A survey of model pruning for deep neural network

Zhuo Li[1,†], Lin Meng[2,*,†]

[1]Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan
[2]College of Science and Engineering, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

Abstract

Model pruning is an important research field for compressing deep neural networks (DNNs) and has attracted extensive studies during the past decades. Although deep learning has achieved great success in various fields, the native characteristics of overwhelming demand for hardware resources with high computation intensity and memory intensity have been heavy burdens that block the effective application of the technique. For the problem, model pruning provides a promising solution to compress DNNs and thus reduce the demand for computation cost. In addition, the pruning method has made amazing achievements. This paper makes a review on the pruning techniques of DNNs to provide overall reference for concerning research. Firstly, the research background is introduced. After that, we provide a brief overview of the pruning process. Then, the current model pruning methods, structured pruning, and unstructured pruning are introduced. Finally, we make a summary and look to the future.

Keywords

Model pruning, deep neural networks, deep learning, structured pruning, unstructured pruning

1. Introduction

In recent years, deep neural networks (DNNs) have made great achievements in artificial intelligence-based tasks, including the application tasks of image classification [1, 2, 3, 4], control theory [5, 6] and objective detection [7, 8]. In 1998, LeCun et al. [9] proposed the LeNet-5 network for simple image classification. In the ImageNet image classification competition in 2012, AlexNet [10] increased the network depth to 8 layers and achieved excellent classification results. VGGNet [11] improves the network performance by extending the depth of the network. He et al. proposed ResNet [12] in 2016, which effectively relieves the gradient disappearance problem of DNNs by adding a residual structure to the model. And so on. As DNNs perform better and better with higher accuracy, they also become more and more complex with increasing demands for computation and memory resources, which takes more hardware resources. Those heavy burdens have seriously limited the effective applications of DNNs. At the same time, it has also been proved that the current DNNs are over-parameterized to a great extent [13, 14].

The 4th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2022), August 24-26, 2022, Ibaraki, Japan
*Corresponding author.
†These authors contributed equally.
lizhuo970604@gmail.com (Z. Li); menglin@fc.ritsumei.ac.jp (L. Meng)
0000-0003-4554-6018 (Z. Li); 0000-0003-4351-6923 (L. Meng)

Therefore, more and more research has been done to compress DNNs by removing the redundant parts of the networks.

As shown in Fig 1, the model compression techniques are simply classified into four classes: model quantization, model pruning, low-rank approximation, and knowledge distillation [15]. Among them, model pruning [16] is quite effective by directly removing the redundant parts

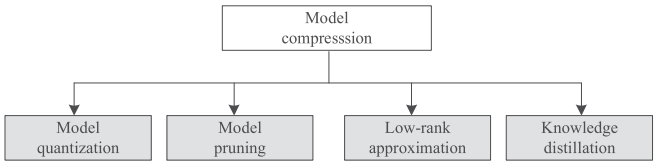

Figure 1: Model compression.

from the bloated networks.

In 1993, Redd [17] provided a survey of pruning algorithms. And a method is presented that trains a large network and then removes the parts that are not needed. In 2018, Liu et al. [18] rethink the value of network pruning. The results showed that a more detailed evaluation is needed in future studies of structured pruning methods. In this paper, we focus on making an investigation of pruning techniques for DNNs, including both structured [19] and unstructured[20] pruning methods.

The rest of this paper is organized as follows: Section 2 details the process for pruning operations on DNNs. Section 3 introduces structured and unstructured pruning. In the end, Section 4 concludes this paper and looks forward to the future.

2. Pruning process

Pruning is a traditional method for reducing model parameters and computational effort [21, 22, 23]. With the rise of deep learning and a large number of applications of DNNs in the field of image classification, various pruning methods have been proposed. In general, the overall process of pruning algorithms is divided into three stages: standard pruning, pruning based on sub-model sampling, and search-based pruning, as is shown in Fig 2.

