symmetry

MDPI

Article

Channel Pruning Base on Joint Reconstruction Error for Neural Network

Bin Li [1,2,*], Shimin Xiong [1] and Huixin Xu [1]

[1] School of Computer Science, Northeast Electric Power University, Jilin 132012, China; 2202000683@neepu.edu.cn (S.X.); xhx18643638196@163.com (H.X.)
[2] Gongqing Institute of Science and Technology, No. 1 Gongqing Road, Gongqing 332020, China
* Correspondence: libinjlu5765114@163.com

Abstract: In this paper, we propose a neural network channel pruning method based on the joint reconstruction error (JRE). To preserve the global discrimination ability of a pruned neural network, we propose the global reconstruction error. To ensure the integrity of information in the forward propagation process of a neural network, we propose the local reconstruction error. Finally, through normalization, the two magnitude mismatched losses are combined to obtain the joint error. The baseline network and pruned network are symmetrical structures. The importance of each channel in the pruned network is determined by the joint error between the channel and the corresponding channel in the baseline network. The proposed method prunes the channels in the pruned network according to the importance score and then restores its accuracy. The proposed method reduces the scale of the neural network and speeds up the model inferring speed without losing the accuracy of the neural network. Experimental results show the effectiveness of the method. For example, on the CIFAR-10 dataset, the proposed method prunes 50% of the channels of the VGG16 model, and the accuracy of the pruned model is 0.46% higher than that of the original model.

Keywords: channel pruning; global reconstruction error; local reconstruction error; joint reconstruction error

Citation: Li, B.; Xiong, S.; Xu, H. Channel Pruning Base on Joint Reconstruction Error for Neural Network. *Symmetry* **2022**, *14*, 1372. http://doi.org/10.3390/sym14071372

Academic Editor: Mihai Postolache

Received: 15 June 2022
Accepted: 30 June 2022
Published: 4 July 2022

1. Introduction

Although neural networks have made great progress in image classification, face recognition, target detection, image generation and video analysis, there is a problem of over parameterization in the current large-scale neural network model. That is, some parameters make little contribution to the final output result, which wastes computing resources and storage space. Many neural network models need to be compressed before they can be applied to the hardware platform with limited computing power. For example, in [1], the main challenge for the author to run the artificial neural network on the field programmable gate arrays (FPGAs) is the lack of hardware resources. Model compression aims to reduce the redundancy of neural network models without significant performance degradation.

Low-rank approximation/decomposition or knowledge distillation are traditional neural network compression techniques. Low rank approximation/decomposition reduces the data dimension by singular value decomposition (SVD) of the weight matrix [2–4]. Although this method is relatively easy to implement, not all convolution kernels are low rank in neural networks, and the compression capacity of low-rank approximation is limited. Knowledge distillation adopts transfer learning to teach the knowledge of large models to small models to the greatest extent to achieve the purpose of compressing models [5–7]. Although the knowledge distillation method is suitable for more network types than the low rank approximation/decomposition method, the design of student models, the selection of dataset for training the student model, the way to select feature layers and design loss function will bring uncertainty to the network model compressed

by knowledge distillation, resulting in unsatisfactory compression ratio and performance after distillation.

To overcome the shortcomings of the above two strategies, researchers have proposed a network compression technology based on channel pruning. At present, channel pruning based on sparse training and channel pruning based on feature map reconstruction error are mature channel pruning techniques. The former obtains the sparse network through sparse training and sets a threshold value to prune the channels below the threshold [8–11]. However, measuring the importance of a channel with a set threshold is obviously too one-sided, and training from scratch is very difficult for complex networks. Compared with channel pruning based on sparse training, channel pruning based on the feature map reconstruction error is obviously more persuasive by minimizing the local output feature map reconstruction error of the pruned network and the baseline network [12–15]. However, this method considers the integrity of information in the forward process from a local perspective and does not consider the functionality of the model from a global perspective. This pruning method may degrade the discrimination ability of the pruned model and make it difficult to fine-tune the accuracy of the pruned model.

