SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2025, Vol. 68, Iss. 1, 112102:1–112102:12
https://doi.org/10.1007/s11432-022-3951-y

Channel pruning on frequency response

Hang LIN[1], Yifan PENG[2], Lin BIE[1], Chenggang YAN[2], Xibin ZHAO[1*] & Yue GAO[1*]

[1]*School of Software, Tsinghua University, Beijing 100084, China;*
[2]*School of Automation, Hangzhou Dianzi University, Hangzhou 310012, China*

Received 17 October 2022/Revised 30 August 2023/Accepted 13 October 2023/Published online 19 December 2024

Abstract Network pruning has a significant role in reducing network parameters and accelerating the inference time of the network. Some existing methods prune the network based on the frequency of the data, and finally obtain a sub-network with high accuracy. However, according to our experimental analysis, different frequencies of information in the data contribute differently to the accuracy of the model, and using this information directly for pruning without making a selection will lead to incorrect results. We believe that pruning should retain the convolutional kernels in the network that process important information, while those kernels that process unimportant information should be removed. In this paper, we first investigate the meaning of each frequency band information in the spectrum and their contribution to the prediction accuracy of the network, and according to these results, we propose a new pruning method based on frequency response (PFR). Our PFR finds and removes the convolutional kernels in the network that specialize in processing unimportant information, resulting in a compact neural network model. PFR obtains significant experimental results on different datasets, for example, a 56.0% raduction of float points operations (FLOPs) on ResNet-50 and only 0.37% of Top-1 accuracy degradation on the ImageNet dataset.

Keywords deep learning, model compression, filter pruning, channel pruning, frequency response

Citation Lin H, Peng Y F, Bie L, et al. Channel pruning on frequency response. Sci China Inf Sci, 2025, 68(1): 112102, https://doi.org/10.1007/s11432-022-3951-y

1 Introduction

Convolutional neural networks (CNNs) have demonstrated their superiority in various fields. However, to achieve better performance in various tasks, the CNNs become increasingly complex in structure. Especially with the increasing depth and width of the network, there is a higher demand for computing resources, greater energy consumption, and more space storage capacity. Therefore, existing CNN models are still difficult to deploy directly on resource-constrained hardware platforms (e.g., Internet of Things (IoT) devices) due to the huge number of parameters and high computational costs. However, although the parameters of a model can express its complexity to a certain extent, related studies have shown that not all parameters contribute to the inference of network models, and some of them have limited effects, express redundancy, and even degrade the performance of the model. So it is necessary to find the redundant parameters in the model and effectively prune the model and minimize the loss of accuracy for existing models, which is beneficial to the landing of deep learning. In recent years, numerous neural network compression techniques have been proposed, including quantization [1–3], network pruning [4–16], low-rank decomposition [17–19], and knowledge distillation [20]. Among the categories mentioned above, our proposed method focuses on channel pruning, a subclass of structured network pruning.

Channel pruning, which belongs to network pruning, can effectively reduce network parameters and float points operations (FLOPs). We focus on developing a novel compression method for channel pruning in this paper. When it comes to pruning, in general, the core problem is how to efficiently prune the model and minimize the loss of accuracy. Existing pruning methods fall into two main categories. One of the strategies is to set numerous neural network parameters to zero by gradient descent, which is used to reduce the complexity of the neural network model [4, 20, 21]. Finally, the performance of the sparse model is restored by fine-tuning. However, these methods only focus on training the network parameters to 0 and can be simply discarded from the model. They fail to investigate the role of the network model parameters themselves by removing the parameters that have low contribution to the model. Another

* Corresponding author (email: zxb@tsinghua.edu.cn, gaoyue@tsinghua.edu.cn)

strategy is to use the intrinsic properties of the network for pruning to determine the importance of the neural network parameters, and restore the performance of the network model by fine-tuning after pruning [5–8, 12, 13]. We also adopt this strategy. Unlike the above articles, this paper is the first work to consider the effect of training data information on the model parameters themselves.

It is a consensus that deep neural networks (DNN) is highly data-dependent. For channel pruning, data-dependent methods are commonly used to update the pruning strategy and evaluate the performance of the pruned model. Clearly, the quality of the data set determines the performance of the network. However, based on our experimental analysis, the information in images is usually redundant and this redundant information plays only a small role in improving the performance of the network. Intuitively, from a frequency-domain perspective, low-frequency information in images plays a more important role in the network's ability to accurately identify and classify targets compared to high-frequency information. Accordingly, due to the redundancy of information, the neural network needs to use more weights to learn this information which is not critical to the performance improvement of the network. This situation has resulted in a redundant network structure. If we can reduce the redundancy of the information in the image, and accordingly, we remove the corresponding filters of the network that deal with the redundant information, then we may prune the network safely with less loss of model performance. In other words, for a network, we only need to retain the part of the network that is dedicated to processing critical information (low-frequency information). To increase the performance of the network (generalization ability, robustness), we usually do some data augmentation. We consider the appropriate use of processed data to improve the performance of the pruned network. That is, combining enhanced data for pruning will result in better performance of the pruned network structure. In this paper, we propose a pruning method based on data augmentation. We first augment the data in the frequency domain, then feed the processed data into a pre-trained model, and prune it according to the activations in the network. In this paper, we propose the concept of a response value, according to which we can know how well a channel is adapted to the information at each frequency. After ranking the information at different frequencies according to their contribution to the accuracy of the network, we retain those channels that are adapted to important information to complete the pruning.

In summary, our main contributions are as follows.

• We investigated the contribution of different frequency band information of the input data to the network prediction, and propose the concept of response value to measure the degree of adaptation of the channels in the network to different frequency data. To our best knowledge, this is the first paper to consider the frequency response on channels.

• We propose a novel channel pruning method for CNNs based on the frequency response of the channel in the network, which enables the pruned network to retain the channels that contribute more to the model prediction.

