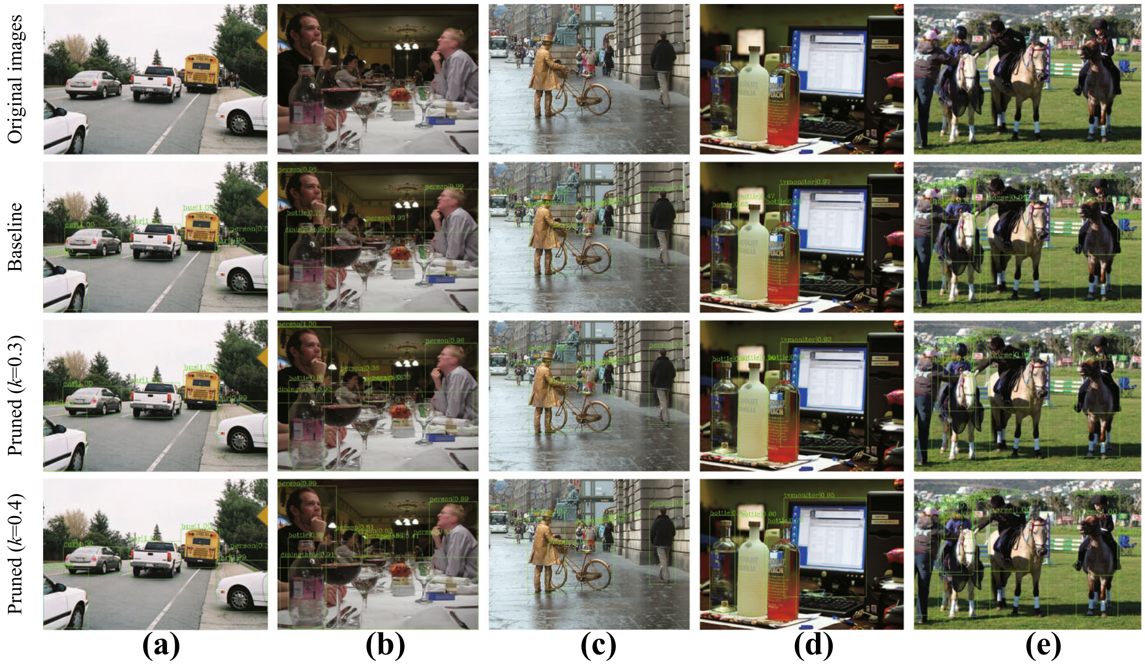


Fig. 5 Visualization of pruning SSD on the PASCAL VOC

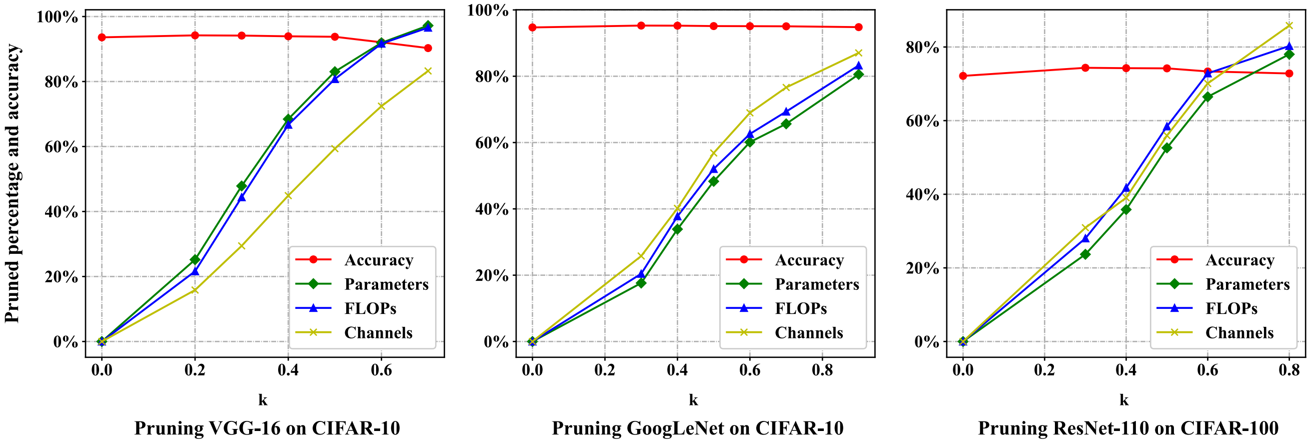
although the overall mAP is slightly lower. To be more specific, the detection precision of some targets will even improve when compressing the network. For instance, the baseline confidence of the bottle in figure (b) is 0.75, however, when $k = 0.3$ it reaches 0.85, and when $k = 0.4$ it even increases to 0.99. It also reveals that pruning can improve the capability of the network recognition for some target classes by reducing model redundancy. Moreover, when $k = 0.4$, the SSD is compressed by more than 50%. At this time, even more persons are accurately found than the baseline in figure(b). It manifests that the ability to distinguish people has been developed. The above experiments further verify that the pruning method in this paper also has good generalization in the field of object detection.