Compared with the above two strategies, we consider the importance of the channel from both global and local perspectives. First, we calculate the channel importance score from a global perspective to preserve the overall functionality of the model, that is, the discrimination ability of the model. Because the baseline network and pruned network are symmetrical structures, the importance of channels is evaluated from the local point of view by calculating the feature map reconstruction error of the pruned network and the baseline network to ensure the integrity of information in the forward propagation process of the model. Finally, the channels are pruned using global and local importance scores. In this way, we can balance the overall functionality of the model with the integrity of information in the forward propagation process. The main contributions of this article are summarized below.

(1) We propose a neural network channel pruning method (JRE) based on the joint error. This method considers the importance of the channel from a global and local perspective. The proposed JRE can evaluate the importance of channels from two perspectives: ensuring the overall functionality of the model and the integrity of information in the forward process.

(2) We transform the sensitivity of each layer of the neural network to prune into a channel importance ranking problem based on the joint reconstruction error. That is, the reserved channels are determined according to their local and global importance and distributed in each layer.

(3) We proved the superior performance of JRE by pruning experiments on VGG16 [16] and ResNet18 [17]. When pruning 50% channels of VGG16, JRE improves the accuracy of the original model by 0.46%, which is 0.29% higher than that of DCP [15]. When pruning ResNet18, JRE improves the accuracy of the original model by 0.14%, which is 0.02% higher than that of the DCP [15] method.

This paper is arranged as follows: The second part briefly reviews the related work, the third part details the proposed model-based pruning neural network channel pruning method (JRE), the fourth part demonstrates the effectiveness of the proposed method through experiments, and the last part summarizes the full text.

2. Related Studies

Low-rank approximation/decomposition: There are usually many correlations between the channels of a neural network, and the low-rank approximation method is to eliminate the redundancy caused by this correlation. The low-rank approximation/decomposition method is easier to implement. For example, Astrid et al. [18] compressed the convolution layer through tensor canonical polyadic (CP) decomposition, which accelerated the task of text recognition by 4.5 times. However, the low-rank approximation method cannot remove redundant channels that are independent of the network's discrimination capability,

and now, some neural networks use 1×1 convolution, which makes it difficult to achieve network acceleration and compression using matrix decomposition.

Distillation of knowledge: Distillation of knowledge uses transfer learning to maximize the knowledge of large models to small models in order to compress them. For example, Hinton et al. suggest that the student model can be modeled to achieve the same precision as the teacher model by mimicking the teacher model, reducing the model by one-third of its parameters [19]. Although the knowledge distillation method is suitable for more network types than the low-rank approximation/decomposition method, this method requires the completion of the student model design, the selection of appropriate training datasets for the student model, the selection of feature layers, and the design of loss functions. The quality of these tasks will create a lot of uncertainties about the performance of the compression model. At present, the compression ratio and the performance of the model compression based on knowledge distillation are not satisfactory.

Channel pruning: Compared with low-rank decomposition/approximation and knowledge distillation, channel pruning is suitable for most deep learning networks today and does not require the design of additional small models. The key of channel pruning is to determine the importance of channels, which can be divided into two ways. The first idea is channel pruning based on sparse training. Li et al. measured the importance of the channel by sparsely training the model and then calculating the F2 norm of the channel [8]. However, it is obviously too one-sided to measure the importance of channels only by F2 norm. Especially in tasks requiring high accuracy, this method may mistakenly prune channels that should not be deleted, resulting in difficulty in restoring the accuracy of the original model. Liu et al. added L1 regularization to the scale factor of BN layer to achieve sparse effect during training and then identified the unimportant channel [9] by the scale factor of the BN layer tending to 0. However, when the scale factor of all BN layers tends to 0, the method will misjudge the channel importance. Gao et al. [10] trained an independent neural network to predict the performance of the sub-network and then guided pruning by maximizing the performance of the sub-network. Li et al. [20] proposed that some filters can be made the same through training, and multiple identical filters can be combined to achieve the effect of reducing the model. However, it is still too one-sided to measure the importance of filters only by sparsity training, especially in the application fields that need high accuracy. It will be difficult to restore the accuracy before pruning if the wrong filter is pruned off. Pruning based on local reconstruction error is more reasonable than the method based on sparse training.