• Extensive experimental results show that our method can achieve the superior performance with CNNS (ResNet and MobileNetV2) on two benchmarks, including ImageNet [22] and CIFAR-10 [23].

2 Related work

2.1 Original data-driven filter pruning

Most of the existing pruning methods directly use the original input data to develop pruning strategies to guide network model pruning. Ref. [5] found that feature maps generated by the same kernel have the same rank and proved by singular value decomposition that the larger the rank is, the more information it contains. Finally, the channels can be ranked in order of importance according to the rank and then pruned. Ref. [24] argued that if a subset of the feature map replaces the original set and makes the error between the output of the intermediate layer and the output of the original as small as possible, then the feature map outside this subset can be considered unimportant, and then the corresponding filter can be pruned. Refs. [6, 8] pruned the network based on the loss function obtained from the input data. There are also studies that use the gradients obtained after sampling the mini-batch data for pruning, such as [25]. Although the above data-driven pruning-based methods can minimize the model performance loss at a higher compression rate, it does not take into account the existence of useless information in the original input data, which can affect the pruning performance.

2.2 Frequency-based work

Some previous studies [26, 27] illustrated that CNN from the frequency domain has advantages that are not available in the spatial domain, such as reduced time and increased efficiency. Ref. [28] investigated the relationship between the frequency spectrum of image data and the generalization behavior of convolutional neural networks. Ref. [10] proposed a Fourier convolutional neural network (FCNN) that is trained entirely in the Fourier domain. The training time has been significantly improved without reducing the training efficiency. In the spatial to frequency-domain transform, not only the Fourier transform but also the more advanced wavelet transform is available, which helps us to better understand the features of convolutional neural network models with different perspectives.

Frequency-domain information plays a huge role in CNN, and it is no exception in pruning work. Ref. [29] proposed a frequency-domain dynamic pruning scheme for spatial correlation, dynamically pruning the frequency-domain coefficients in each iteration, and selecting different thresholds for different frequency bands. Ref. [30] combined frequency and redundancy through a hash function to compress parameters in a frequency-sensitive manner, thus better retaining important model parameters. Ref. [31] reduced the computational burden of convolution operations in CNN by linearly combining the convolution responses of the discrete cosine transform bases. In contrast to the aforementioned approaches, our proposed method uniquely integrates frequency-domain insights from training data into the network's architecture, thereby enabling targeted pruning guided by the magnitudes of these contributions.

3 Proposed method

3.1 Analysis for different frequency bands in data

Compared to the spatial domain, conducting data processing in the frequency domain implies less information loss, probably because operations in the spatial domain are pixel-based while operating on the spectrum implies a smaller granularity [32]. Refs. [28, 33] argued that the components of different frequency bands in the frequency domain have different meanings and roles. In the spectrum, we can easily manipulate the frequencies of specific frequency bands to achieve fine-grained image processing. Generally, it is difficult to represent noisy or useless features in the spatial domain for the features of the data. For example, Ref. [24] claimed to find and retain a subset of the input feature map of a layer and expected a minimal impact on the output of that layer, thus pruning the corresponding kernels. However, the problem is that we cannot equate minimizing the impact with being more beneficial to the whole pruning task. If we prune the model while ensuring that the retained filters are the most contributing ones to the task, it will undoubtedly lead to better performance of the pruned network. Ref. [5] suggested that filters with low-rank feature maps are less informative and thus less crucial for maintaining accuracy, and can therefore be removed first. However, we argue that the quantity of information is not equal to the quality of information. In other words, one channel that contains a lot of useless or redundant information is also unimportant for the network. In the frequency domain, each frequency band in the frequency domain performs its role, for example, the low frequency component mainly represents the outline of the object, the slightly higher frequency represents the detailed information, and the higher frequency represents the noise. Inspired by this information, we believe that removing the noise, redundant and useless information, feeding only the most representative data into the network, and then pruning the network with these representative data will undoubtedly improve the performance of the pruning method. Therefore, we explored the effect of signals of different frequencies on the classification performance of the network. Our experimental design is shown in Figure 1(a). First, we divide the spectrum of an image into a number of equally distant parts and perform zero-setting operations on these parts in turn; then we transform these parts into the spatial domain using the inverse Fourier transform. Now we get a set of images missing different frequency information from a single image. The experiments are shown in Figure 1(b). By inputting the processed images into the pre-trained model for inference, we obtain the accuracy of the network for data with different frequency band information removed. In this experimental setup, we conducted two distinct sets of experiments. In the first set, we partitioned the spectral information of the images into three equal segments, and the results are depicted in the left segment of Figure 1(b). In the second set of experiments, we partitioned the spectral information of the images into four equal segments, and the results are illustrated in the right segment of Figure 1(b). Higher accuracy means missing that frequency band information has less impact on accuracy. In other words, the missing

