Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230600

Study on pruning optimization based on HRank pruning method

Sheng Lu

School of Microelectronics, Fudan University

21307130089@m.fudan.edu.cn

Abstract. Convolutional neural network (CNN), as one of the most important pillars of deep learning technology, has been paid increased attention to by researchers. However, by the expansion of the application of convolutional neural network, it faces more and more complex problems and more and more diverse situations. In order to solve these problems more appropriately, the number of parameters of convolutional neural networks continuously increases. This limits its potential to be deployed to distal devices with relatively low computing power and memory space. To improve this situation, this paper studies the compression of convolutional neural network model with acceptable sacrifice of accuracy. Based on Pytorch, Resnet-56 on CIFAR-10 is pruned. By using HRank pruning method, convolution kernels of each convolution layers are ordered according to the size of their determinant rank. Through deleting low-rank, less importance convolution kernels and reserving high-rank, more important convolution kernels, the pruning purpose is achieved. By changing the default compression ratio of different convolution layer, more important ones are applied with lower compression rates and less important ones are applied with higher compression rates, the paper achieves higher compression rate with the acceptable loss of accuracy. At the same time, the paper tests the efficiency of pruning under different learning rates. Finally, the paper finds that when the preset compression rate of the intermediate convolutional layer is changed slightly, the accuracy maintains at the level of 92.720% but the compress rate of Params reaches to 44.7% with 0.47M left and the compress rate of Flops reaches 51.1% with 61.39M left. In addition, the paper finds that when the learning rate is 0.05, the learning efficiency reaches its optimum.

1. Introduction

With the iterative updating of computer hardware, neural network algorithms, which were once limited by computational power, are now flourishing and widely used in various fields, including picture recognition [1], intelligent driving [2], video repair [3] and so on. However, along with the increasing power of the algorithm and its ability to solve more complex problems and deal with more diverse situations, the amount of needed data is constantly increasing, and the amount of computation is also greatly increased. However, on edge devices, it is difficult to have enough power and memory to support this large amount of computing, thus neural network algorithms are difficult to be deployed locally on devices such as mobile phones and smart bracelets. To use neural networks, most of them need to be connected to cloud servers, which hinders more flexible application methods. At the same time, the pursuit of time efficiency has led to the creation of more sensitive and efficient devices to reduce the wasted time. Pruning optimization and algorithm scarification of deep learning algorithms are two

DOI: 10.54254/2755-2721/6/20230600

important and widely used algorithm improvement techniques that significantly reduce the memory and time required by the algorithm by appropriately sacrificing the accuracy of the algorithm, which have attracted extensive attention from researchers in this field.

Zeng Kai et al. applied L1 regularization to the scaling factor γ of the BN layer of YOLOv3 algorithm which has excellent performance in the industrial field, and then carried out scarification training. The number of model parameters and file size were compressed to 30% of the original, and the calculation time was reduced to 60% of the original, while only 0.01 mAP@0.5 was sacrificed [4]. Cheng-I Jeff Lai et al. realized 10.9%/12.6% WER decrease in subnetworks from wav2vec 2.0 through Prune-Adjust-Re-Prune method [5]. Jia HueiTan et al. found that the pruning schemes extended with encoder pruning gives considerable overall performance and verified it on Up-Down and Object Relation Transformer, achieving good results [6]. Nourhan Bayasi et al. apply multi-domain learning which allows for a fixed network to learn new data domains sequentially over time to CL model used in clinical implementations in patient-care settings, increasing its utility [7]. Mingbao Lin et al. verified that the contribution of low-rank convolution is relatively small. Meanwhile, they found that the average rank of each feature map in different layers is almost unchanged even if the number of image batch processing is different, and then proposed the HRank pruning method based on pruning low-rank convolution kernel. Good performance was achieved in Resnet-50 on CIFAR-10, Densenet-40 on CIFAR-10, Resnet-50 on ImageNet [8]. In May 2021, Mingbao Lin et al. further extended the HRank pruning method, proposed the HRankPlus pruning method and published it on Github, which further optimized the balance between pruning efficiency and accuracy, so that both could be maintained at a high level at the same time.