4.5 Ablation analysis

Then, we conduct the ablation analysis on the proposed GBIP method. This section is composed of the following three parts: the influence of k on the pruning rate, the influence of k on the compressing magnitude in different layers, and the influence of attention transfer, output transfer, and adversarial game on the network performance recovery.

4.5.1 The influence of k on the pruning rate

The pruning threshold factor k is the parameter used to adjust the compression ratio in our proposed pruning algorithm. The larger the k, the greater the pruning threshold, so that the higher the degree of network compression. To reveal the influence of the k, we perform six groups of pruning experiments on VGG-16 by setting different k on the CIFAR-10, and the results are shown in Fig. 6. It can be seen from the figure that as the k increases, the pruning rate of parameters, FLOPs, and channels constantly exceeds. The most important is that the parameters and FLOPs compressing rate are always balanced. When



Pruning VGG-16 on CIFAR-10

Pruning GoogLeNet on CIFAR-10

Pruning ResNet-110 on CIFAR-100

Fig. 6 The influence of k on the accuracy and pruning rate of CNNs

the value of k is small, the cropping ratio rises faster meanwhile the curve is relatively steep. But with the continuous growth of k, the curve of the parameters and the FLOPs gradually tends to be smooth, while the compressing rate of channels almost still linearly rises. From the figure, it is clear that when the compression ratio of parameters and FLOPs are less than 80%, and that of the number of channels is less than 60%, the accuracy of the pruned network remains unchanged or even slightly improved compared to the baseline. Nevertheless, the performance of the compact network begins to decline if continues to compress. This is because when pruning fewer parameters, the redundancy and the impact of overfitting are reduced so that the performance will be improved. But when removing too many parameters, the network is difficult to cope with the classification tasks which causes performance degradation.

4.5.2 The influence of k on the compression magnitude in different layers

To better show the compression amplitude of each layer of the network under different pruning ratios, this section visualizes the number of channels of VGG-16, ResNet-56, and ResNet-110 in CIFAR-10 as Fig. 7. The three rows from top to bottom are VGG-16, ResNet-56, and ResNet-