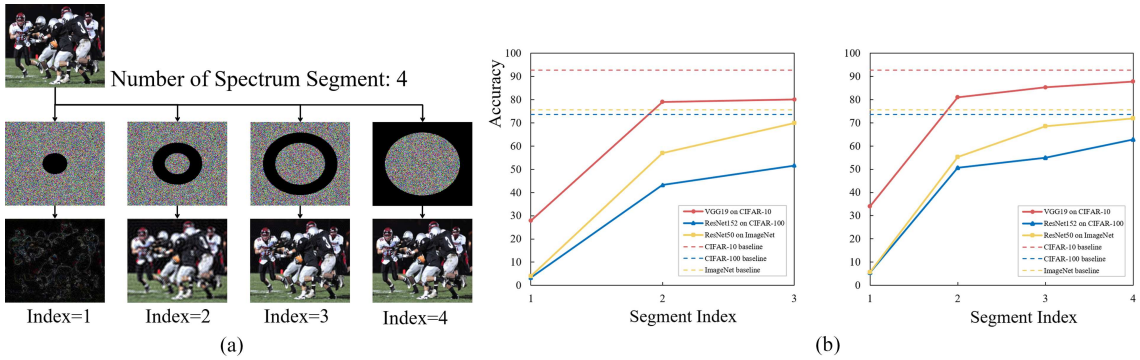

Figure 1 (Color online) Compare the importance of different frequency bands for networks. (a) Experimental design for different frequency bands. We randomly select a batch of images from the dataset, perform fast Fourier transform (FFT) and frequency shift operations on the images to obtain the frequency spectrum of each image, then use the ideal filter formula to process the frequency spectrum of these images, perform zero-setting operations on each component in turn (the zero-setting operations are indicated in black in the figure), and obtain the processed images after the inverse fast Fourier transform (IFFT). (b) The accuracy of different frequency bands. The x-axis represents the index of the missing frequency bands in the data, where a smaller index means a lower frequency of the missing bands, and the y-axis represents the accuracy. We process the data from CIFAR-10, CIFAR-100, and ImageNet datasets respectively and test their accuracy on different networks. The dotted line indicates the accuracy of the unprocessed data in the corresponding network.

part contributes very little to the accuracy, and we consider that the band with greater contribution is more important for the accuracy of the network. From these experiments, as depicted in Figure 1(a), we observed a significant performance degradation when low-frequency information is absent. Conversely, the absence of high-frequency information has a comparatively minor impact on model performance. Therefore, we can conclude that the importance of low-frequency band information is much higher than that of high-frequency information for accuracy.

It is well known that high-frequency information in the spectrum represents noise, but whether high-frequency noise can be equated with redundant information still needs to be demonstrated experimentally. We conducted a study related to redundant information in the frequency domain, and we wanted to find more discriminative features. As shown in Figure 2, we randomly selected several groups of similar images in the Imagenet dataset [22], and we can see that the similarity of the same class of images is more similar as there are fewer high-frequency components in the spectrum of the data. Therefore, we believe that this experiment strongly suggests that the more representative features of the same classification are more often indicated by the low-frequency information of the data. From the experimental results, we can see that as more and more high-frequency information is removed, the similarity of different images of the same class is increasing, which strongly suggests that high-frequency information can be treated as unimportant or redundant compared to low-frequency information in the target classification task. Accordingly, in our approach, we consider high-frequency information to be relatively redundant.

3.2 Pruning on frequency response

A large number of filters are connected in different forms to form a deep neural network, and through experimentation we have been able to clarify the different contributions of different information in the frequency domain to the network. The effect of the different information in the frequency domain on the network is eventually reflected in each filter. As presented in Figure 3, in step 1, we process the data in the frequency domain to obtain data containing different frequency information, and in step 2, we input the data we obtained, use back propagation to obtain the response values of different kernels, and then prune the network according to the response values.

3.2.1 *Data-processing in frequency domain*

In this step, we extract useful information and remove redundant components from images by processing images in the frequency domain. Unlike Subsection 3.1 where we processed the data with the ideal filter formula, we use (1) to extract the low-frequency information. We will demonstrate in Subsection 4.3.2 that this change results in less loss of useful information, where x and y are the distance of points in the spectrum relative to the center of the spectrum in the x-axis direction and the distance in the y-axis direction. By changing the σ we can change the high-frequency content of the spectrum while keeping

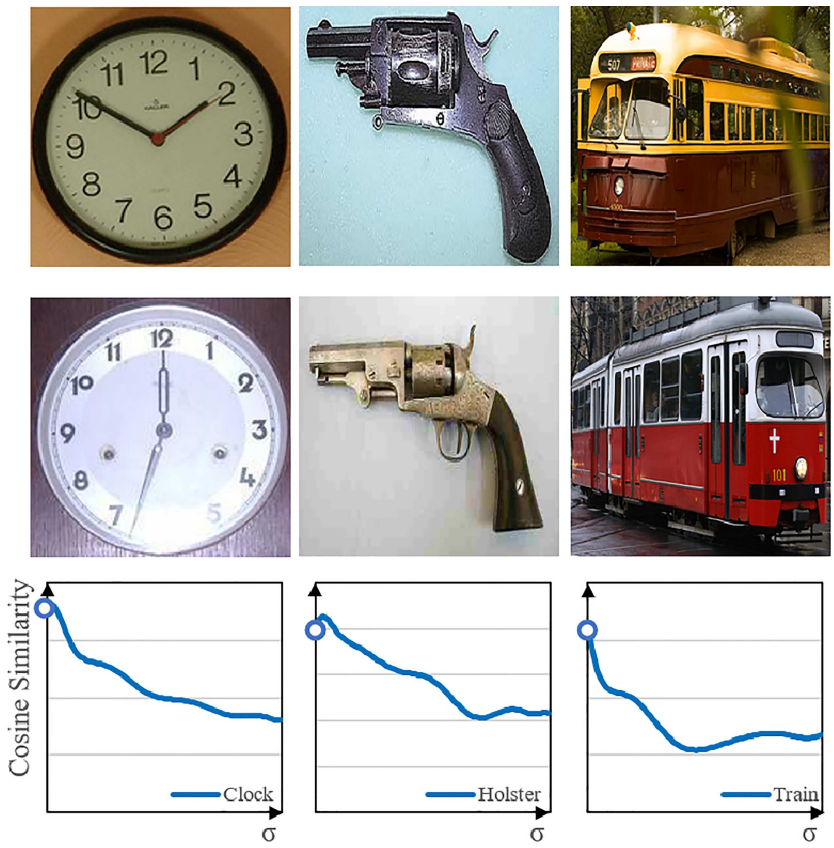

Figure 2 (Color online) Cosine similarity of the sampled images when σ takes different values. We perform low-pass filtering on images in the frequency domain, where the proportion of high-frequency components gradually increases as the σ parameter is raised. We then use histogram similarity to determine the similarity between the two images.