Standard pruning consists of three main parts: training, pruning, and fine-tuning, as shown in Fig 2 a). In addition, the pruning and fine-tuning are iterated multiple times to achieve a higher pruning ratio. The process is detailed as follows.

1. Training. The purpose of training is to configure the parameters of the network to obtain the trained model by learning from a large amount of data concerning the specific task. In the pruning process, the training only needs to be done once.
2. Pruning. The DNN structure mainly contains filters, blocks, and other structures. The significance assessment of the network structure is divided into two approaches: network parameter-driven assessment and data-driven assessment. The key to pruning is to distinguish the important assessment and the superfluous parts of the network structure. The algorithms to identify the redundant structures in the networks are crucial for various pruning proposals, which determine the efficiency of the pruning results.

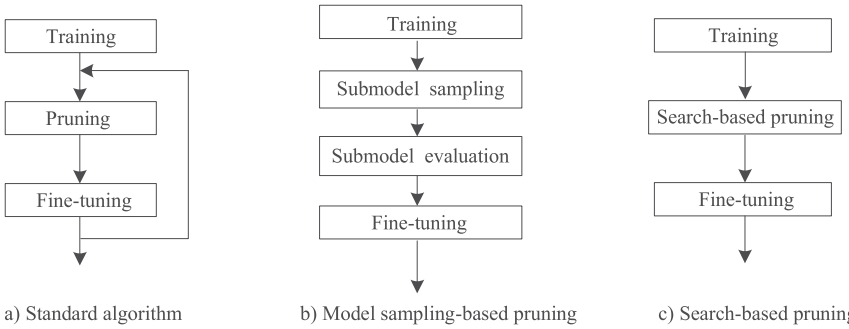

Figure 2: Pruning process. a) Standard pruning algorithm. b) Pruning based on sub-model sampling. c) Search-based pruning.

- The parameter-driven uses information about the parameters of the model structure using the model itself to measure the importance of the model structure, such as l_1 regularization or l_2 regularization of the parameters, and the evaluation process of this type of approach is not dependent on the input data.
- The data-driven approach evaluates the importance of the network structure by using training data, such as the number of 0-values after the filter output is counted through the activation layer, to assess the importance of the filter.

3. Fine-tuning. Fine-tuning is the last step to restoring the expressiveness of the model affected by the pruning operation. Structured model pruning will adjust the original model structure, so the expressiveness of the pruned model will be affected to some extent.
4. Re-pruning. The re-pruning process sends the fine-tuned sub-model to the pruning model, where the model structure is evaluated and the pruning process is performed again. Through the pruning process, each pruning is carried out on top of the model with better performance, and the pruning model is continuously optimized in stages until the model meets the pruning objectives.

In terms of standard pruning, it is the most adopted procedure for the current pruning methods [24]. [24] integrates the pruning process into the model fine-tuning; no more distinguishing between the fine-tuning and pruning parts; proposes a new trainable network layer for the pruning process. This network layer generates the binary code, and the network structure corresponding to the 0 value in the binary code is pruned.

In addition to standard pruning, the methods based on sub-model sampling have recently shown good pruning results. The pruning process based on sub-model sampling is shown in Fig 2 b). Based on the trained model, the sub-model sampling process is performed. The process is as follows:

1. The parable network architecture in the trained original model is sampled according to the pruning target. The sampling process is either random or probabilistic according to the importance of the network architecture.

2. The sampled network architecture is pruned to obtain the picking model. The sub-model sampling process is usually performed n times, obtain n sub-models ($n \geq 1$). Afterward, the performance of each sub-model is evaluated.

Search-based pruning mainly relies on reinforcement learning or neural network architecture search-related theory, and its main process is shown in Fig 2 c). Given the pruning target, search-based pruning searches for the best substructure in the network structure. This search process is often accompanied by a learning process of the network parameters, so some search-based pruning algorithms do not need to be fine-tuned after the pruning is completed.

