Complex & Intelligent Systems
https://doi.org/10.1007/s40747-023-01022-6

ORIGINAL ARTICLE

Differentiable channel pruning guided via attention mechanism: a novel neural network pruning approach

Hanjing Cheng[1] · Zidong Wang[2,3] · Lifeng Ma[4] · Zhihui Wei[5] · Fawaz E. Alsaadi[6] · Xiaohui Liu[3]

Received: 25 September 2022 / Accepted: 17 February 2023

Abstract

Neural network pruning offers great prospects for facilitating the deployment of deep neural networks on computational resource limited devices. Neural architecture search (NAS) provides an efficient way to automatically seek appropriate neural architecture design for compressed model. It is observed that, for existing NAS-based pruning methods, there is usually a lack of layer information when searching the optimal neural architecture. In this paper, we propose a new NAS approach, namely, differentiable channel pruning method guided via attention mechanism (DCP-A), where the adopted attention mechanism is able to provide layer information to guide the optimization of the pruning policy. The training process is differentiable with Gumbel-softmax sampling, while parameters are optimized under a two-stage training procedure. The neural network block with the shortcut is dedicatedly designed, which is of help to prune the network not only on its width but also on its depth. Extensive experiments are performed to verify the applicability and superiority of the proposed method. Detailed analysis with visualization of the pruned model architecture shows that our proposed DCP-A learns explainable pruning policies.

Keywords Artificial Intelligence · Network pruning · Neural Architecture Search · Gumbel-softmax sampling · Attention mechanism

Introduction

Deep neural networks (DNNs) have achieved remarkable accomplishments in a variety of applications such as pattern recognition [5,18,31,41,60], and have also shown sustained superiorities in comparison to other methods. However, the large amount of model parameters and high performance demand on GPUs have also brought about great challenges on storage and time costs. Therefore, much research attention has been devoted to the operation problem of DNNs on computationally limited devices such as mobile equipments and embedded devices. As a rather popular approach, neural network pruning offers a great prospect for facilitating the deployment of DNNs on computational-resource-limited devices. In general, the widely applied neural network pruning approaches can be divided into two categories, namely, weight pruning [8,10,20,63,64] and channel pruning [4,6,12,21,61,62]. Since weight pruning cannot harvest obvious acceleration for modern networks due to its unstructured operation manner, we focus on channel pruning in this paper.

There are two types of channel pruning methods, i.e., criterion- and NAS-based channel pruning methods. The main procedure of the criterion-based channel pruning techniques is to first determine the basic criterion and then prune filters hierarchically, which would require us to *manually* set the pruning ratio for each layer. In practice, the pruning ratio is usually set to be *equal* for each layer so as to simplify the entire process. Unfortunately, such a simplification could

✉ Zidong Wang
Zidong.Wang@brunel.ac.uk

[1] College of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215000, China

[2] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

[3] Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

[4] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

[5] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

[6] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

Published online: 27 March 2023

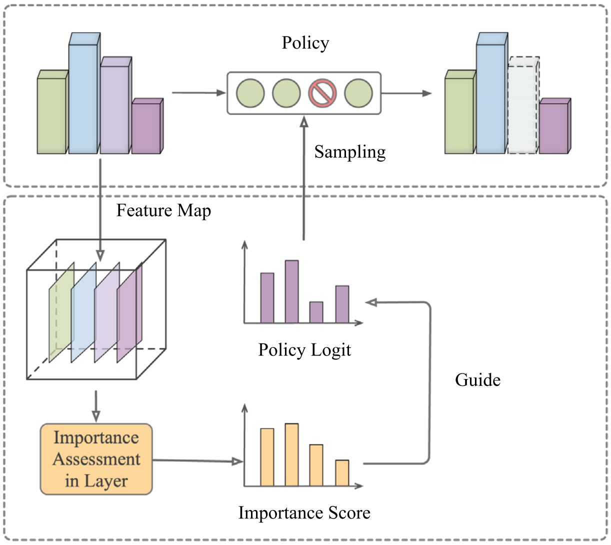


Fig. 1 An illustration of DCP-A where the policy logits are designed for all filters in the network and, in each layer, the importance scores are assessed by a special mechanism to obtain the information among filters and then guide the optimization of policy logits

lead to poor performance due to the fact that different layers possess different redundancies. On the other hand, neural architecture search provides a powerful tool to automatically seek efficient neural architecture. So far, extensive studies have been conducted on the neural architecture search problem with the aim to explore the optimal network structures in a large design space while taking into account the trade-off among the model size, the speed, and the accuracy. Note that, when utilizing traditional NAS-based methods, we usually confront difficulty in searching a large space with unacceptable searching speed. Although some effort has been devoted to reducing the searching space [12,22,24,28,38], the layer information, i.e., the information of filters in one layer, has seldom been taken into consideration when it comes to the optimization of the pruning policy. Basically, most criterion-based pruning methods fail to take the correlation among layers into consideration while NAS-based methods usually ignore the information of individual filters in a layer.

Inspired by the above discussion, in this paper, we propose a new differentiable channel pruning framework guided via attention mechanism (DCP-A), shown in Fig. 1, where certain policy is used to determine the pruning decision and the importance scores in a layer are used to guide the optimization of the policy logit. The importance scores can be obtained by any pruning criterion. In this paper, we choose the importance scores obtained by l_1 norm, l_2 norm and attention mechanism. Here, the policy logit guided by the attention mechanism shows the best experiment result. The attention mechanism is a concept derived from cognitive psychology that allows models to devote limited resources to more important channels [35]. Pruning policy of pruning-or-not is sampled from the policy logit which is defined for each filter in the network. To obtain layer information, attention score with attention-guided loss is adopted to regulate the optimization of policy logit. Hence, the attention score provides the correlation of filters in layer and, meanwhile, the attention guided loss limits the searching space for pruning policy. Moreover, a two-stage training procedure is proposed to ensure that the introduced attention modules are well-trained and easily removed (without increase of the final FLOPs of pruned network).

The main contributions of this paper can be highlighted as follows:

(1) a new NAS-based differentiable channel pruning framework is proposed, where importance scores obtained by different mechanisms (including the attention mechanism) are adopted to provide a layer information for the optimization of pruning policy logit;
(2) a two-stage training procedure with designed training objectives is proposed to optimize the network parameters, the policy logits and the attention modules;
(3) for networks with shortcut structure (e.g. ResNet), the proposed DCP-A algorithm is capable of pruning networks not only on the width but also on the depth;
(4) the proposed DCP-A can be easily extended into the multi-model case;
(5) via extensive experiments, the effectiveness and efficiency of the proposed DCP-A framework are demonstrated in different databases, and detailed analysis is provided through structure visualization to show that the pruning policies learned by DCP-A are explainable.

The remainder of this paper is organized as follows. In “Related work” section, we introduce the related works of model pruning, neural architecture search and attention mechanism. In “Methodology” section, we describe our DCP-A framework in detail. The experimental study and the corresponding analysis are presented in “Experiments” section. “Conclusion” section gives the conclusions of this paper.

Related work

In terms of its objectives, the model pruning can be generally classified into two categories, namely, weight pruning and channel pruning. On one hand, weight pruning directly removes connections in filters, which might lead to unstructured sparsity and, furthermore, make it difficult to accelerate the inference with general-purpose hardware. On the other hand, channel pruning prunes entire filters to deploy existing basic linear algebra subprograms (BLASs) libraries, thereby achieving better acceleration. Considering how to design the pruning policy, we can roughly divide channel pruning methods into criterion-based pruning and NAS-based pruning.

Criterion-based pruning

Generally, criterion-based pruning methods assess the importance of filters by utilizing filter weights or filter activations. In [21], the importance of a filter has been calculated by the corresponding absolute weights sum, according to which the unimportant filters have been pruned. Filters with small l_2 norm have been slightly pruned in [13]. In [14], filters near geometric median have been pruned with the most replaceable contribution. In [4], three criteria have been utilized to find the important filters for satisfying the least replacing loss, the diversity and the high entropy of weights. It is worth noticing that all the aforementioned criterion-based methods use *manual* settings for the pruning ratio for layers.