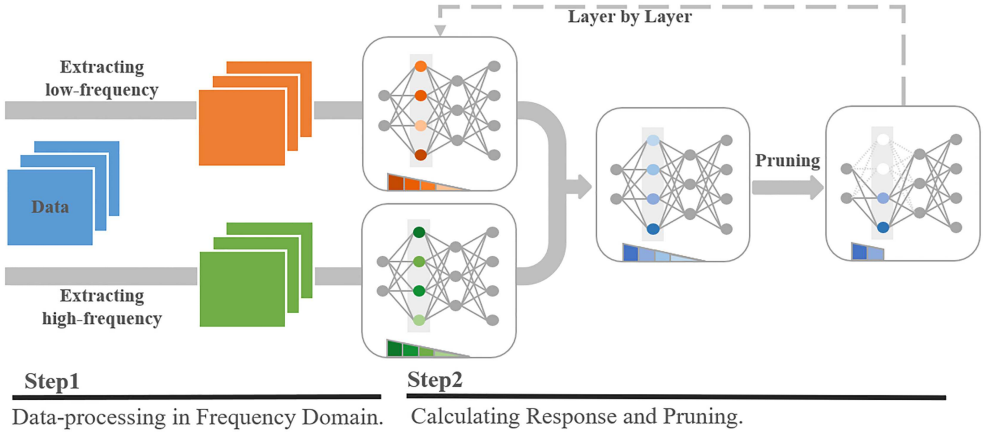

Figure 3 (Color online) Pipeline of our proposed method. In Step 1, we use orange to represent the low frequency information extracted from the image and green to represent the high frequency information. In Step 2, we take the information extracted in Step 1 and put it into the network for back-propagation to obtain the response of each channel in each layer of the network to the input. Step 2 shows the response of each channel at a layer to the input, and according to the wedge color card, we specify that a darker color represents a larger response. The response of the high and low frequency inputs is calculated according to the response formula (6) to obtain the final response. We can see that after pruning the network, the removed channels become white in one layer.

the low-frequency component as much as possible. Similarly, we use (2) to extract the high-frequency information of the spectrum. For convenience, let $G_{\text{low}}\{d\}$ and $G_{\text{high}}\{d\}$ denote the images obtained after processing the data d in the frequency domain by (1) and (2). In other words, $G_{\text{low}}\{d\}$ has many losses of high-frequency information relative to d and $G_{\text{high}}\{d\}$ has many losses of low-frequency information

relative to d,

$$G_{\text{low}}(x,y) = \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}},\ \sigma \in \mathbb{R}, \tag{1}$$

$$G_{\text{high}}(x,y) = 1 - \mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}},\ \sigma \in \mathbb{R}. \tag{2}$$

3.2.2 *Response-based channel selection and pruning*

We first introduce the concept of sensitivity, because of the loss of weight w, the magnitude of the change in loss L is called the sensitivity of w, where the sensitivity of each w in the network can be expressed as follows:

$$\Delta L(w,d) = |\mathcal{L}(\mathcal{F}(W,d),y) - \mathcal{L}(\mathcal{F}(W_{w\to 0},d),y)|, \tag{3}$$

where W is the set of w in the whole network, d is the sampled data of a batch, y is the ground truth, $\mathcal{L}$ is the loss function, and $\mathcal{F}$ is the Forward function. We consider that the change in loss occurs because the information in the data that should have been processed by w cannot be processed because w is removed. Taking inspiration from [25], they approximate the change in loss as the absolute value of the gradient change in the indicator m. m can be thought of as a mask operation on w. They subtly relate the change in loss function after removing weight w_i to the gradient of mask m_i corresponding to w_i at the time of sampling. The magnitude of the m_i gradient is used to express the importance of w_i and can be expressed as follows:

$$\begin{aligned}\Delta L_{w_i}(W,d) \approx g_i(W,d) &= \left.\frac{\partial L(M\odot W,d)}{\partial m_i}\right|_{M=\mathbf{1}} \\ &= \lim_{\delta\to 0}\left.\frac{|L(W,d) - L(W_{m_i\to m_i-\delta},d)|}{\delta}\right|_{M=\mathbf{1}},\end{aligned} \tag{4}$$

where W is the set of w in the whole network, d is the sampled data of a batch, and M is the set of m corresponding to each w. For this formula, we can understand that the removal of w causes the W not to adapt to d, so in the back-propagation calculation, the mask m of weight w generates a gradient prompting w to update in the direction of adapting to d. The larger the absolute value of the gradient, the more w is not adapted to d. Therefore, we suppose that if the less w is adapted to low-frequency data, then the larger the gradient will be obtained after sampling with $G_{\text{low}}\{d\}$ and performing back propagation calculation. We take the L1-norm of the mask corresponding to the gradient of all the weights in the kernel as the response value of that kernel to the input data d. The response value is calculated as follows:

$$\begin{aligned}R_{l,j}[d] = \sum |\Delta L_{w_i}(W,d)|, \\ \text{s.t.}\quad w_i \in k_{l,j},\end{aligned} \tag{5}$$

where $R_{l,j}[\cdot]$ is the response of the j-th kernel at layer l of the network, $k_{l,j}$ is the j-th kernel at layer l of the network, and d is the sampled data of a batch. So $R_{l,j}[d]$ is an indicator for us to determine whether $k_{l,j}$ is suitable for data d. That is, we represent the response of a convolution kernel k to d in terms of the $L1$ norm of all weight gradients in the kernel. We aim to find kernels in the pre-trained model that are best adapted to low-frequency and not sensitive to high-frequency noise. So we define the response formula for each kernel as follows:

$$S_{l,j}(d) = R_{l,j}[G_{\text{high}}\{d\}] - R_{l,j}[G_{\text{low}}\{d\}], \tag{6}$$

where $S_{l,j}(d)$ denotes the response taken for the j-th kernel at layer l of the network.