The second idea is channel pruning based on the reconstruction error of the feature map of the pruned network and the baseline network. For example, the two methods in [12,13] determine which channels need to be pruned by minimizing the feature reconstruction error of the pruned network and the baseline network. Li et al. believed that it was not enough to consider only the reconstruction errors of the latter layer or two layers, and they proposed a method to consider the reconstruction errors of the second-last layer feature map of the network [14]. While considering the reconstruction error, Zhang et al. [15]. introduced discrimination aware loss in the middle layer of the network, which increased the discrimination ability of the middle layer of the network. However, this method only considers the discrimination ability of the channel from a local perspective, and it does not consider the change of the loss function from a global perspective.

3. Proposed Method

3.1. Motivations

The DCP method based on the reconstruction error of the local feature map [15] performs channel pruning through Formula (1).

$$\mathcal{L}(W) = \mathcal{L}_M(W) + \lambda \mathcal{L}_S^P(W) \tag{1}$$

In Formula (1), $\mathcal{L}_M(W)$ represents the reconstruction error of the local feature map, and $\mathcal{L}_S^P(W)$ denotes the discrimination aware loss. $\mathcal{L}_S^P(W)$ can retain the discrimination

ability of some layers in the middle of the model. Since $\mathcal{L}_S^P(W)$ is calculated in the middle layer of the model, the importance ($\mathcal{L}_S^P(W)$) of a channel is calculated based on the information transmitted by this layer and its previous layers. We believe that this channel importance calculation method is a local greedy algorithm, which does not consider the importance of the channel from the global point of view. Especially in networks with deep layers, the importance of each layer cannot be determined only by the information of the previous layers but also by the global information of the whole network.

In this paper, we try to measure the importance of channels from both local and global aspects. The proposed JRE method can not only ensure the integrity of information in the forward propagation process but also ensure the discrimination ability of the model. That is, the functionality of the model itself does not decline. To preserve the discrimination ability of the neural network from a global perspective, we add the global reconstruction error to the neural network. Because other operations (such as normalization, ReLU activation function, maximum pooling, etc.) in convolution neural networks do not play a role in feature extraction, we assume that the feature information is transmitted directly between the convolution layers when calculating the local reconstruction error. The proposed channel pruning method based on joint reconstruction error is shown in Figure 1.

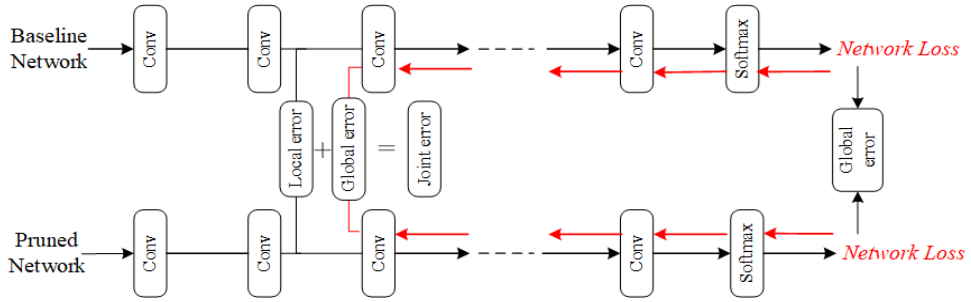


Figure 1. Channel pruning based on joint reconstruction errors.