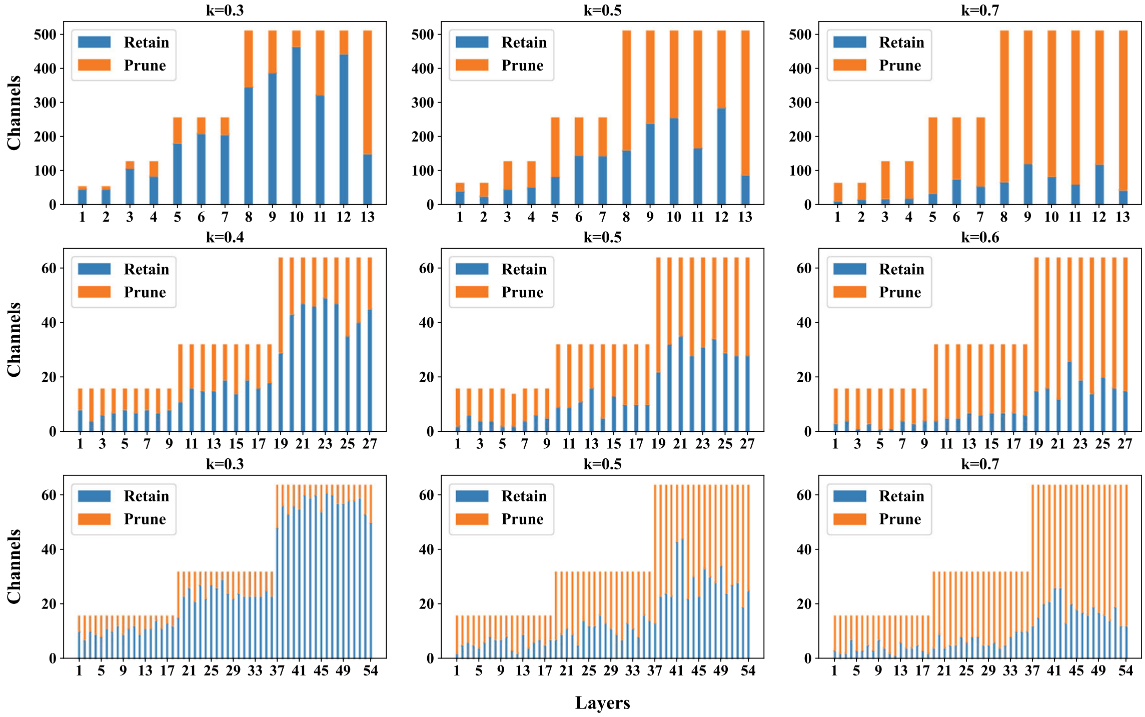


Fig. 7 The influence of k for channels of VGG-16 (top), ResNet-56 (second row), and ResNet-110 (bottom) on CIFAR-10

Table 10 Performance comparison of retraining pruned networks

Method			VGG-16 + CIFAR-10 ($k = 0.3$)		ResNet-56 + CIFAR-100 ($k = 0.5$)		ResNet-18 + ImageNet ($k = 0.5$)	
$\mathcal{L}_{AT}$	$\mathcal{L}_{OT}$	$\mathcal{L}_{AG}$	Acc/%	Acc.drop/%	Acc/%	Acc.drop/%	Top-1 Acc/%	Top-1 Acc.drop/%
			93.94	− 0.34	70.94	0.42	68.12	1.90
✓			94.00	− 0.40	71.06	0.30	68.26	1.76
	✓		94.03	− 0.43	71.09	0.27	68.54	1.48
		✓	94.02	− 0.42	71.07	0.29	68.37	1.65
✓	✓	✓	**94.14**	**− 0.54**	**71.18**	**0.18**	**69.36**	**0.66**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

Table 11 Performance analysis of our pruning method without using knowledge transfer and adversarial game strategy

network+task	Method	Pruned Acc/%	Acc.drop/%	FLOPs.drop/%
ResNet-56+CIFAR-10	HRank [19]	93.17	0.09	42.40
	NISP [24]	–	0.03	43.61
	AMC [29]	91.90	0.90	50.00
	DCP [37]	93.59	0.21	50.00
	SFP [16]	93.35	0.24	52.60
	FPGM [38]	92.89	0.70	52.60
	CCP [41]	93.42	0.08	52.60
	Y.He et al. [46]	93.34	0.25	52.90
	LEGR [23]	93.70	0.20	53.00
	GReg-2 [51]	93.36	0	60.78
	ManiDP [21]	93.64	0.06	62.40
	GBIP (*k*= 0.5)	**93.78**	**− 0.18**	**63.35**
ResNet-56+CIFAR-100	Y.He et al. [46]	70.83	0.58	51.60
	SFP [16]	68.70	2.61	52.60
	FPGM [38]	69.66	1.75	52.60
	GBIP (*k*=0.5)	**70.94**	**0.42**	**68.27**
ResNet-18+ImageNet	MIL [54]	66.33	3.43	33.30
	SFP [16]	67.10	3.18	41.80
	EPFS-F-0.05 [3]	67.81	1.94	42.10
	PFP [56]	65.65	4.11	43.00
	GBIP (*k*=0.5)	**68.12**	**1.90**	**45.55**