In our approach, we use the frequency-domain processed data to perform forward-propagation on the pre-trained model, calculate the gradients in the network by back-propagation but without parameter updates, and then obtain the response values for each convolutional kernel based on the data dominated by low-frequency information and high-frequency information.

Eq. (6) consists of two parts. In the first term of the equation, the magnitude of this term reflects the response value of the kernel to low-frequency information since there is missing low-frequency information

in $G_{\text{high}}\{d\}$. The smaller the response value, the less the kernel is adapted to low-frequency information, and the more it is adapted to high-frequency information. We expect the first term to be as large as possible because it follows from the previous experiments that the information in the low-frequency band is the component that contributes more to all frequency components. In the second term of the equation, the sign of the second term is negative because we want to remove the kernel that responds to high-frequency noise. Likewise, we want the R as small as possible. In order to extract useful information and remove redundant components, the larger the value of response is, the better. The larger the response, the more important the kernel is. In layer-by-layer pruning, we keep the kernel with the large response and remove the kernel with a small response according to the pruning rate.

4 Experiments

4.1 Experimental settings

Datasets and models. To establish the efficacy of our proposed approach, we conducted comprehensive experiments employing diverse models on two datasets (CIFAR-10 [23] and ImageNet [22]). The models used to validate the method include ResNet with a residual block and MobileNetV2. On CIFAR-10, we evaluate our method on ResNet-56, ResNet-110 [34], and MobileNetV2. On the ImageNet dataset, we evaluate our method on ResNet50. For all benchmarks and architectures, we randomly sample 128 images to estimate the response values to low-frequency information and high-frequency information.

Evaluation metrics. As in previous work, we select the number of parameters in the network and the required FLOPs to evaluate the model size and the consumption of arithmetic power. To evaluate the performance of our approach on different tasks, we provide Top-1 accuracy of pruned models on CIFAR-10 and provide Top-1 accuracy and ΔTop-1 of pruned models on ImageNet.

Configurations. The pre-trained models we used are from [5]. We use a layer-by-layer fine-tuning method to recover the accuracy of the pruned model. We fine-tune 30 epochs on the CIFAR-10 with the learning rate being divided by 10 at epochs 10 and 15, and fine-tune 30 epochs on the ImageNet with the learning rate divided by 10 every 10 epochs. After the layer-by-layer pruning, we perform a general fine-tuning. For fine-tuning on CIFAR-10, we use the stochastic gradient descent (SGD) with a momentum of 0.9 and weight decay is 5E−4, and the initial learning rate, batch size, and training epochs are set to 0.01, 128, and 30. For fine-tuning on ImageNet, we use the SGD with a momentum of 0.9 and weight decay is 5E−4, and the initial learning rate, batch size, and training epochs are set to 0.01, 128, and 30. All experiments are implemented on multiple NVIDIA RTX 2080 SUPER GPUs by PyTorch. For CIFAR-10, the images are padded to size 40×40 and then cropped to size 32×32. For ImageNet, images with a resolution 224×224 are sent to the networks. For data processing in the frequency domain, on the CIFAR-10 dataset, we have σ equal to 5 when extracting high-frequency information and low-frequency information. On the ImageNet dataset, we have σ equal to 10 when extracting high-frequency information and low-frequency information.

4.2 Results and analysis

4.2.1 *Results on CIFAR-10*

We analyze the performance of ResNet-56, ResNet-110, and MobileNetV2 on CIFAR-10.

ResNet-56/ResNet-110. We compared the results for ResNet-56 in Table 1 [4, 5, 15, 16, 35–37] and for ResNet-110 in Table 2 [4, 5, 35]. Tables 1 and 2 are sorted by the size of the parameters from top to bottom. We start with ResNet-56. The accuracy of our chosen baseline is 93.26%. Compared to the three methods GAL-0.6, L1, and HRank, our method achieves higher accuracy with lower parameters, and similar FLOPs (94.12% vs. 92.98%, 93.06%, 93.85%). The model after pruning is 0.86% higher than the baseline (94.12% vs. 93.26%). In addition our method achieves 93.96% accuracy for a model with parameters of 0.47 M, outperforming CC ($C = 0.4$) (93.96% vs. 93.87%), NISP (93.96% vs. 93.01%) and HRank with smaller parameters at similar FLOPs (93.96% vs. 93.57%) methods. It is worth noting that even when both the number of parameters and the FLOPs reduction rate exceed 70%, our method can still maintain an accuracy of 93% in the pruned model.

Then we analyzed ResNet-110. The accuracy of our chosen baseline is 93.50%. Our method outperforms the above two methods in terms of accuracy with a parameter of 0.73M and FLOPs smaller than L1 and GAL-0.5 (93.92% vs. 93.3% and 92.55%). Although parameters were slightly larger than HRank,

Table 1 Pruning results for ResNet-56 on CIFAR-10.

Model	Top-1 (%)	FLOPs (PR)	Parameter (PR)
ResNet-56	93.26	125.49M (0.0%)	0.85M (0.0%)
GAL-0.6 [4]	92.98	78.30M (37.6%)	0.75M (11.8%)
L1 [35]	93.06	90.90M (27.6%)	0.73M (14.1%)
HRank [5]	93.85	90.35M (28.0%)	0.66M (22.3%)
PFR (ours)	94.12	88.17M (29.7%)	0.64M (24.7%)
CC ($C = 0.4$) [36]	93.87	72.00M (42.4%)	0.54M (36.5%)
NISP [37]	93.01	81.00M (35.5%)	0.49M (42.4%)
HRank [5]	93.57	65.94M (47.4%)	0.48M (42.8%)
PFR (ours)	93.96	71.05M (43.4%)	0.47M (44.7%)
CC ($C = 0.5$) [36]	93.64	60.00M (52.0%)	0.44M (48.2%)
ABCPruner-70% [16]	93.23	57.47M (54.2%)	0.39M (54.1%)
RL-MCTS [15]	93.56	56.00M (55.0%)	–
PFR (ours)	93.00	36.96M (70.5%)	0.24M (70.0%)

Table 2 Pruning results for ResNet-110 on CIFAR-10.