In the process of model pruning, it is divided into structured and unstructured pruning, where structured pruning consists of a filter and layer pruning. The pruning method is introduced in detail in the next section.

3. Pruning methods

3.1. Structured pruning

Structured pruning is usually performed with the filter(channel) [25] as the basic pruning unit. When a filter is pruned, the previous feature maps and the following feature map corresponding to this filter are removed accordingly. But the architecture of the model is unbroken. Therefore, this type of method is called "structured pruning," as shown in Fig 3.

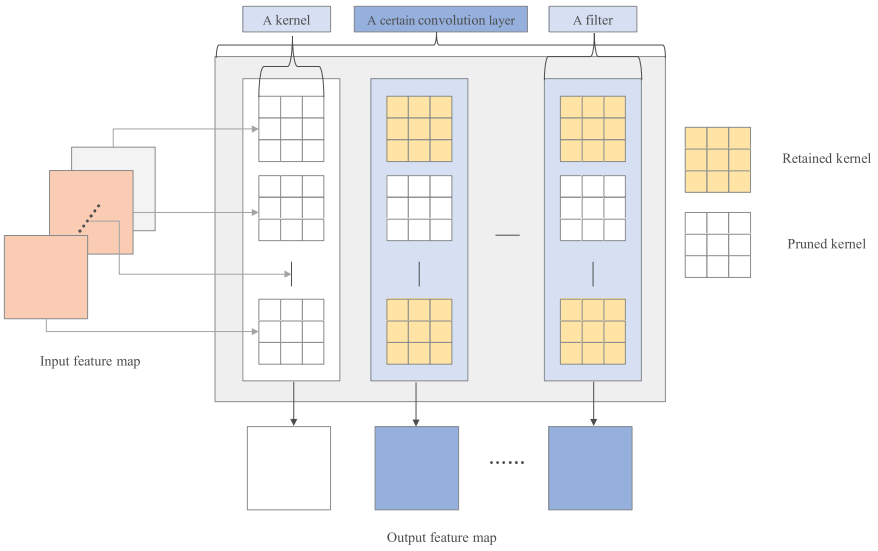

Figure 3: Structured pruning.

3.1.1. Filter-based structured pruning

Filter-based structured pruning is realized by evaluating the importance of filters in the network. In 2016, Li et al. [26] presented a method by removing whole filters in the network together

with their connecting feature maps. By measuring the relative importance of a filter in each layer by calculating the sum of its absolute weights. This method does not result in sparse connectivity patterns. Meanwhile, it does not need the support of sparse convolution libraries. Compared to layer-by-layer iterative fine-tuning, this approach uses a one-shot pruning and retraining strategy, saving the multilayer filter retraining time. This advantage is crucial when pruning deep networks. However, this has a small loss on the output accuracy. To reduce the performance loss during model pruning. In 2018, Lin et al. [27] proposed a global & dynamic pruning (GDP) scheme to prune redundant filters. GDP first globally prunes the insignificant filters of all layers through a proposed global discriminant function based on the prior knowledge of each filter. After that, it dynamically updates the filter saliency all over the pruned sparse network and then recovers the mistakenly pruned filter, followed by a retraining phase to improve the model accuracy. For the non-convex optimization problem corresponding to GDP, stochastic gradient descent with a greedy selection update is adopted.

To better evaluate the importance of filters. In 2022, Kuang et al. [28] obtained the importance of filters by considering the effect of each filter on the task-dependent loss function. The smaller the effect on the task-related loss function, the lower the importance of the filter. Use this to remove unimportant filters. For automated pruning, Chang et al. [29] proposed an automatic channel pruning method. This method first performs hierarchical channel clustering through the similarity of feature maps and performs preliminary pruning of the network. Then, a population initialization method is introduced to transform the pruned structure into a candidate population. Finally, iterative search and optimization are performed based on particle swarm optimization to find the optimal compression structure. In order to reduce the accuracy loss caused by pruning, the compressed network needs to be retrained.