NAS-based pruning

In early results concerning NAS, the optimal network structures have been found by resorting to the reinforcement learning [68] or evolutionary algorithms [55] which would consume substantial computation costs. Gradient-based NAS methods [29,53,54] have been exploited to reduce the cost by making the searching mechanism differentiable or approximately differentiable to enhance the searching efficiency. In [25], a partial order pruning method has been developed to automatically search the architectures with the best tradeoff between speed and accuracy. In [28], channel number in each layer has been searched based on the artificial bee colony algorithm. In [26], the designed hypernetwork has taken the latent vectors as the input and generated the weight parameters of the backbone network. It should be pointed out that, however, the aforementioned methods only take global network information into account, and there is still a lack of layer information when conducting the searching.

Attention mechanism

In [16,48], attention modules have been proposed to help DNNs focus on important channels and achieve a better performance. Recently, the attention mechanism has been considered in model pruning as an importance evaluation criterion of filters. In [52], an attention module has been embedded into model to generate scaling factors for channels that are considered as channels importance scores. In [6], a long short-term memory has been introduced to generate a strategy indicating the number of pruning filters for each layer. In this strategy, attention blocks have been embedded in the network, and filters with less attention scores have been forbidden in a feed-forward manner. In both the methods mentioned above, the attention score has been used directly to rank the filters in a layer.

Methodology

Approach overview

For a network that needs to be pruned, it is our goal to learn a pruning policy that determines the filter to be pruned with the least performance loss. Attention module with an attention score is utilized to evaluate the importance of filters in the layer. Note that attention modules are not expected to directly influence the optimization of network parameters because they will be removed from the pruned network to avoid increasing the FLOPs. Therefore, we define pruning policies for all filters in the whole network and use the attention module as a *guided* tool only.

In Fig. 2, an overview of our proposed DCP-A training approach is illustrated, which consists of two stages in the training epochs: (1) the stage of training parameters of the network, and (2) the stage of optimizing attention modules (Squeeze-and-Excitation block used in this paper) as well as policy logits. To be more specific, such a two-stage approach is explained as follows.

(a) *Stage one:* In the first stage, policy logits and parameters of attention blocks are fixed, while the parameters of network are free (to be optimized). It should be mentioned that attention modules do not participate in feed-forward in this stage, and only the average attention score of each attention block is recorded.

(b) *Stage two:* In the second stage, the parameters of network are fixed, while the parameters of attention blocks and policy logits are set to be free (to be optimized). Here, attention modules are activated for updating parameters. Attention scores obtained in the previous stage will be utilized as a *guidance* for optimizing the policy logits.

By repeating two stages alternately during training, optimal pruning pattern can be learned, resulting in a well-pruned network. Gumbel-softmax trick is utilized to make the training process differentiable. The details of our approach will be described in the following.

Attention mechanism

In this paper, the Squeeze-and-Excitation (SE) block proposed in [16] is employed to obtain the attention scores. The SE module (also known as the channel attention module) is able to select the most useful feature among channels, thereby improving the effectiveness of the feature representa-

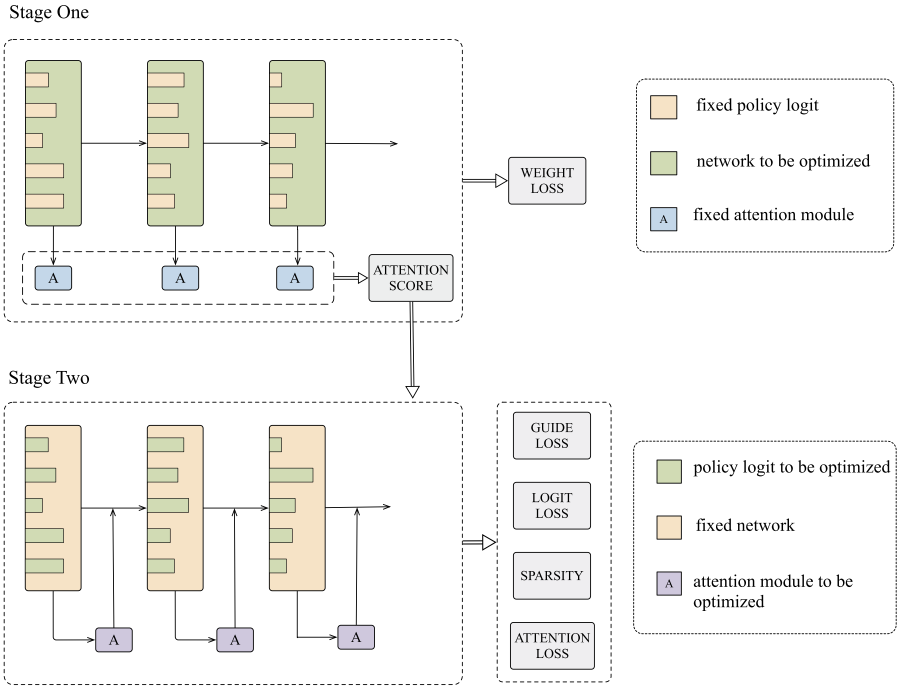


Fig. 2 An overview of DCP-A training approach where the policy logits and the attention modules are fixed in stage one and freed in stage two. The parameters of the network are optimized in stage one and fixed in stage two. Especially, attention scores of individual filters in a layer are recorded in stage one and used to guide the optimization in stage two

tions. Moreover, SE block is an effective attention block that can be flexibly embedded into most existing network structures and, consequently, the SE block has been widely used in computer vision applications [45]. An SE block contains two parts in its structure, namely, squeeze and excitation.

(a) *Squeeze:* In the squeeze part, the global information of each feature channel is obtained by an average pooling layer. Assume that the input of lth SE block is $X_l = [x_l^1, x_l^2, \ldots, x_l^C] \in \mathbb{R}^{H\times W\times C}$, then the average global information of each channel is defined as

$$z_l^k = \mathcal{A}(x_l^k) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_l^k(i, j) \tag{1}$$

where $\mathcal{A}(\cdot)$ is the global average pooling function, and $x_l^k(i, j)$ represents the pixel value.

(b) *Excitation:* In the excitation part, the global information are fused as follows to obtain the attention score S_l of each channel:

$$S_l = \delta(W_2 \sigma(W_1 z_l)) \tag{2}$$

where $W_1 \in \mathbb{R}^{\frac{C}{r}\times C\times 1\times 1}$ and $W_2 \in \mathbb{R}^{C\times \frac{C}{r}\times 1\times 1}$ are the correlation of channels; r is the reduction ratio; σ represents the activation function ReLU; and δ denotes the activation function Sigmoid.

In the literature, it has been shown that the SE block possesses the ability to generate importance scores for channels and, therefore, enhancing the network performance. As shown in Fig. 3, pruning filters in one layer can be performed based on the attention scores. For example, we can set the threshold to be 0.5, and prune nearly half of filters with attention scores less than 0.5. However, in the whole network, such a technique is not applicable anymore as the attention score only reflects the relationship of filters in the same layer. In Fig. 4, it can be seen that attention scores of different layers are extremely separated, while those in the same layer are relatively concentrated within a very small area. Obviously, the network pruning would fail if we were to directly set a threshold for attention scores to prune the whole network.

With the purpose of conquering the above-mentioned difficulty, we define a policy of pruning-or-not for each filter in the network.

Network pruning policy

Assume that a neural network has L layers with weights $W_l \in \mathbb{R}^{K\times K\times C_l^I \times C_l^O}$, where K is the kernel size, C_l^I and C_l^O represent the sizes of input and output channels, respectively.