As shown in Figure 1, the baseline network and the pruned network are symmetrical to each other. The local error represents our proposed local reconstruction error, and the global error represents our proposed global reconstruction error. We will select channels based on joint errors calculated by local reconstruction errors and global reconstruction errors. The Network Loss (*NL*) in Figure 1 is the loss of the whole network, which will be used to calculate the global reconstruction errors in the back propagation process.

3.2. Local Reconstruction Error

Local reconstruction errors ensure that the integrity of information in the forward propagation process of the network. We try to find such a set of convolution kernels from the convolution layer to satisfy Formula (2),

$$minL_p = min \parallel O^b_{i,:,:,:} - O_{i,j,:,:} \parallel_F^2 \tag{2}$$

where $O^b_{i,:,:,:}$represents the output feature map of the ith layer of the benchmark model b, $O_{i,j,:,:}$ represents the output feature map of the ith layer of the model after pruning, and j represents the remaining channels. The formula means that we are trying to select a set of kernels in the ith layer of the baseline model so that the F-norm between $O_{i,j,:,:}$ and $O^b_{i,:,:,:}$ is the minimum, which ensures that nonessential channels are pruned. Because selecting a series of combinations that satisfy formulas from hundreds of kernels in a layer requires a lot of computing resources, we reduce the amount of calculation by equating Formula (2) with Formula (3),

$$minL_p = min \parallel Q_{i,p,:,:} \parallel_F^2 \tag{3}$$

where $Q_{i,p,:,:}$ represents the output feature map calculated from the pruned convolution kernels in the ith layer of the model and P represents the eliminated channel. The problem

of finding preserved kernels in Formula (2) is transformed into a problem of finding pruned kernels, which reduces the scale of the problem. The local reconstruction error can be calculated by Formula (4).

$$L_p = \| Q_{i,p,:,:} \|_F^2 \tag{4}$$

The local reconstruction error L_p represents the channel importance score based on a local calculation. Since other operations in the network (such as normalization, Sigmoid activation function, maximum pooling, etc.) do not perform feature extraction, when calculating the local channel importance score, we assume that the information is the direct transmission between convolution layers. That is, the importance of layers other than convolution is not considered.

3.3. Global Reconstruction Error

The global reconstruction error guarantees the discrimination ability of the model. That is, the functionality of the model itself does not decline. We sought to find such a set of convolution kernels to satisfy Formula (5),

$$minL_{os} = min|NL(D|h = 0) - NL(D|h = h_i)| \tag{5}$$

where L_{os} represents the difference of the Network Loss (NL) before and after channel h is pruned. $NL(D|h = 0)$ represents the Network Loss (NL) value when channel h is 0. That is, the loss function value of channel h is pruned. In addition, $NL(D|h = h_i)$ represents the NL value when channel h is h_i. Curve fitting is introduced into Formula (5), as shown in Formula (6).

$$min(L_{os})^2 = min(|NL(D|h = 0) - NL(D|h = h_i)|)^2 \tag{6}$$

Complex neural networks usually have a large number of random parameters, so it is difficult to make neural networks before and after pruning reach the same state. Therefore, the first-order Taylor expansion formula is introduced, the function $NL(D|h)$ is expanded at h_i, and all except the first-order terms are discarded to obtain the first-order Taylor expansion of $NL(D|h)$ at h=0 , as shown in Formula (7).

$$NL(D|h = 0) = NL(D|h = h_i) - \frac{\delta NL}{\delta h_i}(h = h_i) \tag{7}$$

Bring Formula (7) into Formula (6) to obtain Formula (8),

$$min(L_{os})^2 = (\frac{\delta NL}{\delta h_i}(h = h_i))^2 \tag{8}$$

where $\frac{\delta C}{\delta h_i}$ represents the gradient of the loss function for h_i during back propagation. Our demand is transformed into finding such a channel h, and the product of its value and its corresponding back-propagation gradient is close to 0. If the product of the value of the channel h and its corresponding back-propagation gradient is close to 0, the channel h has little effect on the discrimination ability of the neural network. The global reconstruction error can be calculated by Formula (9).

$$L_{os} = \frac{\delta NL}{\delta h_i}(h = h_i) \tag{9}$$

where L_{os} represents the channel importance score based on global calculation. When the score is smaller, it indicates that the kernel is less important and vice versa. The global reconstruction error L_{os} reflects the importance of a channel to the whole model.