Model	Top-1 (%)	FLOPs (PR)	Parameter (PR)
ResNet-110	93.50	252.89M (0.0%)	1.72M (0.0%)
L1 [35]	93.30	155.00M (38.7%)	1.16M (32.6%)
GAL-0.5 [4]	92.55	130.20M (48.5%)	0.95M (44.8%)
PFR (ours)	93.92	88.65M (64.9%)	0.73M (57.6%)
HRank [5]	93.81	101.97M (59.6%)	0.72M (58.1%)
PFR (ours)	93.85	84.64M (66.5%)	0.67M (61.0%)
PFR (ours)	93.61	76.63M (69.7%)	0.54M (68.6%)
HRank [5]	93.23	71.69M (71.6%)	0.54M (68.3%)
PFR (ours)	93.68	71.05M (71.9%)	0.47M (72.7%)

Table 3 Pruning results for MobileNetV2 on CIFAR-10.

Model	Base Top-1 (%)	Pruned Top-1 (%)	ΔTop-1 (%)	FLOPs (↓) (%)
WM [38]	94.47	94.02	−0.45	26.0
DCP [39]	94.47	94.69	+0.22	26.0
PFR (ours)	94.17	94.72	+0.55	48.0

a higher accuracy was achieved with smaller FLOPs (88.65M vs. 101.97M, 93.92% vs. 93.81%). Our method exceeds HRank at 0.67M with both parameters and FLOPs larger than our method (93.85% vs. 93.81%). Also at 0.54M, our method exceeds HRank for accuracy (93.61% vs. 93.23%). With a 72.7% decrease in parameters and a 71.9% decrease in FLOPs, our accuracy is still higher than baseline (93.68% vs. 93.50%).

MobileNetV2. MobileNetV2 is a computationally efficient model, which makes it harder to prune. We compared the results for MobileNetV2 in Table 3 [38, 39]. With the highest pruning rate (48%), our method outperforms the recent state-of-the-art methods. The results on MobileNetV2 show that our proposed method can improve the pruned models for both parameter-heavy models and computation-efficient models.

4.2.2 *Results on ImageNet*

We compare the results for ResNet50 in Table 4 [4, 5, 15, 16, 21, 36, 40–45], where we present the Top-1 accuracy after pruning and the ΔTop-1 in the model relative to the baseline accuracy after pruning. Because of computational resource constraints, we performed only two sets of experiments on ResNet50. The accuracy of our chosen baseline is 76.01%. For GAL (71.95%, 43.0%), HRank (74.98%, 43.8%), DSA (74.69%, 50.0%), CC ($C = 0.5$) (75.59%, 52.9%), SCP (75.27%, 54.4%), Hinge (74.70%, 54.4%), and C-SGD-50 (74.54%, 55.8%), we have outperformed them all with smaller FLOPs (56.0%) and larger Top-1 accuracies (75.64%). For both methods, SRR-GR (75.76%) and Liu et al. (76.42%), although the accuracy of Top-1 after pruning is higher than our method, we believe it is related to the accuracy of baseline (76.13%, 76.79%), and in terms of ΔTop-1, with smaller FLOPs (56.0% vs. 44.1%, 50.3%),

Table 4 Pruning performance (%) for ResNet-50 on ImageNet.

Model	Base Top-1	Pruned Top-1	ΔTop-1	Base Top-5	Pruned Top-5	ΔTop-5	FLOPs (↓)
GAL [4]	76.15	71.95	−4.20	92.87	90.94	−0.54	43.0
HRank [5]	76.15	74.98	−1.17	92.87	92.33	−1.17	43.8
SRR-GR [40]	76.13	75.76	−0.37	92.86	92.67	−0.19	44.1
DSA [41]	76.15	74.69	−1.33	92.87	92.45	−0.80	50.0
Liu et al. [21]	76.79	76.42	−0.37	–	–	–	50.3
CC ($C = 0.5$) [36]	76.15	75.59	−0.56	92.87	92.64	−0.23	52.9
SCP [42]	76.15	75.27	−0.62	92.87	92.30	−0.68	54.4
Hinge [43]	76.15	74.70	−1.40	–	–	–	54.4
RL-MCTS [15]	77.34	76.46	−0.88	93.37	92.83	−0.34	55.0
SRR-GR [40]	76.13	75.11	−1.02	92.86	92.35	−0.51	55.1
C-SGD-50 [44]	75.33	74.54	−0.79	92.56	92.09	−0.47	55.8
PFR (ours)	76.01	75.64	−0.37	92.87	92.60	−0.27	56.0
GReg-2 [45]	76.13	75.36	−0.77	–	–	–	56.8
ABCPruner-100 [16]	76.01	74.84	−1.61	–	–	–	59.7
HRank [5]	76.15	71.98	−4.17	92.87	92.61	−0.26	62.3
CC ($C = 0.6$) [36]	76.15	74.54	−1.61	92.87	92.25	−0.62	62.7
PFR (ours)	76.01	74.40	−1.61	92.87	92.53	−0.34	66.1

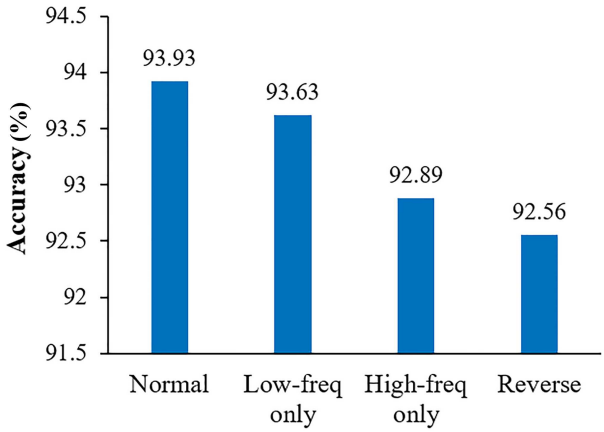

Figure 4 (Color online) Accuracy of different response function.