For kth filter $f_{l,k}$ in lth layer, we introduce a binary-valued variable $u_{l,k}$ to determine pruning or not. It should be mentioned that the probability of pruning $f_{l,k}$ is sampled from a discrete probability distribution, and the back-propagation is not allowed because of non-differentiability problem. Hence, we employ the Gumbel-Softmax trick [17] to substitute the original non-differentiable sample (from a discrete distribution) with a differentiable sample (from a corresponding Gumbel-Softmax distribution) [12,44].

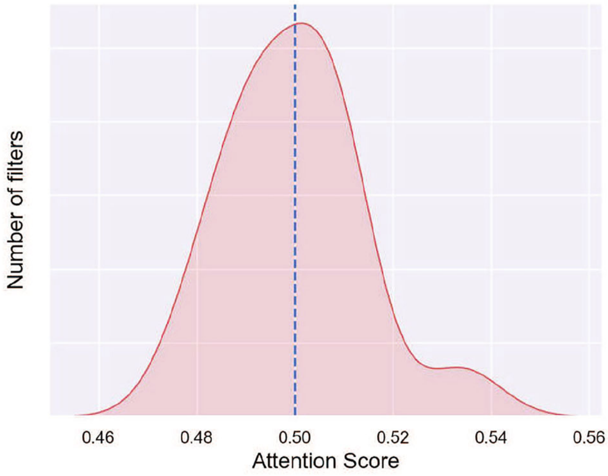


Fig. 3 An illustration of attention score distribution for single layer

We use $\pi_{l,k} = [1 - \alpha_{l,k}, \alpha_{l,k}]$ to represent the distribution vector of $u_{l,k}$, where the logit $\alpha_{l,k}$ indicates the possibility of pruning $f_{l,k}$. Then, in Gumbel-softmax sampling, $u_{l,k}$ is generated as

$$u_{l,k} = \arg\max_{j \in \{0,1\}} \{\log \pi_{l,k}(j) + G_{l,k}(j)\} \tag{3}$$

where

$$G_{l,k} = -\log(-\log U_{l,k})$$

is a standard Gumbel distribution with $U_{l,k}$ sampled from a uniform i.i.d. distribution $\mathcal{U}(0, 1)$. Then, the one-hot vector of $u_{l,k}$ is reformulated to the soft decision $v_{l,k}$ with reparameterization trick as follows:

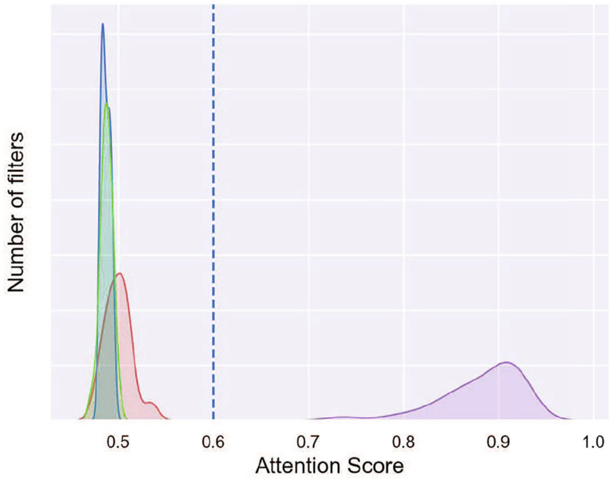


Fig. 4 An illustration of attention score distributions for different layers. If a threshold is set to be 0.6, then filters of three layers (red, blue and green) will be completely pruned. In contrast, filters of one layer (purple) will be entirely reserved

$$v_{l,k}(j) = \frac{\exp((\log \pi_{l,k}(j) + G_{l,k}(j))/\tau)}{\sum_{i \in \{0,1\}} \exp((\log \pi_{l,k}(i) + G_{l,k}(i))/\tau)} \tag{4}$$

where $j \in \{0, 1\}$ and τ is the softmax temperature. When $\tau \to \infty$, the Gumbel-softmax distribution is smooth and $\alpha_{l,k}$ can be optimized with gradient descent. When $\tau \to 0$, $v_{l,k}$ becomes one-hot.

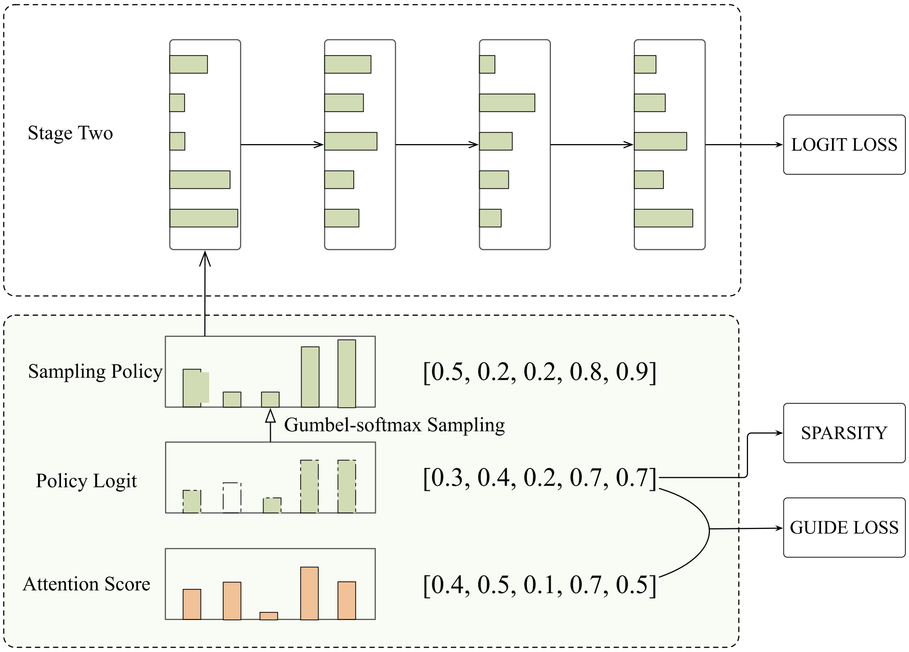


Fig. 5 An illustration of training objectives in stage two. The training losses of accuracy contain accuracy losses of optimizing parameters in the network, SE block and policy logit. Sparsity regularization ensures the possibility of pruning filters and attention score guided loss is introduced to guide the optimization

Algorithm 1 The DCP-A framework

```
Input: training data $X$; parameters of network $\theta_W$; parameters of SE block $\theta_{SE}$; policy logit $\theta_\pi$; FLOPs limitation $\gamma$; Temperature $\tau$
Output: pruned network with parameters $\theta_{W*}$;
Randomly split $X$ into sub-training data $X_1$ and sub-training data $X_2$;
for $epoch = 1$ to $max$ do
    Fix $\theta_{SE}$ and $\theta_\pi$.
    Free $\theta_W$.
    for $x$ in $X_1$ do
        Sample policy with $\theta_\pi$ and $\tau$ based on (4).
        Optimize $\theta_W$.
        Record attention score $S$.
    end for
    Free $\theta_{SE}$ and $\theta_\pi$.
    Fix $\theta_W$.
    for $x$ in $X_2$ do
        Sample policy with $\theta_\pi$ and $\tau$ based on (4).
        Optimize $\theta_{SE}$ and $\theta_\pi$ with $S$.
    end for
    Update $\tau$
end for
Prune network with $\gamma$ and updated policy logit $\theta_{\pi*}$.
Train the pruned network and obtain $\theta_{W*}$.
```

Training objectives

For training objectives, training losses of accuracy contain $\mathcal{L}(\theta_W)$, $\mathcal{L}(\theta_{SE})$ and $\mathcal{L}(\theta_\pi)$, which represent accuracy losses of optimizing parameters in network, SE block and policy logit, respectively.

In consideration of pruning mission, sparsity regularization $\mathcal{L}_{sparsity}(\theta_\pi)$ is adopted to ensure the possibility of pruning filters, which is defined as

$$\mathcal{L}_{sparsity}(\theta_\pi) = \frac{1}{L}\sum_l \left(w_l \sum_i (1-\alpha_{l,i}) \right) \tag{5}$$

where w_l represents the influence imposed by lth layer on FLOPs of pruning filters.