3.4. Joint Loss Function

Due to the magnitude of the global reconstruction error and the local reconstruction error being different, we use the normalization method to calculate the joint loss function. The global reconstruction error is normalized by Formula (10).

$$L_{os}^{R} = \frac{L_{os} - min(L_{os})}{max(L_{os}) - min(L_{os})} \tag{10}$$

The local reconstruction error is normalized by Formula (11), and the joint error can be obtained by Equation (12).

$$L_{p}^{R} = \frac{L_{p} - min(L_{p})}{max(L_{p}) - min(L_{p})} \tag{11}$$

$$L = L_{os}^{R} + L_{p}^{R} \tag{12}$$

3.5. Channel Pruning Based on Joint Reconstruction Error

The proposed JRE method can be described by Algorithm 1. Start with the pre-trained model M and take stage i as an example. Feature maps propagate forward layer by layer in the network. Algorithm 1 first determines whether the layer is a convolution layer. If it is a convolution layer, L_p is calculated by Formula (4). Then, L_{os} is calculated by Formula (9) during back propagation, and finally, the joint error L is calculated by Formula (12). Prune the pre-trained model M according to the ascending sorting result of L.

In addition, because fine tuning is very important to compensate the accuracy loss of model M to suppress the cumulative error, we add fine tuning at the end of Algorithm 1.

At each stage, we calculate the importance (joint error L) of each channel in model M and then sort the importance, so the pruning rate ratios of different layers can be potentially included in our method. We do not need to analyze the sensitivity of each layer to pruning nor do we need to manually define the pruning rate ratio of different layers. We only need to predefine a desired global pruning rate before the model pruning operation. When the global pruning rate is reached, the pruning operation will stop automatically.

Algorithm 1: Channel pruning

```
Input: Pre-trained model M,training data
while the pruning rate is not reached do
    for module in M do
        #Forward propagation process
        if module is conv then
            Calculate L_p using Formula (4);
            Save h_i;
        end
    end
    Calculate L_os using Formula (9) when backward;
    #Back propagation process
    Calculate L using Formula (12) and sort L in ascending order;
    Select a minimum batch of filters according to L and Prune filters from M;
    Fine-tune M;
end
```

4. Experiments

In this section, we will evaluate the performance of the channel pruning method based on joint error. Several advanced methods are adopted as the baseline, including L1 norm [8], thinet [12], and DCP [15]. We evaluate the performance of our method on the CIFAR-10 dataset [21]. CIFAR-10 contains 50 K training images, 10 K test images and 10 categories.

4.1. Implementation Details

We implement the proposed JRE on pytorch. Based on the pre-trained model, the JRE is applied to select the channel. On the CIFAR-10 dataset, we use SGD for optimization. In each round of iteration, we set the batch size to 128, the number of channels to be pruned is 128, the number of trainings used to evaluate the importance of channels is 40 epochs, the number of fine-tuning after pruning is 20 epochs, and the learning rate is 0.1.

4.2. Comparisons on CIFAR-10

We pruned ResNet-18 [17] and VGG16 [16] on the CIFAR-10 dataset. Table 1 reports the performance of the models obtained by pruning VGG16 by other methods and the models obtained by pruning VGG16 by our JRE. Table 2 shows the performance of the models obtained by pruning ResNet-18 by other methods and the models obtained by pruning ResNet-18 by our JRE. In Tables 1 and 2, the pruning rate of 50% represents 50% of the channels in the model to be pruned. Accuracy (%) indicates the change of model accuracy before and after pruning. For example, +0.14 indicates that the accuracy is 0.14% higher than that of the model before pruning. The FLOPs↓ represents the multiple of the reduction of the number of floating-point operations of the model. Value 2.1× indicates that the number of floating-point operations of the model is reduced by 2.1 times.