Filter-based pruning has also been applied in the fields of image segmentation and object detection. Sawant et al. [30] proposed optimal-score-based filter pruning (OSFP) approach to prune redundant filters according to their relative similarity in feature space. OSFP eliminates redundant filters and improves segmentation performance while speeding up network learning. Unlike multiple pruning, the OSFP approach globally prunes the redundant filters at once. As a special pruning method, sparse training [31] and Mask learning [32] are able to establish new connections during pruning. Chu et al. [33] proposed a three-stage model compression method: dynamic sparse training, group channel pruning, and spatial attention distilling in the field of object detection. Group channel pruning divides the network into multiple groups according to the scale of the feature layers and the similarity of the module structures in the network. Then, the channels in each group are pruned with different pruning thresholds.

3.1.2. Layer-based structured pruning

Layer-based structured pruning is realized by evaluating the importance of layers in the network. To obtain the importance of network parameters in a better way. In 2017, Liu et al. [34] presented network slimming, which requires no special software/hardware accelerators for the model. During training, insignificant channels are automatically identified and pruned afterward. It employs L1 regularization on the weights of BN layers to achieve the sparsity of the parameters. After that, multiple fine-tunings are performed to achieve a high pruning rate. Yang et al. [35] proposed an energy-aware pruning algorithm. The algorithm guides the process of pruning by

using the energy consumption of the convolutional neural network (CNN). The implementation of pruning is layer-by-layer, that is, more aggressively than previously proposed pruning methods by minimizing the error in the output feature maps instead of the filter weights. For each layer, the weights are first pruned and then locally fine-tuned with closed-form least squares to restore model accuracy. After all, layers are pruned, and the entire network is globally fine-tuned using backpropagation. In 2021, Fan et al. [36] presented layered channel pruning, which groups the different layers by decreasing the model accuracy of the pruned network. The network is retrained after pruning each layer in a specific order. While there is a small decrease in the accuracy of the network model, the computational resources for neural networks to be deployed on the hardware are greatly reduced.

To reduce the computational cost of multiple training. In 2021, Chen et al. [37] proposed Only-Train-Once (OTO), a training and pruning framework. OTO greatly simplifies the complex multi-stage training pipeline of current pruning methods. Meanwhile, they proposed Half-Space Stochastic Projected Gradient, a method that solves the problem of structured-sparsity inducing regularization. Compared with multiple fine-tuning, OTO only needs one time, which greatly simplifies the pruning process. Chung et al. [38] pruned out some of the convolutional filters in the first layer of the pre-trained CNN. This first-layer pruning greatly facilitates the filter compression of the subsequent convolutional layers. However, the input to this method is a single channel. To address this issue. In 2022, Chen et al. [39] proposed a solution to strategically manipulate neurons by "grafting" appropriate levels of linearized insignificant ReLU neurons, to eliminate the non-linear components. However, this method needs to optimize the associated slopes and intercepts of the replaced linear activations to restore model performance.

With the continuous update of the structured pruning algorithm, whether it's layer-based or filter-based, the original multiple pruning and fine-tuning are developed to only be needed once. However, pruning only once needs to find the unimportant parameters accurately, which requires the development of more advanced algorithms to filter out the redundant parts of the network. Nevertheless, pruning only once is still a trend for future research.

3.2. Unstructured pruning

The unstructured pruning is to shield the unimportant neurons [40], as shown in Fig 4. Unimportant neurons usually refer to the parameters which contribute little to the network, taking values close to zero. At the same time, the connections between the pruned neurons and other neurons are ignored in the computation.

In 1989, LeCun et al. [41] put forward optimal brain damage, which uses second-derivative information to make a tradeoff between network complexity and training set error. In this way, unimportant weights are removed from the network. With the continuous development of pruning technology. In 2015, Han et al. [42] described a method, train-prune-retrain, to reduce the storage and computation of neural networks by learning only the important connections. The performance is improved by an order of magnitude without affecting their accuracy. The proposal applies regularization on weights of DNNs to learn the connectivity which result in sparse connections which can be used to distinguish the important and redundant connections. Then, the redundant connections are screened out, and then they can be removed. In addition, comparison experiments are made to adopt L1 regularization and L2 regularization, respectively.