In most existing techniques, only $\mathcal{L}_{sparsity}(\theta_\pi)$ and $\mathcal{L}(\theta_\pi)$ are used to optimize θ_π. Since attention score is introduced to provide the layer information in this paper, the attention score guided loss should be taken into consideration as an objective, which is defined as follows:

$$\mathcal{L}_{guided}(\theta_\pi) = \frac{1}{L}\sum_l dist(S_l, 1-\alpha_l) \tag{6}$$

where $S_l \in \mathbb{R}^{1\times 1\times C}$ is the average attention score of SE block obtained in the first stage, and $dist(\cdot)$ measures the cosine distance as follows:

$$dist(u, v) = 1 - \frac{u \cdot v}{\|u\|\|v\|} \tag{7}$$

The exhibition of training objectives in stage two is shown in Fig. 5.

Finally, the total loss function is defined as

$$\begin{aligned}\mathcal{L}_{total} =& \mathcal{L}(\theta_W) + \mathcal{L}(\theta_{SE}) + \mathcal{L}(\theta_\pi) \\ &+ \lambda_1 \mathcal{L}_{sparsity}(\theta_\pi) + \lambda_2 \mathcal{L}_{guided}(\theta_\pi)\end{aligned} \tag{8}$$

where λ_1 and λ_2 control the weights of $\mathcal{L}_{sparsity}$ and $\mathcal{L}_{guided}$, respectively, and θ_W, θ_{SE} and θ_π will be optimized alternately during training.

Consequently, we describe the whole DCP-A framework in Algorithm 1.

Architectural design

Since network block with shortcut has been widely used nowadays, in this paper, two types of block architecture (basic block and bottleneck block) are considered with special design.

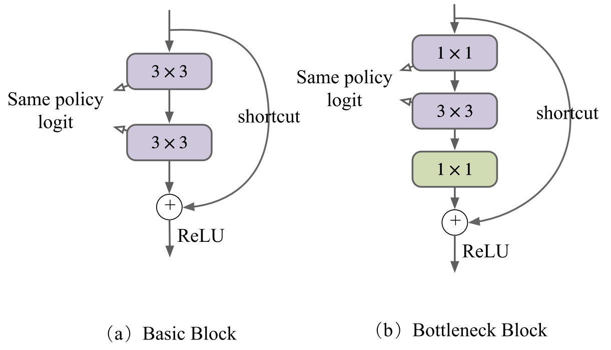
As shown in Fig. 6, for basic block consisting of two convolutional layers and a shortcut, we use the same policy logit for layers in the same block. For a bottleneck block with three convolutional layers, we use the same policy for the input and middle layer, and a new policy for the output layer. The pruning ratios of layers in the same block are set to be the same. Note that shortcut is protected in our method. Due to the special architecture of shortcut, the output equals the input if the policies are zero vectors, which is equivalent to skipping the whole block. Hence, protecting shortcut will help DCP-A skip network block and change the depth of the network.

Extension to multi-model pruning

As shown in Fig. 7, “Widen-Compression” is provided in DCP-A for multi-model pruning case. Assuming that the original layer in one model has 4 filters, the policy logit will be widened to 8 (doubled). When both strategies A and B choose to reserve a filter in the same position (e.g. the 7th position in Fig. 7), this filter will be shared in pruned models. Hence, DCP-A can help design the shared structure in multi-model pruning.

Experiments

Our implementation is in PyTorch [37] with an NVIDIA 2080Ti GPU. Experiments on different databases have proved the effectiveness of our method. We also exhibit various details of pruned model visually to further explore the rationality of DCP-A.



(a) Basic Block (b) Bottleneck Block

Fig. 6 An illustration of architectural design. For basic block, the same policy logit is used in the same block. For bottleneck block, the same policy is used for the input and middle layer, and a new policy is used for the output layer

Experimental settings

Databases

We evaluate our established DCP-A framework on the following databases: (1) CIFAR-10 and CIFAR-100 [19] that contain 60,000 color images in each database, with 50,000 training images and 10,000 testing images; (2) ILSVRC-2012 [39] (ImageNet) which is a large-scale dataset containing 1.28 million training images and 50,000 validation images of 1,000 classes; and (3) NYU-v2 [42] which is comprised of 1,449 video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras, and include 795 images for training and 654 images for validation. We use 40-class annotation for semantic segmentation. During the training, we resize the input images to 224 × 224 and test on the full resolution 256 × 512.

Performance metrics

To evaluate the network compression and testing performance, the following measures are applied:

Acc.: The accuracy of testing on image classification. Acc. ↓ (%) is the accuracy drop between pruned and the baseline models. The smaller, the better. For CIFAR-10, top-1 accuracy is provided, while for ILSVRC-2012, both top-1 and top-5 accuracies are reported.

FLOPs: The overall floating point operations (FLOPs) is used as an indicator of computation costs. We use FLOPs ↓ (%) to describe the percentage of reduced FLOPs.

Pixel Acc.: Pixel Accuracy (Pixel Acc) on semantic segmentation. The higher, the better. It is defined as follows:

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \qquad (9)$$

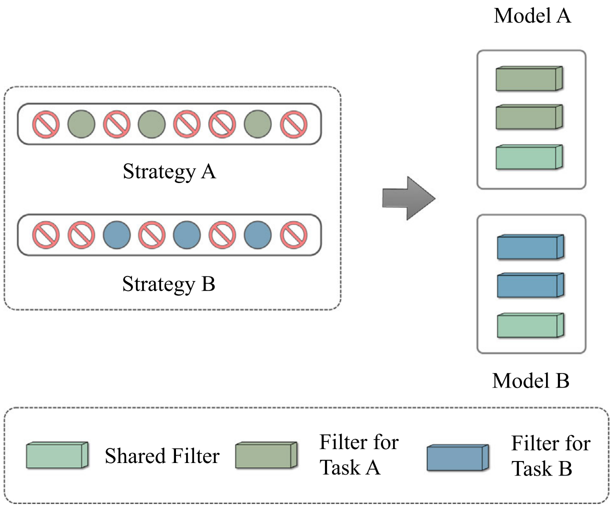


Fig. 7 An illustration of DCP-A in multi-model pruning. The policy logit will be widened in multi-model pruning case

where p_{ij} means the number of pixels belonging to ith class but predicted to be in jth class; k is the number of classes.

mIoU: Mean Intersection over Union (mIoU) on semantic segmentation. The higher, the better. It is defined as follows:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \qquad (10)$$

$\Delta\mathcal{T}$: Following [36,44], a single relative performance with respect to the baseline is defined for semantic segmentation of multiple metrics M as follows:

$$\Delta\mathcal{T} = \frac{1}{|M|} \sum_{j=0}^{|M|} (M_{\mathcal{T},j} - M_{baseline,j}) / M_{baseline,j} \qquad (11)$$

where $|M|$ represents the number of metrics.

Network architecture

We mainly focus on pruning ResNet [11] which has less redundancy than VGG-net [43]. An illustration of pruned MobileNet structure has also been provided in "Pruned result visualization" section.

Training setting

For image classification, we train the parameters of network and attention blocks with optimizer (Stochastic Gradient Descent algorithm, SGD), initial learning rate (0.1), momentum (0.9), batch size (256) and weight decay (0.0005). Following [44,53], Adam is used for optimizing policy logit and the constant learning rate is set to be 0.01. τ is initial-

Table 1 Different guidance on DCP

Depth	Method	Acc. (%)	Acc.↓ (%)	FLOPs↓ (%)
56	DCP-WOL	93.12	0.47	53.2
	DCP-L2-norm	93.28	0.31	53.4
	DCP-L1-norm	93.34	0.25	53.1
	DCP-A	**93.56**	**0.03**	**53.9**

Bold indicates the best results of compressed models under similar compression ratios

ized as 5 and then decayed to near 0. The loss constraint weights λ_1 and λ_2 are both set to be 0.5. On CIFAR, the network is trained for 50 epochs to learn the policy logit and the value is 10 for ImageNet. After training, we can obtain the optimal policy logit of network. Then, we prune the network according to the limit on FLOPs. Attention blocks will be removed from the pruned network, hence they will not increase the FLOPs. Pruned models will be trained for 200 epochs on CIFAR. Pre-trained model is used on ImageNet and the total epoch is 100. Baseline training schedule follows [14]. The learning rate is divided by 5× at epoch 60, 120 and 160.