Bold indicates the experimental results of the method in this paper, which is used to compare with the peer's method

110. It can be found from the first row that the last layer of VGG-16 has the most remarkable redundancy. The last layer has eliminated more than 50.00% of the channels with $k = 0.3$, while the discarding ratio of the network is small. As the pruning ratio increases, the number of retained channels in the 9th and 12th layers is more than that in the other layers. It implies that the impact of the two layers on extracting target information is more pivotal than that of other layers. For ResNet-56, the number of channels reserved in the 23th layer is more than that of the 22th layer with little compression. But when $k = 0.6$, the cropping ratio raises, the number of channels saved in the 22th layer is significantly more than the other layers. It indicates that as the pruning rate changes, the importance of different layers also varies to improve the performance as much as possible. At the same time, it reiterates that the pruning strategy proposed in this paper can adaptively adjust the pruning range of each layer according to different compression rates to obtain a compact network that meets the performance requirements.

4.5.3 The influence of three modules on the performance recovery

To analyze the influence of attention transfer, output transfer, and adversarial game on the performance recovery of the compact network, we retrain the pruned VGG-16 and ResNet-18 via different strategies on the CIFAR-10/100 and ImageNet respectively. The number of training epochs and other hyperparameter settings remain the same. The results are tabulated in Table 10. As we can see from the table that the VGG-16 network trained by all the three strategies has the highest accuracy, which can reach 94.14%. The accuracy obtained using the output transfer is 94.03%, which is 0.03% higher than applying the attention transfer. When utilizing adversarial game, the accuracy is 94.02% which is 0.08% higher than that of retraining without the three strategies. Therefore, output transfer plays the most considerable role in the three modules. The results of ResNet-56 on CIFAR-100 and ResNet-18 on ImageNet also demonstrate the conclusion above. For large scale task ImageNet, the accuracy obtained via all three strategies is 69.36%, which is the best and 1.24% higher than training without any strategies. It can also be seen from the table that the performance via the adversarial game alone is better than only using the attention transfer. It is because the attention map only works on the intermediate output feature maps and does not restrict the final output, while the adversarial game can directly optimize the output features, so it can better improve the performance of the pruned network. In addition, we also compare our pruning strategy without using knowledge migration and adversarial games with some existing pruning methods that do not use optimization algorithms. The results are shown in Table 11. As we can see from the table that the pruning method in this paper still has leading performance in terms of network compression rate and accuracy without optimization.

5 Conclusion and future work

In this paper, we propose a global balanced iterative pruning method. The unimportant parameters and FLOPs can be eliminated in similar amplitude based on the magnitude distribution of the intermediate features. And then, we design a performance recovery scheme, so that the performance of the compressed network can be recovered as soon as possible after each pruning step. In this way, we can complete continuous iterative pruning in the training process of the network. The final compact network obtained will restore the accuracy through retraining from scratch. We conduct extensive experiments for pruning VGGNet, ResNet, and GoogLeNet on image classification datasets of CIFAR-10, CIFAR-100, and ILSVRC-2012. The results have manifested that GBIP is comparable with state-of-the-art network pruning methods in performance and pruning rate of parameters and FLOPs. On CIFAR-10, after compressing 75.29% of the parameters and 78.60% of the FLOPs of ResNet-56, the accuracy only drops by 0.39%. In the object detection task PASCAL VOC, when removing more than 50% of the parameters and FLOPs of the SSD, the mAP is only reduced by 0.9%. The final ablation analysis reveals that the pruning factor can achieve flexible control of the compression rate. The experiments fully show that the proposed method can be widely applied in different CNNs, image datasets, and various computer vision tasks. In the future, we will integrate the channel pruning method with other compression schemes such as quantization. Furthermore, we will consider applying existing approaches to accelerate other real-world vision tasks and even natural language processing.