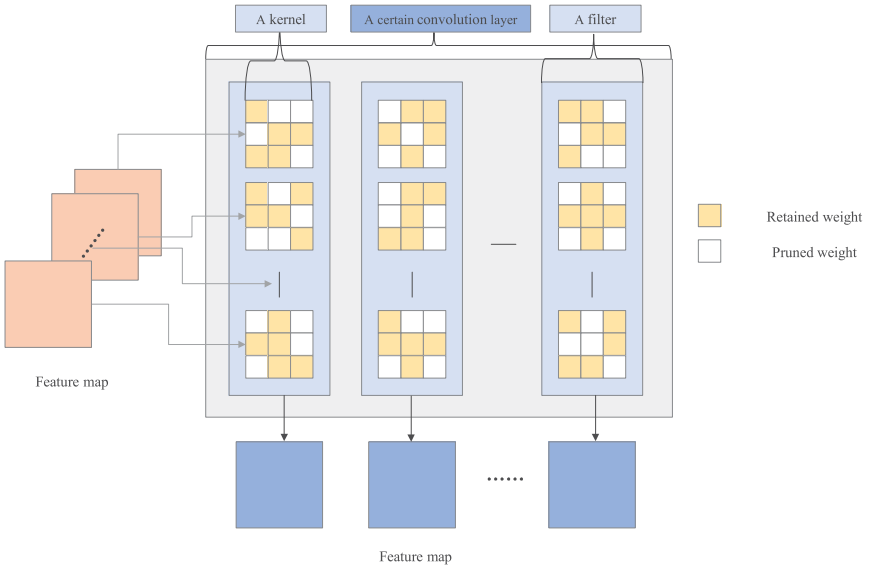

Figure 4: Unstructured pruning.

According to the experimental results, pruning with L1 regularization gets better accuracy than L2 regularization after pruning without retraining. This is due to that L1 regularization converts more parameters closer to zero. However, L2 regularization outperforms L1 regularization after pruning with retraining since this process does not benefit from pushing the value close to zero.

To improve the real-time pruning. In 2016, Guo et al. [43] proposed dynamic network surgery, which reduces network complexity significantly by pruning connections in real-time. Unlike previous approaches that greedily accomplished this task. They appropriately included connection splicing throughout the process to avoid incorrect pruning. By adding a learning process to the process of filtering important and unimportant parameters, it is possible to better find those parameters that are important. In 2021, Rosenfeld et al. [44] developed a scaling law that accurately estimates the error when pruning a single network with interactive magnitude pruning. Employing an invariant, it is possible to allow error-preserving interchangeability among depth, width, and pruning density.

Unstructured pruning greatly reduces the number of parameters and the theoretical computation of the model. However, the unstructured pruning is to set the redundant neurons to zero at present rather than remove these parts from the network, which simple generates a sparse network featuring in irregularity [45]. As a result, the non-regular sparsity is hard to be fully utilized to accelerate the model with current hardware architectures. Therefore, the pruned network by utilizing unstructured pruning techniques remains further study to be accelerated for current DNN hardware platforms.

4. Conclusion

In this paper, we present a survey on the research of model pruning for DNNs. Firstly, the detailed pruning process is demonstrated. Then, the current research on pruning techniques is introduced, which are classified into two categories based on the pruning targets in DNNs: the structured pruning techniques and the unstructured pruning techniques. In conclusion, the technique of model pruning is to remove redundant connections and neurons in the network, which further compresses and accelerates the running speed of DNNs. Both the advantages and the disadvantages of the current pruning methods are analyzed. The investigation provides exhaustive references to researchers and promote the further development of model pruning technology. Although the research on pruning techniques has made a series of achievements at present, it still has certain defects, such as the computational and time cost of multiple training and fine-tuning which are boring, and the complicated algorithms for screening out the redundant parts of DNNs are deficiency, and so on. Hence, the effectiveness of model pruning for DNNs still deserves further study.