For segmentation, the learning rate of network parameters is set to be constant (0.001) with weight decay (0.0001), batch size (8), training epoch (50) for optimizing and training epoch (50) for warm-up. λ_1 and λ_2 are set to be 0.01 and 0.1, respectively. The total re-training epoch is 300. τ is also initialized as 5.

At training time, we randomly split the original training database into two sub-training databases for two stages.

We compare DCP-A with existing state-of-art pruning algorithms, namely, MIL [7], PFEC [21], SFP [13], FPGM [14], TAS [9], MetaPruning [30], ChannelSelection [15], LSTM-SEP [6] and MFIS [4]. Among them, there exist NAS-based methods as well as hierarchically pruning methods with criterion.

Different guidance

In DCP-A, attention score provides layer information for optimizing policy logit. For comparison, we test DCP with another layer information calculation as well as without layer information. As exhibited in Table 1, DCP-WOL represents performing DCP without layer information. DCP-L1-norm and DCP-L2-norm describe replacing attention score S_l with the l_1 norm and l_2 norm of weights, respectively. The results show that layer information has a positive impact on facilitating network performance (DCP-L1-norm, DCP-L2-norm versus DCP-WOL). Moreover, DCP with attention (DCP-A) performs best because the attention mechanism learns better layer information.

Table 2 Comparison of pruning ResNet on CIFAR-10

Depth	Method	Acc. (%)	Acc. ↓ (%)	FLOPs ↓ (%)
32	SFP [13]	92.08	0.55	41.5
	MFIS [4]	**92.45**	**0.18**	41.5
	Ours	92.43	0.20	**47.0**
	TAS [9]	93.16	0.73	49.4
	LFPC [12]	92.12	0.51	52.6
	FPGM [14]	**92.82**	**−0.19**	53.2
	MFIS [4]	92.14	0.49	53.2
	Ours	92.37	0.26	**55.4**
56	PFEC [21]	91.31	1.75	27.6
	Ours	**93.58**	**0.01**	**44.7**
	CS [15]	93.31	0.40	50.0
	SFP [13]	92.26	1.33	52.6
	FPGM [14]	92.89	0.70	52.6
	MFIS [4]	93.27	0.32	52.6
	TAS [9]	93.69	0.77	52.7
	LFPC [12]	93.34	0.25	52.9
	Ours	**93.56**	**0.03**	**53.9**
110	PFEC [21]	92.94	0.61	38.6
	SFP [13]	93.38	0.30	40.8
	Ours	**94.20**	**−0.52**	**42.7**
	FPGM [14]	93.85	−0.17	52.3
	MFIS [4]	94.01	−0.33	52.3
	TAS [9]	94.33	0.64	53.0
	LFPC [12]	93.79	−0.11	**60.3**
	Ours	**94.24**	**−0.56**	55.5

Bold indicates the best results of compressed models under similar compression ratios

Table 3 Comparison of pruning ResNet on CIFAR-100

Depth	Method	Acc. (%)	Acc.↓ (%)	FLOPs↓ (%)
56	MIL [7]	68.37	2.96	39.3
	Ours	**71.30**	**0.11**	**40.0**
	TAS [9]	72.25	0.93	51.3
	LFPC [12]	70.83	0.58	51.6
	SFP [13]	68.79	2.61	**52.6**
	FPGM [14]	69.66	1.75	**52.6**
	Ours	**71.07**	**0.34**	**52.6**

Bold indicates the best results of compressed models under similar compression ratios

Pruning on CIFAR-10

ResNet has a special design for CIFAR that contains basic blocks, while we use the same policy logit for layers in the same block as mentioned.

We test DCP-A for ResNet with depth 32, 56, 110 on CIFAR-10 and compare the results with state-of-the-art

Table 4 Comparison of pruning ResNet-18 on NYU-v2

Model	FLOPs↓ (%)	Semantic segmentation		
		(Higher better)		
		mIoU	Pixel Acc	$\Delta\mathcal{T}(\%)$
Uniform baseline	–	26.6	57.9	–
Uniform baseline	49.7	25.0	57.0	− 3.8
Ours	**50.7**	**27.1**	**58.6**	**+ 1.5**
Uniform baseline	**60.2**	26.0	**59.0**	− 0.2
Ours	**60.2**	**26.6**	58.4	**+ 0.4**
Uniform baseline	69.7	25.7	57.3	− 2.2
Ours	**70.0**	**26.2**	**58.7**	**− 0.1**

Bold indicates the best results of compressed models under similar compression ratios

methods in Table 2. Moreover, we choose FLOPs reduced of 45% and 55% for our methods. The experiment results validate the effectiveness of the developed method, where DCP-A achieves better performances with more FLOPs reduced in almost all situations. Specifically, for depth 56, our method shows the results of 0.03% accuracy drop which is better than 0.25% of LFPC, whereas the acceleration rate of our method is 1.0% higher than that of MFIS. Likewise, better results and higher FLOPs drop occur in other depths by adopting our proposed method. For depth 32, MFIS performs slightly better (0.02%) while DCP-A earns more FLOPs reduction (6.5%). For depth 110, although LFPC reduces more FLOPs (4.8%), DCP-A can improve the performance with 0.56% compared of 0.11% accuracy improvement by LFPC.

Pruning on CIFAR-100

We also provide similar experiments on CIFAR-100 with ResNet-56 and show the results in Table 3. It can be seen that DCP-A can achieve better results than other methods

Table 5 Comparison of pruning ResNet-50 on ILSVRC-2012

Depth	Method	Baseline Top-1 Acc. (%)	Baseline Top-5 Acc. (%)	Pruned Top-1 Acc. (%)	Pruned Top-5 Acc. (%)	Top-1 Acc.↓ (%)	Top-5 Acc.↓ (%)	FLOPs↓ (%)
50	SFP [13]	76.15	92.87	62.14	84.60	14.01	8.27	41.8
	LSTM-SEP [6]	76.12	93.00	–	–	0.90	**0.27**	43.0
	TAS [9]	–	–	76.20	93.07	1.26	0.48	43.5
	Ours	76.15	92.87	75.66	92.51	**0.49**	0.36	**50.9**
	MetaPruning [30]	76.60	–	75.40	–	1.20	–	51.2
	CS [15]	76.13	–	75.56	–	**0.56**	**0.36**	51.3
	FPGM [14]	76.15	92.87	74.83	92.32	1.32	0.55	53.5
	MFIS [4]	76.15	92.87	75.23	92.50	0.92	0.37	53.5
	LFPC [12]	76.15	92.87	74.46	92.04	1.69	0.83	60.8
	Ours	76.15	92.87	74.61	92.18	1.54	0.69	**60.9**

Bold indicates the best results of compressed models under similar compression ratios

Table 6 Comparison of pruning ResNet on CIFAR-10 and CIFAR-100

Depth	Method	Acc. (%)		Acc. ↓ (%)			FLOPs ↓ (%)	
		CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	ALL	CIFAR-10	CIFAR-100
32	FPGM [14]	92.75	–	0.43	–	–	41.5	–
		–	69.44	–	0.84	–	–	41.5
	MFIS [4]	**92.96**	70.12	**0.22**	0.16	0.19	41.5	41.5
	Ours	91.02	**75.47**	1.61	**−5.19**	**−1.79**	**42.6**	**42.1**
56	FPGM [14]	93.55	–	0.21	–	–	52.6	–
		–	70.51	–	1.28	–	–	52.6
	MFIS [4]	**93.60**	71.16	**0.16**	0.63	0.40	52.6	52.6
	Ours	91.67	**76.78**	1.92	**−4.52**	**−1.30**	**54.4**	**54.6**
110	FPGM [14]	**94.22**	–	**−0.17**	–	–	52.3	–
		–	72.80	–	1.08	–	–	**52.3**
	MFIS [4]	**94.22**	73.04	**−0.17**	0.84	0.34	52.3	**52.3**
	Ours	92.50	**76.48**	1.18	**−2.60**	**−0.71**	**55.9**	51.5