our method achieves the same ΔTop-1 (−0.37% vs. −0.37%, −0.37%) as in their method. For GReg-2, we obtained greater accuracy (75.64% vs. 75.36%) for comparable FLOPs. For larger pruning rates, our method achieves higher accuracy (74.40% vs. 71.98%) than HRank with smaller FLOPs (66.1% vs. 62.3%). When compared with CC ($C = 0.6$), our method achieves almost the same ΔTop-1 (−1.61% vs. −1.61%) with smaller FLOPs (66.1% vs. 62.7%). Based on the data presented in Table 4, it is evident that at a pruning rate of 56%, our pruned model achieves a superior ΔTop-1 accuracy compared to the state-of-the-art methods. Furthermore, there is a minimal decrease of only 0.26% in the Top-5 accuracy, which also approaches the optimal performance. Experiments demonstrate that our PFR can achieve better performance and acceleration at higher compression rates.

4.3 Ablation study

4.3.1 *Different response formulas*

In the response formula we use two parts of the data, the data processed by the low-pass filtering formula and processed by the high-pass filtering formula. As a comparison, we are performing three sets of experiments: (1) We only low-pass filter the data in the frequency domain and let the response formula disregard the response value of the high-frequency information; (2) we only high-pass filter the data in the frequency domain and let the response formula disregard the response value of the low-frequency information; (3) we multiplied the response formula by −1. The experiments were all performed with ResNet-56 on CIFAR-10, and the final accuracy is shown in Figure 4.

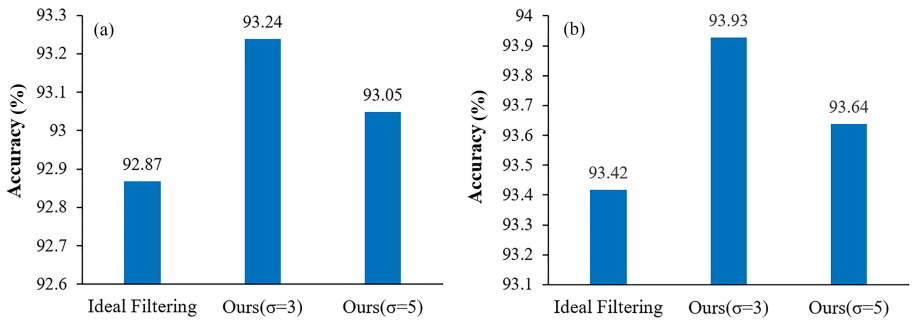

Figure 5 (Color online) Impact of filtering formulas on the accuracy. (a) VGGNet on the CIFAR-10; (b) ResNet-56 on the CIFAR-10.

Among all the variant methods, including the "Low-freq only", "High-freq only", and "Reverse" approaches, the "Low-freq only" method exhibits the highest performance. The analysis shows that although the high-frequency part of the data is removed, it retains the low-frequency part that contains relatively more information. Moreover, our method clearly outperforms its variants, while Reverse performs the worst, which proves the effectiveness of our method.

4.3.2 *Our filtering formula vs. ideal filtering formula*

The mid-frequency band in the spectrum is a relatively neglected part, but we believe that neglecting the mid-frequency band causes a loss of information. In our filtering equations (1) and (2), the formula value gradually changes as the center point is moved away (the low-pass filtering formula gradually decreases and the high-pass filtering formula gradually increases), that means some middle-frequency band has been retained in addition to the low-frequency band. The ideal filtering formula corresponds to a very sharp change, which directly sets the frequency outside the cutoff frequency to zero. We compared the impact of the pruned model on accuracy after processing with these two formulas in the frequency domain in Figure 5.

In Figures 5(a) and (b), we have done three sets of experiments on VGGNet [44] and ResNet-56, respectively. In the first group we use the ideal filtering formula as the frequency-domain processing formula, and in the second and third groups we use the frequency-domain filtering formula we designed to process the image in the frequency domain. For VGGNet, in the frequency-domain processing, using our method will obtain higher accuracy than using ideal filtering (93.24%, 93.05% vs. 92.87%). Similar results were observed for ResNet-56 (93.93%, 93.64% vs. 93.42%).

This experimental result demonstrates that the information in the mid-frequency band also plays a role in contributing to the accuracy and the effectiveness of our frequency-domain filtering formula. We believe that the reason for this result is that the ideal filtering formula directly discards the frequency information beyond the cutoff-frequency, which leads to a decrease in the accuracy rate. Moreover, the number of operations in the frequency domain is the same for both formulas.

5 Conclusion

In this paper, we investigate the role and meaning of each frequency band in the spectrum of an image. Based on the experimental results we propose a frequency-based pruning method, which achieves good results on both the CIFAR-10 dataset and the ImageNet dataset. We first take out the information of each frequency band of the data in the frequency domain using the ideal filtering formula and then return it to the spatial domain to investigate the contribution of each frequency band to the accuracy rate. It turns out that the contribution of the low-frequency band information is much larger than that of the high-frequency band. We then conducted experiments to find whether the low-frequency information or the high-frequency information can represent more essential features. The experiments show that for two very similar images, less high-frequency information can make the images more similar, i.e., the low-frequency information can represent more essential features. Based on our findings, we propose a frequency-based pruning method. The method considers that in a pre-trained model, if the response of a channel is greater for low-frequency information, i.e., a greater change needs to be made to fit the low-frequency information, the importance of the kernel corresponding to this channel is lower. Compared with the

state-of-the-art methods, the pruned networks obtained by our method can achieve better performance with less computational cost. For example, our method can reduce 56.0% FLOPs of ResNet-50 with only 0.37% Top-1 accuracy degradation on ImageNet.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. U20A6003, 62076146, 62021002, U19A2062, U1911401, 6212780016), and Industrial Technology Infrastructure Public Service Platform Project, Ministry of Industry and Information Technology of China (Grant No. 2022-233-225).