References

[1] X. Yue, H. Li, Y. Fujikawa, L. Meng, Dynamic dataset augmentation for deep learning-based oracle bone inscriptions recognition, ACM Journal on Computing and Cultural Heritage (JOCCH) (2022).

[2] Y. Fujikawa, H. Li, X. Yue, C. Aravinda, G. A. Prabhu, L. Meng, Recognition of oracle bone inscriptions by using two deep learning models, International Journal of Digital Humanities (2022) 1–15.

[3] L. Meng, T. Hirayama, S. Oyanagi, Underwater-drone with panoramic camera for automatic fish recognition based on deep learning, IEEE Access 6 (2018) 17880–17886.

[4] X. Yue, H. Li, K. Saho, K. Uemura, C. Aravinda, L. Meng, Machine learning based apathy classification on doppler radar image for the elderly person, Procedia Computer Science 187 (2021) 146–151.

[5] S. Wen, M. Deng, A. Inoue, Operator-based robust non-linear control for gantry crane system with soft measurement of swing angle, International Journal of Modelling, Identification and Control 16 (2012) 86–96.

[6] A. Wang, M. Deng, Robust nonlinear multivariable tracking control design to a manipulator with unknown uncertainties using operator-based robust right coprime factorization, Transactions of the Institute of Measurement and Control 35 (2013) 788–797.

[7] Q. Zheng, X. Tian, M. Yang, H. Su, Clmip: cross-layer manifold invariance based pruning method of deep convolutional neural network for real-time road type recognition, Multidimensional Systems and Signal Processing 32 (2021) 239–262.

[8] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, Z. Li, Yolov4-5d: An effective and efficient object detector for autonomous driving, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–13.

[9] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).
[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[13] H. Li, Z. Wang, X. Yue, et al., An architecture-level analysis on deep learning models for low-impact computations, Artif Intell Rev (2022). doi:10.1007/s10462-022-10221-5.
[14] H. Li, Z. Wang, X. Yue, W. Wang, T. Hiroyuki, L. Meng, A comprehensive analysis of low-impact computations in deep learning workloads, in: Proceedings of the 2021 on Great Lakes Symposium on VLSI, 2021, pp. 385–390.
[15] G. Hinton, O. Vinyals, J. Dean, et al., Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 2 (2015).
[16] H. Li, X. Yue, Z. Wang, Z. Chai, W. Wang, H. Tomiyama, L. Meng, Optimizing the deep neural networks by layer-wise refined pruning and the acceleration on fpga, Computational Intelligence and Neuroscience 2022 (2022).
[17] R. Reed, Pruning algorithms-a survey, IEEE transactions on Neural Networks 4 (1993) 740–747.
[18] Z. Liu, M. Sun, T. Zhou, G. Huang, T. Darrell, Rethinking the value of network pruning, arXiv preprint arXiv:1810.05270 (2018).
[19] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, arXiv preprint arXiv:1611.06440 (2016).
[20] A. Aghasi, A. Abdi, N. Nguyen, J. Romberg, Net-trim: Convex pruning of deep neural networks with performance guarantee, Advances in neural information processing systems 30 (2017).
[21] S. Kundu, M. Nazemi, M. Pedram, K. M. Chugg, P. A. Beerel, Pre-defined sparsity for low-complexity convolutional neural networks, IEEE Transactions on Computers 69 (2020) 1045–1058.
[22] N. Lee, T. Ajanthan, P. H. Torr, Snip: Single-shot network pruning based on connection sensitivity, arXiv preprint arXiv:1810.02340 (2018).
[23] H. Tanaka, D. Kunin, D. L. Yamins, S. Ganguli, Pruning neural networks without any data by iteratively conserving synaptic flow, Advances in Neural Information Processing Systems 33 (2020) 6377–6389.
[24] J.-H. Luo, J. Wu, Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference, Pattern Recognition 107 (2020) 107461.
[25] Q. Xiang, X. Wang, Y. Song, L. Lei, R. Li, J. Lai, One-dimensional convolutional neural networks for high-resolution range profile recognition via adaptively feature recalibrating and automatically channel pruning, International Journal of Intelligent Systems 36 (2021) 332–361.
[26] H. Li, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, Pruning filters for efficient convnets, arXiv preprint arXiv:1608.08710 (2016).
[27] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang, B. Zhang, Accelerating convolutional networks via global & dynamic filter pruning., in: Proceedings of the Twenty-Seventh International Joint