This paper first introduces the basic principle of pruning and expounds the application of pruning method based on the rank of convolution kernel as its feature. Then introduces residual block, one of the most important features of Resnet network, explaining its function and importance. Afterward the paper introduces the experimental environment and its content and studies the compression of convolutional neural network model with acceptable sacrifice of accuracy. By using HRank pruning method, convolution kernels of the convolution layers are ranked according to the size of their determinant rank ordering. To achieve the purpose of the pruning, HRank method deletes low-rank, less important convolution kernels. By changing the default compression ratio and giving higher compression rate to relatively less important convolution layer, the paper achieves higher compression rate under less loss of its accuracy. By testing the pruning performance under different preset compression rates, including Params, Flops and Accuracy, and transverse comparison of various parameters, the researcher selected the better parameters and better effects. At the same time, the pruning performance under different learning rates was compared, thus the paper obtained the available learning rate interval, using its accuracy as the evaluation standard. Finally, the experiment content is summarized, and the future research direction is discussed.

2. Method

2.1. The pruning basis and HRank method

Consider that a certain algorithm of CNN has N convolutional layers, Ci represents the i-th layer, in which there are mi convolution kernels. The set of convolution kernels in this layer is denoted as Mi, and w_j^i represents the j-th convolution kernel in the i-th convolutional layer. During pruning, the convolution kernels of each convolutional layer are divided into two sets, more important convolution kernels set I and less important convolution kernels set U. For example, for the Ci layer，$I_{C^i} = \{w^i_{I^1_{C^i}}, w^i_{I^2_{C^i}}, w^i_{I^3_{C^i}}, \dots, w^i_{I^{m_I^i}_{C^i}}\}$ represent the m_I^i-th important convolution kernels, while $U_{C^i} = \{w^i_{U^1_{C^i}}, w^i_{U^2_{C^i}}, w^i_{U^3_{C^i}}, \dots, w^i_{U^{m_U^i}_{C^i}}\}$ represent the m_U^i-th unimportant convolution kernels. Obviously，for m_I^i and m_U^i，they follow Equation (1).

DOI: 10.54254/2755-2721/6/20230600

$$m_I^i + m_U^i = m^i \tag{1}$$

For I_{C^i} and U_{C^i}, they follow Equation (2) and (3).

$$I_{C^i} \cup U_{C^i} = M^i \tag{2}$$

$$I_{C^i} \cap U_{C^i} = \emptyset \tag{3}$$

The pruning process of the i-th convolutional layer can be described as removing the most m_U^i-th unimportant convolution kernels of the i-th convolutional layer, and the pruning process of the whole algorithm is shown in Equation (4).

$$min \sum_{i=1}^{N} \sum_{j=1}^{m^i} \delta_{ij} L(w_j^i), \ s.t. \sum_{j=1}^{m^i} \delta_{ij} = m_U^i \tag{4}$$

Where $\delta_{ij} = 0$, when $w_j^i \in I_{C^i}$,and $\delta_{ij} = 1$, when $w_j^i \in U_{C^i}$. L is a measure of the set to which w_j^i belongs and is usually a different function in different pruning methods.

In CNN, each feature map generates a new feature map after passing the filter, where the filter is a matrix. When a certain row/column vector in a matrix can be linearly expressed by several other row/column vectors, these row/column vectors are described as linearly correlated; When any row/column vector in a set of row/column vectors cannot be linearly expressed by any other row/column vector, this set of row/column vectors are described as linearly independent. The vector group containing at most linearly independent row/column vectors of a set of row/column vectors is called the maximum linearly independent group of this set of row/column vectors, and the number of linearly independent row/column vectors it contains is called the rank of the vector group.