The results in Table 1 show that when the pruning rate is 50%, the number of floating-point operations of the model is reduced by 2.1 times, and the JRE improves the accuracy of VGG16 by 0.46%. Table 2 shows that when the pruning rate is 50%, the number of floating-point operations of the model is reduced by 2.2 times, and the JRE improves the accuracy of ResNet-18 by 0.14%. The model pruned by the JRE method achieves higher accuracy than the model pruned by other methods at the same pruning rate.

Table 1. Comparisons on CIFAR-10 (VGG16).

Model	**L1-Norm [8]**	**ThiNet [12]**	**DCP [15]**	**JRE**
Pruning rate (%)	50%	50%	50%	50%
Accuracy (%)	+0.15	−0.14	+0.17	+0.46
FLOPs↓	2.1×	2.1×	2.1×	2.1×

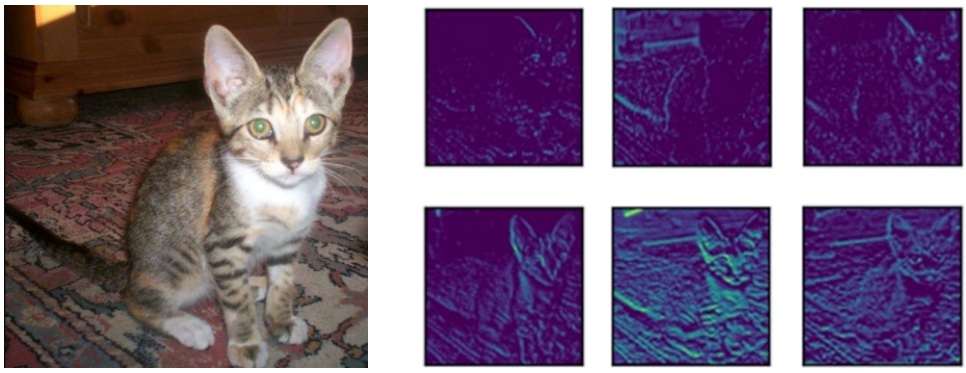
Table 2. Comparisons on CIFAR-10 (ResNet-18).

Model	**L1-Norm [8]**	**ThiNet [12]**	**DCP [15]**	**JRE**
Pruning rate (%)	50%	50%	50%	50%
Accuracy (%)	+0.09	−0.2	+0.12	+0.14
FLOPs↓	2.2×	2.2×	2.2×	2.2×

4.3. Visualization of Feature Maps

We visualized the output channel of the third convolution layer in VGG16. The first and second lines of Figure 2b show three pruned channel images and three retained channel images, respectively.

As shown in Figure 2b, the retained channels contain more detailed contour features and high-level semantic features than the pruned channels. This shows that our proposed JRE can leave more information-rich channels to preserve the discrimination ability of the network and the integrity of information in the process of forward after pruning.



(**a**) (**b**)

Figure 2. Visual comparison of pruned/reserved channels of the third convolution layer in VGG16. (**a**): Input diagram; (**b**): output channel comparison diagram.

4.4. Ablation Study

Tables 3 and 4 show the results of ablation study using VGG-16 and ResNet-18, respectively. The two compared pruning methods are the pruning method based on the global reconstruction error (GRE) proposed in this paper and the JRE including the local reconstruction error and global reconstruction error.

It can be seen from Table 3 that the accuracy of the model obtained by 50% pruning VGG-16 with GRE is 0.25% higher than that of the original model. This precision is higher than that of L1-norm, ThiNet, and DCP in Table 1. At the same time, it can be seen from Table 4 that the accuracy of the model obtained by 50% pruning ResNet-18 with GRE is 0.11% higher than that of the original model. This precision is higher than that of L1-norm and ThiNet in Table 2 but slightly lower than that of DCP (0.12%). This proves the effectiveness of the proposed global reconstruction error. The proposed JRE achieves higher accuracy by combining the global reconstruction error and local reconstruction error, which proves the effectiveness of the two reconstruction errors proposed in this paper.