Acknowledgements This work was supported in part by the Anhui Provincial Key Research and Development Program under Grant 202004a05020040, in part by the National Key Research and Development Program under Grant 2018YFC0604404, in part by Intelligent Network and New Energy Vehicle Special Project of Intelligent Manufacturing Institute of HFUT under Grant IMIWL2019003, and in part by Fundamental Research Funds for the Central Universities under Grant PA2021GDGP0061.

Data availability All data generated or analysed during this study are included in this article.

Declarations

Conflict of interest We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
2. Han S, Pool J, Tran J, Dally WJ (2015) Learning both weights and connections for efficient neural networks. In: NIPS, vol 28
3. Xu S, Chen H, Gong X, Liu K, Lü J, Zhang B (2021) Efficient structured pruning based on deep feature stabilization. Neural Comput Appl 33(13):7409–7420. https://doi.org/10.1007/s00521-021-05828-8
4. Liu S, Ni'mah I, Menkovski V, Mocanu DC, Pechenizkiy M (2021) Efficient and effective training of sparse recurrent neural networks. Neural Comput Appl 33(15):9625–9636. https://doi.org/10.1007/s00521-021-05727-y
5. Komodakis N, Zagoruyko S (2017) Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: ICLR
6. Lin S, Ji R, Yan C, Zhang B, Cao L, Ye Q, Huang F, Doermann DS (2019) Towards optimal structured CNN pruning via generative adversarial learning. In: CVPR, pp 2790–2799. https://doi.org/10.1109/CVPR.2019.00290
7. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: ICLR