Bold indicates the best results of compressed models under similar compression ratios

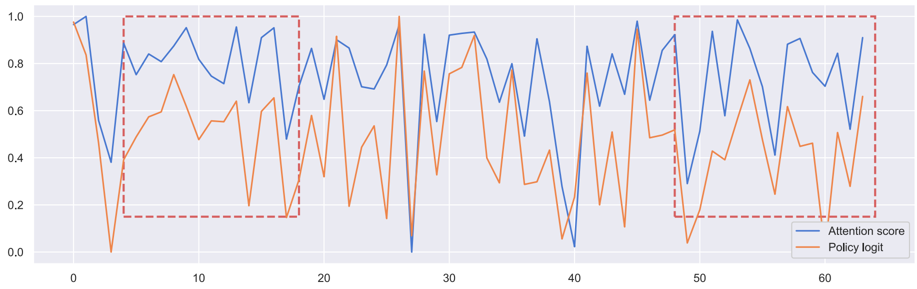


Fig. 8 An illustration of attention score and policy logit in the same layer. The value of both are normalized into [0, 1]. We can see that two lines maintain a similar trend for channels, which can be obviously observed in dash boxes

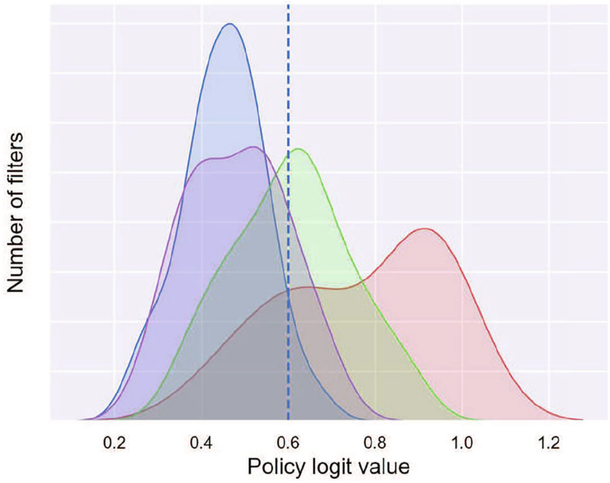


Fig. 9 An illustration of learned policy logit distributions for different layers. Compared to Fig. 4, if a threshold is set to be 0.6, each block will prune a proper number of filters

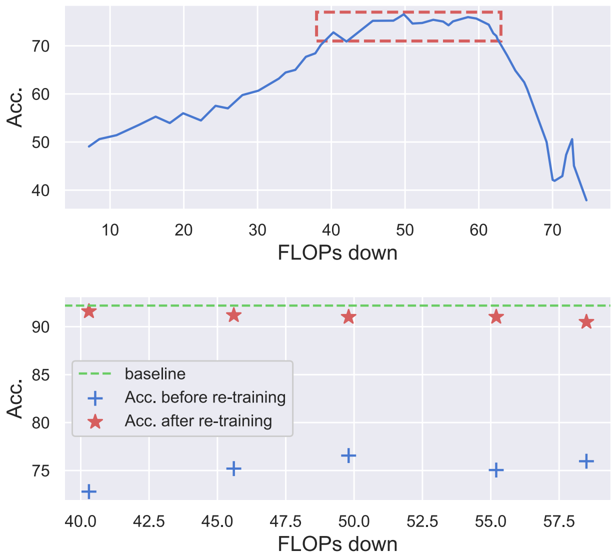


Fig. 10 Accuracies after training and re-training with different FLOPs limitations

under both 40% and 50% FLOPs reduced limitations, which also validates the effectiveness of our method.

Pruning on semantic segmentation

We test DCP-A for semantic segmentation application on NYU-v2 database. The Deeplab-ResNet [2] with atrous convolution is used as a baseline network. For comparison, we apply uniform pruning with different FLOPs limitation as uniform baselines. DCP-A also outperforms the uniform baselines on semantic segmentation as shown in Table 4.

Pruning on ImageNet

The proposed framework is then tested on ILSVRC-2012 with ResNet-50. ResNet-50 has a standard bottleneck block and we use the same policy for the input and middle layer, and a new policy for the output layer. The results are described in Table 5 and compared with state-of-the-art methods.

Pruning in multi-model

Finally, the proposed framework is tested in multi-database case (CIFAR-10 and CIFAR-100) and compared with FPGM and MFIS. Note that FPGM is performed on CIFAR-10 and CIFAR-100 separately. MFIS is a multi-task pruning method and the multi-task pruning results are adopted for comparison. As shown in Table 6, DCP-A performs better on CIFAR-100 while MFIS gives better results on CIFAR-10. Specifically, for depth 32, although MFIS shows the best result of accuracy drop 0.22% on CIFAR-10, DCP-A can improve the performance of 5.19% on CIFAR-100 which is much better than MFIS and FPGM. 'ALL' shows the average accuracy decline on all databases. We can see that the proposed DCP-A achieves the best results on all depths.

Pruned result visualization

Our approach designs the pruned network automatically and experiments on several databases have proved the effectiveness of DCP-A. Next, we are interested in the learned pruning results. Here, details of pruning results are exhibited to further exploit our method in the following.

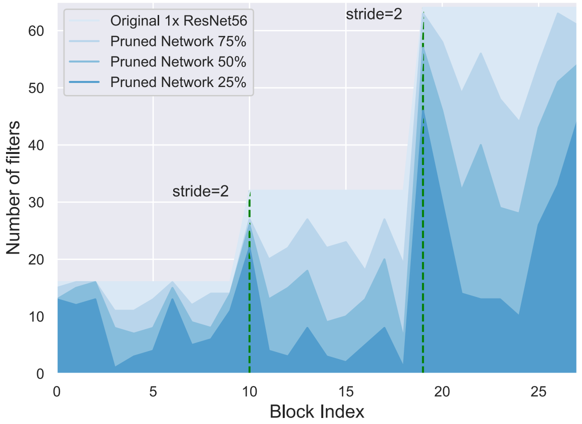


Fig. 11 An illustration of pruned ResNet-56 structure

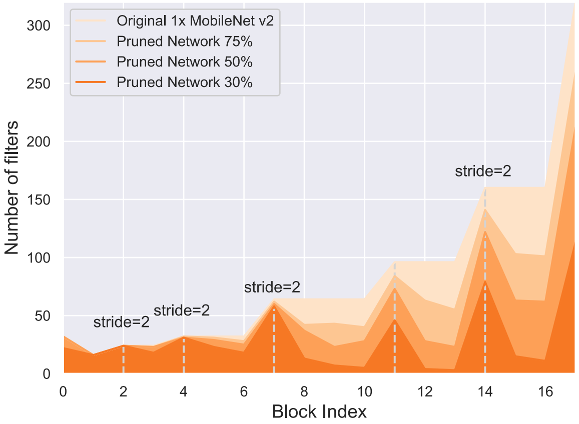


Fig. 12 An illustration of pruned MobileNet v2 structure

Attention score guidance

Figure 8 shows the attention score and policy logit in one layer, where the values of both are normalized into [0, 1] to exhibit the variation tendency clearly. We can see that the two lines maintain a similar trend for channels, which can be obviously observed in the dash boxes. The pruned filters in the same layer will be similar when using the attention score or policy logit as pruning criterion. Hence, the decision of pruning filters has been affected by the attention score-guided loss. This means that attention score acts as a guidance for optimizing policy logit in DCP-A training.