Table 3. Ablation study using VGG-16.

Model	GRE	JRE
Pruning rate (%)	50%	50%
Accuracy (%)	+0.25	+0.46
FLOPs↓	2.1×	2.1×

Table 4. Ablation study using ResNet-18.

Model	GRE	JRE
Pruning rate (%)	50%	50%
Accuracy (%)	+0.11	+0.14
FLOPs↓	2.1×	2.2×

5. Conclusions

In this paper, a channel pruning neural network compression method based on joint error is proposed. The experimental results on the CIFAR-10 dataset show that this method is superior to several advanced methods under the same pruning rate. The results of ablation studies prove the effectiveness of the global reconstruction error, local reconstruction error and the final method (JRE). Although the proposed JRE is proved to be effective by experiments, if the pruning rate exceeds 50%, the accuracy of the pruned model may be slightly lower than the original model. In the future, we plan to optimize the JRE based on the idea of dynamic pruning. Instead of pruning directly, we will mask the filters to be pruned to stop the current round of parameter updates.

Author Contributions: Conceptualization, methodology, resources, funding acquisition, B.L.; software, validation, visualization, S.X.; writing—original draft preparation, writing—review and editing, S.X. and H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Development Plan Project of Jilin Province under Grant 20200201165JC. Funder: Jilin Provincial Department of Science and Technology; Funding Number: 20200201165JC.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This article is not about human research.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pano-Azucena, A.D.; Tlelo-Cuautle, E.; Ovilla-Martinez, B.; de la Fraga, L.G.; Li, R. Pipeline FPGA-based Implementations of ANNs for the Prediction of up to 600-steps-ahead of Chaotic Time Series. *J. Circuits Syst. Comput.* **2021**, *30*, 2150164. [CrossRef]
2. Jaderberg, M.; Vedaldi, A.; Zisserman, A. Speeding up convolutional neural networks with low rank expansions. *arXiv* **2014**, arXiv:1405.3866.
3. Denton, E.L.; Zaremba, W.; Bruna, J.; LeCun, Y.; Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. *Adv. Neural Inf. Process. Syst.* **2014**, 27.
4. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1529–1538.
5. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
6. Mishra, A.; Marr, D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv* **2017**, arXiv:1711.05852.
7. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4133–4141.
8. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning structured sparsity in deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, 29.
9. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
10. Gao, S.; Huang, F.; Cai, W.; Huang, H. Network pruning via performance maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9270–9280.
11. Zhuang, T.; Zhang, Z.; Huang, Y.; Zeng, X.; Shuang, K.; Li, X. Neuron-level structured pruning using polarization regularizer. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9865–9877.
12. Luo, J.-H.; Wu, J.; Lin, W. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5058–5066.
13. He, Y.; Zhang, X.; Sun, J. Channel pruning for accelerating very deep neural networks. In Proceedings of IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1389–1397.
14. Yu, R.; Li, A.; Chen, C.-F.; Lai, J.-H.; Morariu, V.I.; Han, X.; Gao, M.; Lin, C.-Y.; Davis, L.S. Nisp: Pruning networks using neuron importance score propagation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9194–9203.
15. Zhuang, Z.; Tan, M.; Zhuang, B.; Liu, J.; Guo, Y.; Wu, Q.; Huang, J.; Zhu, J. Discrimination-aware channel pruning for deep neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, 31.
16. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Astrid, M.; Lee, S.-I. Cp-decomposition with tensor power method for convolutional neural networks compression. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Korea, 13–16 February 2017; pp. 115–118.
19. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
20. Ding, X.; Ding, G.; Guo, Y.; Han, J. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4943–4953.
21. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report, University of Toronto: Toronto, ON, Canada, 2009.