At the same time, since the row transformation of a matrix does not change the rank of the row vector group, nor does it change the rank of the column vector group, the row number of the identity matrix in the standard form can directly correspond to the rank of the row vector group of the matrix after the basic row transformation. Column vectors of matrix are in the same logic. Therefore, the number of the row vector of maximal linearly independent row vector group of the matrix contains and the number of the column vector of maximal linearly independent column vector group of the matrix contains are the same, all equaling the order of the identity matrix contained in the standard form of the matrix. That number is described as the rank of the matrix. The rank of a matrix can reflect the amount of effective information contained in the matrix. By sorting the filters in each convolutional layer according to their rank, the contribution of each filter to the result can be measured. The more rank a filter contains, the more valid information it contains, and the more impact it has on the result. On CIFAR-10, the rank of filter has little correlation with the training image batch, so the rank of each filter can be distinguished with fewer images. In each epoch, the filter containing less rank will be pruned, the filter containing moderate rank will be updated, and the filter containing more rank will be frozen, that is, entering the next epoch with no modification or minimal modification, as shown in Figure 1.

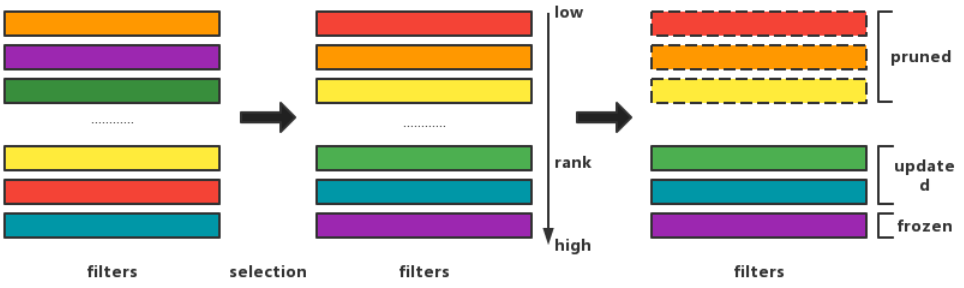

Figure 1. Schematic diagram of HRank pruning method.

Correspondingly, when the HRank pruning method is used, the function $L(w_j^i)$ in Equation (4) is replaced by $Rank(w_j^i)$. At the same time, the experiment shows that the Rank of filter is robust and has

DOI: 10.54254/2755-2721/6/20230600

little change for different images. Therefore, it is feasible to evaluate the rank by a small number of images. Finally, Equation (5) is obtained.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{m^i} \delta_{ij} \, \mathrm{Rank}(w_j^i), \mathrm{s.t.} \sum_{j=1}^{m^i} \delta_{ij} = m_U^i \tag{5}$$

2.2. Residual block

The pruning target network of this paper is Resnet-56. Compared with other networks, the most significant feature of Resnet is the establishment of a residual path between the input and output of multi-layer (usually 2-3 layers) convolutional layers, as shown in Figure 2. The residual path added between input and output of a single convolutional layer is rarely applied due to its small lifting effect. After establishing the residual path, the layer with the residual path only needs to train the residual F(x), which is generally much smaller than X itself, and it is easy to achieve better training results. Moreover, the existence of residual module provides a good solution to the gradient disappearance problem and provides a method for building deeper convolutional neural networks.

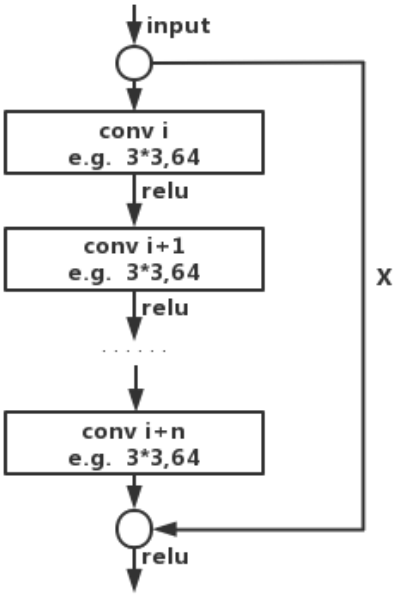

Figure 2. A brief description of residual block.

2.3. Improved method

For the pre-set compression rates that have been provided, this paper finds that there is still some room for improvement in the preset compression rate. Toward less important, making less influence on the result and deeper convolution, the default compression rate can be raised appropriately. With the small sacrifice of accuracy in exchange for a larger model pruning rate, the quantity and the floating-point computation is reduced, improving the feasibility of the deployment of the neural network on the edge equipment with weak computational ability and less available memory. Meanwhile, toward the more important convolutional layer, the preset compression rate can be appropriately lowered to reach higher accuracy and improve the reliability of the algorithm in practical application at the cost of slightly increasing the operating pressure.