8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: CVPR, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
9. Szegedy C, Liu W, Jia Y, Sermanet P, Reed SE, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: CVPR, pp 1–9. https://doi.org/10.1109/CVPR.2015.7298594
10. Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images
11. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein MS, Berg AC, Li F (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
12. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single Shot MultiBox Detector. In: ECCV, vol 9905, pp 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
13. Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A (2015) The PASCAL visual object classes challenge: a retrospective. Int J Comput Vis 111(1):98–136. https://doi.org/10.1007/s11263-014-0733-5
14. Li H, Kadav A, Durdanovic I, Samet H, Graf HP (2017) Pruning filters for efficient convnets. In: ICLR
15. Polyak A, Wolf L (2015) Channel-level acceleration of deep face representations. IEEE Access 3:2163–2175. https://doi.org/10.1109/ACCESS.2015.2494536
16. He Y, Kang G, Dong X, Fu Y, Yang Y (2018) Soft filter pruning for accelerating deep convolutional neural networks. In: IJCAI, pp 2234–2240. https://doi.org/10.24963/ijcai.2018/309
17. Molchanov P, Tyree S, Karras T, Aila T, Kautz J (2017) Pruning convolutional neural networks for resource efficient inference. In: 5th International conference on learning representations, ICLR. https://openreview.net/forum?id=SJGCiw5gl
18. Liu Z, Li J, Shen Z, Huang G, Yan S, Zhang C (2017) Learning efficient convolutional networks through network slimming. In: ICCV, pp 2755–2763. https://doi.org/10.1109/ICCV.2017.298
19. Lin M, Ji R, Wang Y, Zhang Y, Zhang B, Tian Y, Shao L (2020) Hrank: filter pruning using high-rank feature map. In: CVPR, pp 1526–1535. https://doi.org/10.1109/CVPR42600.2020.00160
20. Li H, Ma C, Xu W, Liu X (2020) Feature statistics guided efficient filter pruning. In: IJCAI, pp 2619–2625. https://doi.org/10.24963/ijcai.2020/363
21. Tang Y, Wang Y, Xu Y, Deng Y, Xu C, Tao D, Xu C (2021) Manifold regularized dynamic network pruning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5018–5028
22. Wu H, Tang Y, Zhang X (2021) A pruning method based on the measurement of feature extraction ability. Mach Vis Appl 32(1):1–11. https://doi.org/10.1007/s00138-020-01148-4
23. Chin T-W, Ding R, Zhang C, Marculescu D (2020) Towards efficient model compression via learned global ranking. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1515–1525. https://doi.org/10.1109/CVPR42600.2020.00159
24. Yu R, Li A, Chen C-F, Lai J-H, Morariu VI, Han X, Gao M, Lin C-Y, Davis LS (2018) Nisp: pruning networks using neuron importance score propagation. In: CVPR, pp 9194–9203. https://doi.org/10.1109/CVPR.2018.00958
25. Lee N, Ajanthan T, Torr PHS (2019) Snip: single-shot network pruning based on connection sensitivity. In: ICLR
26. Guo S, Wang Y, Li Q, Yan J (2020) DMCP: differentiable Markov channel pruning for neural networks. In: CVPR, pp 1536–1544. https://doi.org/10.1109/CVPR42600.2020.00161
27. You Z, Yan K, Ye J, Ma M, Wang P (2019) Gate decorator: global filter pruning method for accelerating deep convolutional neural networks. In: NeurIPS, pp 2130–2141
28. Guo J, Ouyang W, Xu D (2020) Channel pruning guided by classification loss and feature importance. Proc. AAAI Conf. Artif. Intell. 34:10885–10892
29. He Y, Lin J, Liu Z, Wang H, Li L-J, Han S (2018) Amc: Automl for model compression and acceleration on mobile devices. In: ECCV, pp 815–832. https://doi.org/10.1007/978-3-030-01234-2_48
30. Liu Z, Mu H, Zhang X, Guo Z, Yang X, Cheng K-T, Sun J (2019) Metapruning: meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3296–3305
31. Lin M, Ji R, Zhang Y, Zhang B, Wu Y, Tian Y (2020) Channel pruning via automatic structure search. In: IJCAI, pp 673–679. https://doi.org/10.24963/ijcai.2020/94
32. Ding G, Zhang S, Jia Z, Zhong J, Han J (2020) Where to prune: using lstm to guide data-dependent soft pruning. IEEE Trans Image Process 30:293–304. https://doi.org/10.1109/TIP.2020.3035028
33. Ba J, Caruana R (2014) Do deep nets really need to be deep? In: NeurIPS, pp 2654–2662
34. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network
35. Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y (2015) Fitnets: hints for thin deep nets. In: ICLR
36. Liu Z, Sun M, Zhou T, Huang G, Darrell T (2019) Rethinking the value of network pruning. In: ICLR
37. Zhuang Z, Tan M, Zhuang B, Liu J, Guo Y, Wu Q, Huang J, Zhu J (2018) Discrimination-aware channel pruning for deep neural networks. In: NeurIPS, pp 883–894
38. He Y, Liu P, Wang Z, Hu Z, Yang Y (2019) Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4340–4349. https://doi.org/10.1109/CVPR.2019.00447
39. Ruan X, Liu Y, Yuan C, Li B, Hu W, Li Y, Maybank S (2020) Edp: an efficient decomposition and pruning scheme for convolutional neural network compression. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2020.3018177
40. Li T, Wu B, Yang Y, Fan Y, Zhang Y, Liu W (2019) Compressing convolutional neural networks via factorized convolutional filters. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3977–3986. https://doi.org/10.1109/CVPR.2019.00410
41. Peng H, Wu J, Chen S, Huang J (2019) Collaborative channel pruning for deep networks. In: International conference on machine learning, pp 5113–5122
42. Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J (2019) Importance estimation for neural network pruning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11264–11272. https://doi.org/10.1109/CVPR.2019.01152
43. Gao S, Huang F, Cai W, Huang H (2021) Network pruning via performance maximization. In: IEEE conference on computer vision and pattern recognition, CVPR 2021, pp 9270–9280
44. Zhao C, Ni B, Zhang J, Zhao Q, Zhang W, Tian Q (2019) Variational convolutional neural network pruning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2780–2789. https://doi.org/10.1109/CVPR.2019.00289
45. Ding G, Zhang S, Jia Z, Zhong J, Han J (2021) Where to prune: using LSTM to guide data-dependent soft pruning. IEEE Trans Image Process 30:293–304. https://doi.org/10.1109/TIP.2020.3035028
46. He Y, Ding Y, Liu P, Zhu L, Zhang H, Yang Y (2020) Learning filter pruning criteria for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF conference on