Policy logits

Figure 4 illustrates why we do not directly use attention score as a pruning criterion. Here, for comparison, an illustration of policy logit distributions has been presented in Fig. 9 with the same layers. Apparently, policy logit can be utilized for pruning. For example, if we set the threshold to be 0.6, then each layer will prune proper filters under this constraint.

Pruned results with different limitations

In the following, we will prove that DCP-A does not require repeated optimizing processes of policy logits under different FLOPs constraints. As exhibited in Fig. 10, the top figure shows the accuracies of pruned networks before re-training with different FLOPs constraints. To verify that the pruned network structures can lead to good performances, we retrain 5 pruned networks under FLOPs reduced constraints from 40% to 60%, and show the results in the bottom figure. It can be seen that all the pruned networks achieve acceptable performances with different limitations.

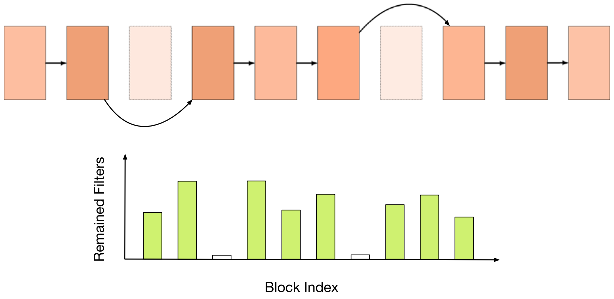


Fig. 13 An illustration of pruned ResNet-20 structure. The 3rd block and 7th block are removed according to the pruning policy

Pruned network structure

The pruned network structures for ResNet-56 and MobileNet v2 [40] under different FLOPs limitations have been exhibited in Figs. 11 and 12. We can observe that significant peeks exist in the pruned network, when there is a down-sampling operation with a stride 2 depth-wise convolution. Such a phenomenon also occurs in MetaPruning [30] when pruning MobileNet, which is mainly because network tries to make up for the loss of information caused by the resolution degradation in the feature map size. Hence, it proves that our DCP-A can learn an explainable policy for network architecture.

Skip block

There exist skipping blocks when pruning network is shown in Fig. 13. According to the learned pruning policy, all filters in the 3rd block and 7th block will be pruned and only the shortcut will be reserved. This equals skipping the 3rd block and 7th block. Hence, DCP-A can shrink the network structure not only in width but also in-depth when pruning the network with a shortcut.

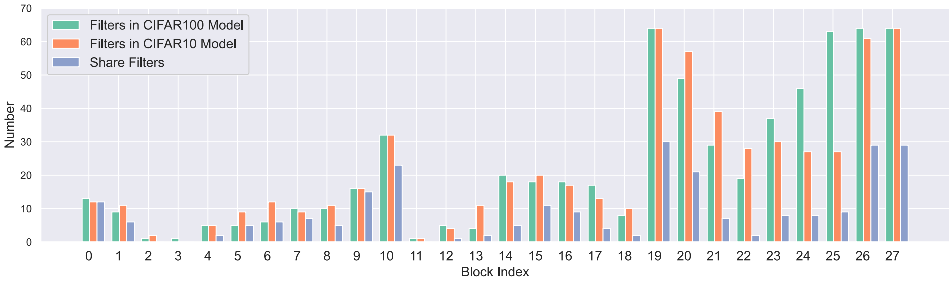


Fig. 14 Pruning results of ResNet-56 on CIFAR-10 and CIFAR-100

Multi-model pruning results

Figure 14 illustrates multi-model pruning results of ResNet-56 on CIFAR-10 and CIFAR-100. The number of shared filters is counted as well as the number of filters in a single model. DCP-A can adaptively design the shared structure of models. Specifically, there exists no filter sharing in the 2nd, 3rd and 11th module.

Conclusion

In this paper, a new differentiable channel pruning framework guided via an attention mechanism has been proposed and verified with experiments. Attention mechanism has been adopted as a guidance to provide layer information for policy optimization. The training process is differentiable with Gumbel-softmax sampling and a two-stage training procedure has been proposed to optimize the network parameters, policy logit and attention modules alternately. Special design has been provided for network blocks with shortcut and showed that protecting shortcut can assist DCP-A prune the network not only in width but also in depth. Detailed analysis has been given with pruned model visualization. Limitations also exist in the proposed method. More guidance mechanisms can be considered in addition to attention guidance. Moreover, DCP-A can be extended into multi-task pruning. In the future, we will 1) consider more different guidance mechanisms with layer information [32,46,47,49,56,57,66], 2) introduce control strategies to enhance the model robustness [3,27,33,50,51,58], and 3) extend our approach to other complicated multi-task learning problems [1,23,34,59,65,67].

Acknowledgements The Deanship of Scientific Research (DSR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia has funded this project, under grant no. (RG-2-611-43). This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 820776 (INTEGRADDE), the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Bao G, Ma L, Yi X (2022) Recent advances on cooperative control of heterogeneous multi-agent systems subject to constraints: a survey. Syst Sci Control Eng 10(1):539–551
2. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
3. Chen Y, Ma K, Dong R (2022) Dynamic anti-windup design for linear systems with time-varying state delay and input saturations. Int J Syst Sci 53(10):2165–2179
4. Cheng H, Wang Z, Wei Z, Ma L, Liu X (2021) Multi-task pruning via filter index sharing: a many-objective optimization approach. Cogn Comput 13:1070–1084
5. Cheng H, Wang Z, Wei Z, Ma L, Liu X (2022) On adaptive learning framework for deep weighted sparse autoencoder: a multiobjective evolutionary algorithm. IEEE Trans Cybern 52(5):3221–3231
6. 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
7. Dong X, Chen S, Pan SJ (2017) Learning to prune deep neural networks via layer-wise optimal brain surgeon. In: Advances in neural information processing systems (NIPS), pp 4857–4867
8. Dong X, Huang J, Yang Y, Yan S (2017) More is less: a more complicated network with less inference complexity. In: Conference on computer vision and pattern recognition (CVPR), Jul 2017
9. Dong X, Yang Y (2019) Network pruning via transformable architecture search. In: Advances in neural information processing systems (NIPS), pp 759–770
10. Guo Y, Yao A, Chen Y (2016) Dynamic network surgery for efficient dnns. In: Advances in neural information processing systems (NIPS), pp 1379–1387
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Conference on computer vision and pattern recognition (CVPR), Jun 2016