Considering the different learning rates, the iterative process of the algorithm will produce great differences. When the learning rate is set too small, the iteration is prone to be too slow. Once the deviation between the random initial value and the optimal value is large, it will lead to the waste of time, and the result is not satisfactory. When the learning rate is set too large, it is easy to create invalid iterations, and finally it is difficult to achieve results with high precision. Through the horizontal

DOI: 10.54254/2755-2721/6/20230600

comparison of different learning rates, a better range of learning rates can be determined to provide a reference for future research.

3. Experiment

3.1. Experimental environment and results

On Linux, Resnet-56 [9] on CIFAR-10 [10] was first pruned with different compression rates for 30 epochs. Before pruning, the Params and Flops of Resnet-56 were 0.85M and 125.44M, respectively. After pruning under different compress rates, Params, Flops and Accuracy were recorded, as shown in Table 1.

Table 1. Experimental results under different compression rates.

Rank	Compress Rate	Params	Flops	Accuracy
1	[0.1]+[0.60]*35+[0.0]*2+[0.6]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.49M	62.72M	92.570%
2	[0.2]+[0.60]*35+[0.0]*2+[0.6]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.49M	62.66M	92.880%
3	[0.05]+[0.60]*35+[0.0]*2+[0.6]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.49M	62.74M	92.750%
4	[0.1]+[0.50]*35+[0.0]*2+[0.6]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.51M	69.02M	92.820%
5	[0.1]+[0.70]*35+[0.0]*2+[0.6]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.49M	53.75M	91.990%
6	[0.1]+[0.60]*35+[0.0]*2+[0.5]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.51M	64.04M	92.750%
7	[0.1]+[0.60]*35+[0.0]*2+[0.7]*6+[0.4]*3+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]+[0.1]+[0.4]	0.47M	61.39M	92.720%

In the column of Compress rate, "*" represents the same pre-set compression rate of successive layers. For example, [0.60]*35 means that the pre-set compression rate of layers 2 to 36 is 0.60. Then Resnet-56 were pruned for 30 epochs with different learning rates under the preset compression rate of group 1, and the results are shown in Figure 3

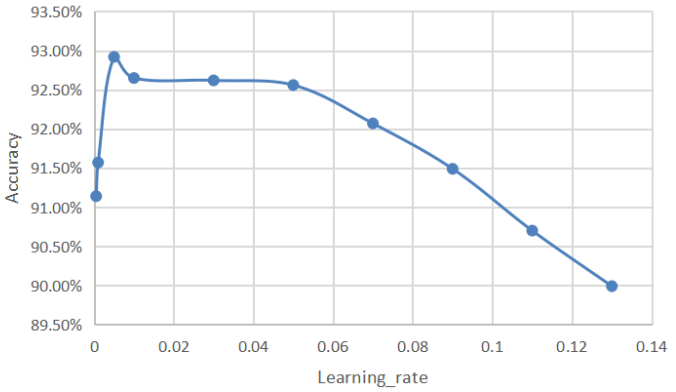

Figure 3. Accuracy results under different learning rate.

3.2. Params

As shown in Table 1, when the preset compression rate is group 7, Params is reduced to the lowest, which is 0.47M, and the compression is 44.7%. When the preset compression rate is set as group 4 and

DOI: 10.54254/2755-2721/6/20230600

group 6, the Params reach 0.51M and the compression rate drops to 40.0%. In addition, when the first convolutional layer is changed, the change of Params is very small, almost keeping at 0.49m and the compression rate is kept at 42.3%. In general, the higher the preset compression rate, the smaller the Params, which is in line with the general understanding. In group 5, although the preset compression rate changed the compression degree of more layers, it had no significant impact on Params, indicating that the amount of Params contained in the shallow convolutional layer was less than that in the deeper convolutional layer.

3.3. Flops

As shown in Table 1, when the preset compression rate is group 5, Flops decreases to the lowest, which is 53.75M, and the compression is 57.1%. When the preset compression rate is the fourth group, the Flops is the lowest (44.9%) and the Flops is 69.02M. However, for Group 6 and Group 7, which had a deeper change and less preset compression rate of convolution layer, the change in Flops was small. The reasons for this phenomenon may be as follows: deep convolution layer has a small impact on the overall Flops, and the number of convolution layers affected by changing the preset compression rate is positively correlated with the number of Flops affected.