computer vision and pattern recognition, pp 2009–2018. https://doi.org/10.1109/CVPR42600.2020.00208

47. He Y, Zhang X, Sun J (2017) Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE international conference on computer vision, pp 1389–1397
48. Gao S, Huang F, Pei J, Huang H (2020) Discrete model compression with resource constraint for deep neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1899–1908. https://doi.org/10.1109/CVPR42600.2020.00197
49. Wang Z, Li C, Wang X (2021) Convolutional neural network pruning with structural redundancy reduction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14913–14922
50. Wang Z, Li C (2022) Channel pruning via lookahead search guided reinforcement learning. In: IEEE/CVF winter conference on applications of computer vision, WACV, pp 3513–3524. https://doi.org/10.1109/WACV51458.2022.00357
51. Wang H, Qin C, Zhang Y, Fu Y (2021) Neural pruning via growing regularization. In: 9th International conference on learning representations, ICLR 2021. https://openreview.net/forum?id=o966_Is_nPA
52. Singh P, Kadi VSR, Namboodiri VP (2020) FALF convnets: fatuous auxiliary loss based filter-pruning for efficient deep cnns. Image Vis Comput 93:103857. https://doi.org/10.1016/j.imavis.2019.103857
53. Guan Y, Liu N, Zhao P, Che Z, Bian K, Wang Y, Tang J (2022) Dais: automatic channel pruning via differentiable annealing indicator search. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3161284
54. Dong X, Huang J, Yang Y, Yan S (2017) More is less: a more complicated network with less inference complexity. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5840–5848. https://doi.org/10.1109/CVPR.2017.205
55. Ning X, Zhao T, Li W, Lei P, Wang Y, Yang H (2020) Dsa: more efficient budgeted pruning via differentiable sparsity allocation. In: ECCV 2020, pp 592–607. https://doi.org/10.1007/978-3-030-58580-8_35
56. Liebenwein L, Baykal C, Lang H, Feldman D, Rus D (2020) Provable filter pruning for efficient neural networks. In: ICLR 2020
57. Gao X, Zhao Y, Dudziak L, Mullins RD, Xu C (2019) Dynamic channel pruning: feature boosting and suppression. In: ICLR 2019
58. Luo J, Zhang H, Zhou H, Xie C, Wu J, Lin W (2019) Thinet: pruning CNN filters for a thinner net. IEEE Trans Pattern Anal Mach Intell 41(10):2525–2538. https://doi.org/10.1109/TPAMI.2018.2858232
59. Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J (2019) Importance estimation for neural network pruning. In: CVPR, pp 11264–11272. https://doi.org/10.1109/CVPR.2019.01152
60. O'Keeffe S, Villing R (2018) Evaluating extended pruning on object detection neural networks. In: 2018 29th Irish signals and systems conference (ISSC), pp 1–6. https://doi.org/10.1109/ISSC.2018.8585345

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.,