12. He Y, Ding Y, Liu P, Zhu L, Zhang H, Yang Y (2020) Learning filter pruning criteria for deep convolutional neural networks acceleration. In: Conference on computer vision and pattern recognition (CVPR), Jun 2020
13. He Y, Kang G, Dong X, Fu Y, Yang Y (2018) Soft filter pruning for accelerating deep convolutional neural networks. In: Proceedings of international joint conference on artificial intelligence (IJCAI), Jul 2018
14. He Y, Liu P, Wang Z, Hu Z, Yang Y (2019) Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Conference on computer vision and pattern recognition (CVPR), Jun 2019
15. Herrmann C, Bowen RS, Zabih R (2020) Channel selection using gumbel softmax. In: Computer vision–ECCV, pp 241–257
16. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 42(8):2011–2023
17. Jang E, Gu S, Poole B (2017) Categorical reparameterization with gumbel-softmax. In: International conference on learning representations (ICLR)
18. Ji D, Wang C, Li J, Dong H (2021) A review: data driven-based fault diagnosis and RUL prediction of petroleum machinery and equipment. Syst Sci Control Eng 9(1):724–747
19. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
20. Kusupati A, Ramanujan V, Somani R, Wortsman M, Jain P, Kakade SM, Farhadi A (2020) Soft threshold weight reparameterization for learnable sparsity. In: Proceedings of international conference on machine learning (ICML), vol 119, pp 5544–5555
21. Li H, Kadav A, Durdanovic I, Samet H, Graf HP (2017) Pruning filters for efficient convnets. In: International conference on learning representations (ICLR)
22. Li W, Niu Y, Cao Z (2022) Event-triggered sliding mode control for multi-agent systems subject to channel fading. Int J Syst Sci 53(6):1233–1244
23. Li X, Song Q, Liu Y, Alsaadi FE (2022) Nash equilibrium and bang-bang property for the non-zero-sum differential game of multi-player uncertain systems with Hurwicz criterion. Int J Syst Sci 53(10):2207–2218
24. Li X, Song Q, Zhao Z, Liu Y, Alsaadi FE (2022) Optimal control and zero-sum differential game for Hurwicz model considering singular systems with multifactor and uncertainty. Int J Syst Sci 53(7):1416–1435
25. Li X, Zhou Y, Pan Z, Feng J (2019) Partial order pruning: For best speed/accuracy trade-off in neural architecture search. In: Conference on computer vision and pattern recognition (CVPR), Jun 2019
26. Li Y, Gu S, Zhang K, Gool LV, Timofte R (2020) DHP: differentiable meta pruning via HyperNetworks. In: Computer vision–ECCV, pp 608–624
27. Li Z, Hu J, Li J (2021) Distributed filtering for delayed nonlinear system with random sensor saturation: a dynamic event-triggered approach. Syst Sci Control Eng 9(1):440–454
28. Lin M, Ji R, Zhang Y, Zhang B, Wu Y, Tian Y (2020) Channel pruning via automatic structure search. In: Proceedings of international joint conference on artificial intelligence (IJCAI), Jul 2020
29. Liu H, Simonyan K, Yang Y (2019) DARTS: differentiable architecture search. In: International conference on learning representations (ICLR)
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: International conference on computer vision (ICCV), Oct 2019
31. Lu P, Song B, Xu L (2021) Human face recognition based on convolutional neural network and augmented dataset. Syst Sci Control Eng 9(s2):29–37
32. Luo X, Wu H, Wang Z, Wang J, Meng D (2022) A novel approach to large-scale dynamically weighted directed network representation. IEEE Trans Pattern Anal Mach Intell 44(12):9756–9773
33. Luo X, Yuan Y, Chen S, Zeng N, Wang Z (2022) Position-transitional particle swarm optimization-incorporated latent factor analysis. IEEE Trans Knowl Data Eng 34(8):3958–3970
34. Luo X, Wu H (2022) Li Z NeuLFT: a novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2022.3176466
35. Lyu K, Li Y, Zhang Z (2020) Attention-aware multi-task convolutional neural networks. IEEE Trans Image Process 29:1867–1878
36. Maninis K-K, Radosavovic I, Kokkinos I (2019) Attentive single-tasking of multiple tasks. In: Conference on computer vision and pattern recognition (CVPR), Jun 2019
37. Paszke A, Gross S, Chintala S, Chanan G, Yang E, Devito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
38. Qu F, Zhao X, Wang X, Tian E (2022) Probabilistic-constrained distributed fusion filtering for a class of time-varying systems over sensor networks: a torus-event-triggering mechanism. Int J Syst Sci 53(6):1288–1297
39. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
40. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Conference on computer vision and pattern recognition (CVPR), Jun 2018
41. Shakiba FM, Shojaee M, Azizi SM, Zhou M (2022) Real-time sensing and fault diagnosis for transmission lines. Int J Netw Dyn Intell 1(1):36–47
42. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from RGBD images. In: Computer vision–ECCV, vol 7576, pp 746–760
43. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations (ICLR)
44. Sun X, Panda R, Feris R, Saenko K (2020) Adashare: learning what to share for efficient deep multi-task learning. In: Advances in neural information processing systems (NIPS)
45. Szankin M, Kwasniewska A (2022) Can AI see bias in X-ray images? Int J Netw Dyn Intell 1(1):48–64
46. Su Y, Cai H, Huang J (2022) The cooperative output regulation by the distributed observer approach. Int J Netw Dyn Intell 1(1):20–35
47. Tao H, Tan H, Chen Q, Liu H, Hu J (2022) H_∞ state estimation for memristive neural networks with randomly occurring DoS attacks. Syst Sci Control Eng 10(1):154–165
48. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems (NIPS), pp 5998–6008
49. Wang L, Liu S, Zhang Y, Ding D, Yi X (2022) Non-fragile l_2-l_∞ state estimation for time-delayed artificial neural networks: an adaptive event-triggered approach. Int J Syst Sci 53(10):2247–2259
50. Wang M, Wang H, Zheng H (2022) A mini review of node centrality metrics in biological networks. Int J Netw Dyn Intell 1(1):99–110
51. Wang X, Sun Y, Ding D (2022) Adaptive dynamic programming for networked control systems under communication constraints: a survey of trends and techniques. Int J Netw Dyn Intell 1(1):85–98
52. Wang XJ, Yao W, Fu H (2019) A convolutional neural network pruning method based on attention mechanism. In: Proceedings of international conference on software engineering and knowledge engineering, Jul 2019

53. Wu B, Keutzer K, Dai X, Zhang P, Wang Y, Sun F, Wu Y, Tian Y, Vajda P, Jia Y (2021) FBNet: hardware-aware efficient ConvNet design via differentiable neural architecture search. In: Conference on computer vision and pattern recognition (CVPR), Jun 2019. IEEE Transactions on Circuits and Systems for Video Technology 31(2):512–522
54. Xie S, Zheng H, Liu C, Lin L (2019) SNAS: stochastic neural architecture search. In: International conference on learning representations (ICLR)
55. Xu L, Song B, Cao M (2021) A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint. Syst Sci Control Eng 9(1):138–149
56. Yang J, Ma L, Chen Y, Yi X (2022) L_2-L_∞ state estimation for continuous stochastic delayed neural networks via memory event-triggering strategy. Int J Syst Sci 53(13):2742–2757
57. Yao F, Ding Y, Hong S, Yang S-H (2022) A survey on evolved LoRa-based communication technologies for emerging internet of things applications. Int J Netw Dyn Intell 1(1):4–19
58. Yu H, Hu J, Song B, Liu H, Yi X (2022) Resilient energy-to-peak filtering for linear parameter-varying systems under random access protocol. Int J Syst Sci 53(11):2421–2436
59. Yu L, Cui Y, Liu Y, Alotaibi ND, Alsaadi FE (2022) Sampled-based consensus of multi-agent systems with bounded distributed time-delays and dynamic quantisation effects. Int J Syst Sci 53(11):2390–2406
60. Yu N, Yang R, Huang M (2022) Deep common spatial pattern based motor imagery classification with improved objective function. Int J Netw Dyn Intell 1(1):73–84
61. 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: Conference on computer vision and pattern recognition (CVPR), Jun 2018
62. Yuan Y, Ma G, Cheng C, Zhou B, Zhao H, Zhang H-T, Ding H (2020) A general end-to-end diagnosis framework for manufacturing systems. Natl Sci Rev 7(2):418–429
63. Yuan Y, Tang X, Zhou W, Pan W, Li X, Zhang H-T, Ding H, Goncalves J (2019) Data driven discovery of cyber physical systems. Nat Commun 10(1):1–9
64. Yuan Y, Zhang H, Wu Y, Zhu T, Ding H (2016) Bayesian learning-based model-predictive vibration control for thin-walled workpiece machining processes. IEEE/ASME Trans Mechatron 22(1):509–520
65. Zhang Q, Zhou Y (2022) Recent advances in non-Gaussian stochastic systems control theory and its applications. Int J Netw Dyn Intell 1(1):111–119
66. Zhao G, Li Y, Xu Q (2022) From emotion AI to cognitive AI. Int J Netw Dyn Intell 1(1):65–72
67. Zhao Y, He X, Ma L, Liu H (2022) Unbiasedness-constrained least squares state estimation for time-varying systems with missing measurements under round-robin protocol. Int J Syst Sci 53(9):1925–1941
68. Zoph B, Le QV (2017) Neural architecture search with reinforcement learning. In: International conference on learning representations (ICLR)

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