3.4. Accuracy

As shown in Table 1, when the preset compression rate is increased, more convolution kernels classified as $\bigcup_{i=1}^{N} U_{C^i}$ are increased and more convolution kernels are pruned, thus Accuracy decreases. For example, in group 7 and Group 5, Accuracy is 92.720% and 91.990%, respectively. As the convolution kernels classified as $\bigcup_{i=1}^{N} U_{C^i}$ are reduced, fewer convolution kernels are pruned, so Accuracy increases. For example, in group 3 and Group 4, Accuracy is 92.750% and 92.820%, respectively. At the same time, it can be found that changing the deeper network has less impact on Accuracy than the shallow network, because the shallow network is a more basic network structure and plays a more important role.

3.5. Comprehensive comparison

Figure 4 reflects the relationship between Accuracy and Flops. Point 1 to 7 corresponds to Line 5, 7, 1, 3, 2, 6, 4 in Table 1. The Flops is the X-axis while Accuracy is the Y-axis. As shown in Figure. 4, with the decline in Flops and Params, Accuracy decreases, but the decrease is small. Even under some errors, Accuracy increases. Among several sets of presets in the experiment, the optimal Compress rate was group 7, which still maintained an accuracy of 92.720% while squeezing Params 44.7% to 0.47m and Flops 51.1% to 61.39m. Compared with the pruning results of group 1 in the initial compression rate, The Params compression rate increased by 2.3 percentage, the FLOPS compression rate increased by 1.1 percentage, and the accuracy rate only decreased by 0.45%.

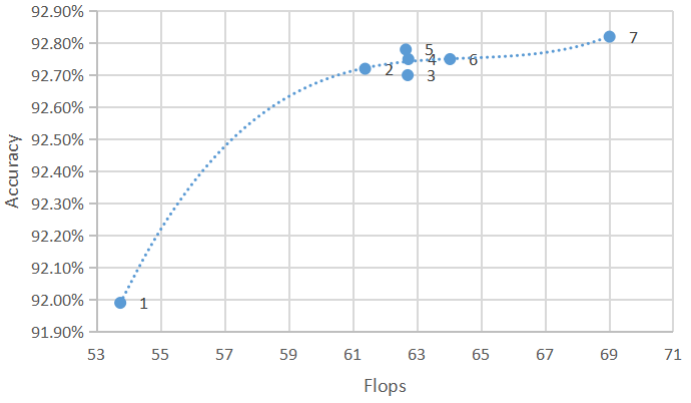

Figure 4. Accuracy results along with different flops.

DOI: 10.54254/2755-2721/6/20230600

3.6. Accuracy changes about Flops

The dashed line is created by polynomial fitting. From Figure. 4, it can be found that before the Flops are compressed to 61.00M, the decrease of Accuracy is not significant (about 92.70%). However, when the Flops are compressed to less than 61.00m, the decrease of Accuracy is rapid. Therefore, it can be inferred that in the process of pruning and optimizing the Resnet-56 model, the filter with low rank that has little influence on the results is pruned and removed preferentially. Because in the stage of Flops no less than 61.00M, the pruned filter decreases the Flops, but does not significantly affect Accuracy. When the Flops is lower than 61.00M, the filter that has little impact on the result has been almost completely pruned. Thus, in order to further improve the compression rate and reduce Flops to meet the present requirements, the HRank pruning method starts pruning the part that has a greater impact on the result, that is, the filter with higher rank, resulting in a rapid decline in Accuracy. The appearance of 61.00M cut-off point represents that HRank pruning method has a good ability to distinguish filters with different influence degrees on the results, that is, filters with different rank sizes. Otherwise, Accuracy should decline with Flops more evenly, and the decline rate should be roughly the same for different Flops. Such ability ensures that HRank pruning method can achieve higher Accuracy at a given compression rate, or otherwise achieve better pruning effect under a certain Accuracy requirement.