References

1 Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2704–2713
2 Rastegari M, Ordonez V, Redmon J, et al. XNOR-Net: ImageNet classification using binary convolutional neural networks. In: Proceedings of European Conference on Computer Vision, 2016. 525–542
3 Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 28
4 Lin S, Ji R, Yan C, et al. Towards optimal structured CNN pruning via generative adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 2790–2799
5 Lin M, Ji R, Wang Y, et al. HRank: filter pruning using high-rank feature map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 1529–1538
6 Hanson S, Pratt L. Comparing biases for minimal network construction with back-propagation. In: Proceedings of Advances in Neural Information Processing Systems, 1988. 1
7 LeCun Y, Denker J, Solla S. Optimal brain damage. In: Proceedings of Advances in Neural Information Processing Systems, 1989. 2
8 Hassibi B, Stork D. Second order derivatives for network pruning: optimal brain surgeon. In: Proceedings of Advances in Neural Information Processing Systems, 1992. 5
9 Li B, Wu B, Su J, et al. EagleEye: fast sub-net evaluation for efficient neural network pruning. In: Proceedings of European Conference on Computer Vision, 2020. 639–654
10 Pratt H, Williams B, Coenen F, et al. FCNN: Fourier convolutional neural networks. In: Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2017. 786–798
11 Lin M, Zhang Y, Li Y, et al. $1 \times N$ pattern for pruning convolutional neural networks. IEEE Trans Pattern Anal Mach Intell, 2022, 45: 3999–4008
12 Lin M, Cao L, Li S, et al. Filter sketch for network pruning. IEEE Trans Neural Netw Learn Syst, 2021, 33: 7091–7100
13 Lin M, Cao L, Zhang Y, et al. Pruning networks with cross-layer ranking & k-reciprocal nearest filters. IEEE Trans Neural Netw Learn Syst, 2023, 34: 9139–9148
14 Zhang Y, Lin M, Lin Z, et al. Learning best combination for efficient N: M sparsity. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 941–953
15 Wang Z, Li C. Channel pruning via lookahead search guided reinforcement learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022. 2029–2040
16 Lin M, Ji R, Zhang Y, et al. Channel pruning via automatic structure search. In: Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence, 2021. 673–679
17 Yu X, Liu T, Wang X, et al. On compressing deep models by low rank and sparse decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017. 7370–7379
18 Phan A H, Sobolev K, Sozykin K, et al. Stable low-rank tensor decomposition for compression of convolutional neural network. In: Proceedings of European Conference on Computer Vision, 2020. 522–539
19 Pan Y, Xu J, Wang M, et al. Compressing recurrent neural networks with tensor ring for action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 4683–4690
20 Lin Y K, Wang C F, Chang C Y, et al. An efficient framework for counting pedestrians crossing a line using low-cost devices: the benefits of distilling the knowledge in a neural network. Multimed Tools Appl, 2021, 80: 4037–4051
21 Liu L, Zhang S, Kuang Z, et al. Group fisher pruning for practical network compression. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 7021–7032
22 Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
23 Krizhevsky A, Hinton G. Learning Multiple Layers of Features From Tiny Images. Technical Report, 2009
24 Luo J H, Wu J, Lin W. ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5058–5066
25 Lee N, Ajanthan T, Torr P H S. SNIP: single-shot network pruning based on connection sensitivity. 2018. ArXiv:1810.02340
26 Xu K, Qin M, Sun F, et al. Learning in the frequency domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 1740–1749
27 Gueguen L, Sergeev A, Kadlec B, et al. Faster neural networks straight from JPEG. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 31
28 Wang H, Wu X, Huang Z, et al. High-frequency component helps explain the generalization of convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 8684–8694
29 Liu Z, Xu J, Peng X, et al. Frequency-domain dynamic pruning for convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 31
30 Chen W, Wilson J, Tyree S, et al. Compressing convolutional neural networks in the frequency domain. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 1475–1484

31 Wang Y, Xu C, You S, et al. CNNpack: packing convolutional neural networks in the frequency domain. In: Proceedings of Advances in Neural Information Processing Systems, 2016. 29

32 Rippel O, Snoek J, Adams R P. Spectral representations for convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 28

33 Yin D, Lopes R G, Shlens J, et al. A Fourier perspective on model robustness in computer vision. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 32

34 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

35 Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient ConvNets. In: Proceedings of the 5th International Conference on Learning Representations, 2017

36 Li Y C, Lin S H, Liu J Z, et al. Towards compact CNNs via collaborative compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 6438–6447

37 Yu R, Li A, Chen C F, et al. NISP: pruning networks using neuron importance score propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 9194–9203

38 Howard A G, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. 2017. ArXiv:1704.04861

39 Zhuang Z, Tan M, Zhuang B, et al. Discrimination-aware channel pruning for deep neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2018. 31

40 Wang Z, Li C, Wang X. Convolutional neural network pruning with structural redundancy reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 14913–14922

41 Ning X, Zhao T, Li W, et al. DSA: more efficient budgeted pruning via differentiable sparsity allocation. In: Proceedings of European Conference on Computer Vision, 2020. 592–607

42 Kang M, Han B. Operation-aware soft channel pruning using differentiable masks. In: Proceedings of the 38th International Conference on Machine Learning, 2020. 5122–5132

43 Li Y, Gu S, Mayer C, et al. Group sparsity: the hinge between filter pruning and decomposition for network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 8018–8027

44 Ding X, Ding G, Guo Y, et al. Centripetal SGD for pruning very deep convolutional networks with complicated structure. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4943–4953

45 Wang H, Qin C, Zhang Y, et al. Neural pruning via growing regularization. In: Proceedings of the 9th International Conference on Learning Representations, 2021