3.7. Learning rate

With the same preset compression rate and different learning rates adjusted, the Accuracy achieved by training in 30 epochs is shown in Figure. 3. It can be inferred that under 30 epochs, the setting range of the optimal learning rate is 0.003 to 0.05. If the learning rate is higher than 0.05, the precision of learning is too low, and the model can hardly adjust appropriate parameters. Due to the high learning rate, the Accuracy is almost not improved in the final several epochs, which wastes time and computational power.

However, when the learning rate is lower than 0.003, the amplitude of parameters adjusted for each epoch is too small, which makes the efficiency of each epoch extremely low. In order to realize the transformation from the initial value to the optimal value, a large number of epochs need to be cycled, which consumes a lot of computational power and time. Although the final result may be more refined with higher Accuracy, such an improvement in Accuracy is not worthy considering the cost of time and computing power. In the case of limited epochs, such as 30 epochs in this experiment, setting the learning rate below 0.003 will not cause invalid epochs caused by high learning rate, but the number of epochs is not enough to meet the needs, and the final result is still not ideal. When the learning rate is set between 0.003 and 0.05, the precision of learning and the efficiency of learning are both taken into account to achieve the most efficient use of epochs and to achieve the best possible results in a limited number of epochs. Particularly, Accuracy peaks when the learning rate is set to 0.005. The possible reason for this phenomenon is that the learning-rate is set appropriately to achieve the highest precision and learning efficiency, making full use of each epoch. At the same time, the initial value is relatively good, resulting in excellent results.

4. Conclusion

In this paper, mainly based on Pytorch and CIFAR-10 dataset, the pruning ability toward Resnet-56 model by HRank pruning method was tested under different preset compression rates. The paper finds that in the default compression rate for [0.1]+[0.60]*35+[0.0]*2+[0.7]*6+[0.4]*3+[0.1]+[0.4] +[0.1] +[0.4]+[0.1]+[0.4]+[0.1]+[0.4], the compression performance of the model is best. Under the default compression ratio, the model performs best when the learning rate is set to 0.005. At the same time, the causes of the phenomenon are analyzed. The deficiency of the experiment is that the sample is small, and the conclusion is relatively single. In subsequent experiments, it may be possible to continue to analyze the changes in the function of the accuracy rate on the learning rate under different preset compression rates to achieve a more universal result.

DOI: 10.54254/2755-2721/6/20230600

References

[1] Yin Q, Zhang R, Shao X L. 2019 CNN and RNN mixed model for image classification MATEC web of conferences. *EDP Sci*, **277**: 02001.

[2] Xinyu Z, Hongbo G, Jianhui Z, et al. 2018 Overview of deep learning intelligent driving methods. J. Tsinghua Univ., 58(4): 438-444.

[3] Qing L I U, Shi-chao L I, Wen-shan W, et al. 2021 Image stabilization repair method combining time series network and pyramid fusion. J. Graph., 42(1): 65.

[4] Zeng Kai, Li Xiang, Jia Jian-Mei, Wen Ji-Feng, Wang Xiang, 2022 Optimal Model for Defect Detection Based on YOLOv3-spp, Soft. Tec. 213-219.

[5] Lai C I J, Zhang Y, Liu A H, et al. 2021 Parp: Prune, adjust and re-prune for self-supervised speech recognition. Adv. Neu. Inf. Proc. Sys. 34: 21256-21272.

[6] Tan J H, Chan C S, Chuah J H. 2022 End-to-end supermask pruning: Learning to prune image captioning models. Pat. Rec. 122: 108366.

[7] Bayasi N, Hamarneh G, Garbi R. 2021 Culprit-Prune-Net: Efficient Continual Sequential Multi-domain Learning with Application to Skin Lesion Classification. Inter. Conf. Med. Image Compu. Com. Inter. 165-175.

[8] Lin M, Ji R, Wang Y, et al. 2020 HRank: Filter pruning using high-rank feature map, Conf. Com. Vis. Pat. Rec. 1529-1538.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016 Deep residual learning for image recognition. Conf. Com. Vis. Pat. Rec., 12, 5-16.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. 2009 Learning multiple layers of features from tiny images. Tec. Rep. 2-5.