Conference on Artificial Intelligence (IJCAI), volume 2, Stockholm, 2018, pp. 2425–2432.
[28] J. Kuang, M. Shao, R. Wang, W. Zuo, W. Ding, Network pruning via probing the importance of filters, International Journal of Machine Learning and Cybernetics (2022) 1–12.
[29] J. Chang, Y. Lu, P. Xue, Y. Xu, Z. Wei, Automatic channel pruning via clustering and swarm intelligence optimization for cnn, Applied Intelligence (2022) 1–21.
[30] S. S. Sawant, J. Bauer, F. Erick, S. Ingaleshwar, N. Holzer, A. Ramming, E. Lang, T. Götz, An optimal-score-based filter pruning for deep convolutional neural networks, Applied Intelligence (2022) 1–23.
[31] U. Evci, T. Gale, J. Menick, P. S. Castro, E. Elsen, Rigging the lottery: Making all tickets winners, in: International Conference on Machine Learning, PMLR, 2020, pp. 2943–2952.
[32] Q. Huang, K. Zhou, S. You, U. Neumann, Learning to prune filters in convolutional neural networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 709–718.
[33] Y. Chu, P. Li, Y. Bai, Z. Hu, Y. Chen, J. Lu, Group channel pruning and spatial attention distilling for object detection, Applied Intelligence (2022) 1–19.
[34] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2736–2744.
[35] T.-J. Yang, Y.-H. Chen, V. Sze, Designing energy-efficient convolutional neural networks using energy-aware pruning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5687–5695.
[36] Y. Fan, W. Pang, S. Lu, Hfpq: deep neural network compression by hardware-friendly pruning-quantization, Applied Intelligence 51 (2021) 7016–7028.
[37] T. Chen, B. Ji, T. Ding, B. Fang, G. Wang, Z. Zhu, L. Liang, Y. Shi, S. Yi, X. Tu, Only train once: A one-shot neural network training and pruning framework, in: Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 19637–19651.
[38] G. S. Chung, C. S. Won, Filter pruning by image channel reduction in pre-trained convolutional neural networks, Multimedia Tools and Applications 80 (2021) 30817–30826.
[39] T. Chen, H. Zhang, Z. Zhang, S. Chang, S. Liu, P.-Y. Chen, Z. Wang, Linearity grafting: Relaxed neuron pruning helps certifiable robustness, arXiv preprint arXiv:2206.07839 (2022).
[40] X. Dong, S. Chen, S. Pan, Learning to prune deep neural networks via layer-wise optimal brain surgeon, Advances in Neural Information Processing Systems 30 (2017).
[41] Y. LeCun, J. Denker, S. Solla, Optimal brain damage, Advances in neural information processing systems 2 (1989).
[42] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network, Advances in neural information processing systems 28 (2015).
[43] Y. Guo, A. Yao, Y. Chen, Dynamic network surgery for efficient dnns, Advances in neural information processing systems 29 (2016).
[44] J. S. Rosenfeld, J. Frankle, M. Carbin, N. Shavit, On the predictability of pruning across scales, in: International Conference on Machine Learning, PMLR, 2021, pp. 9075–9083.
[45] J. Frankle, M. Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks, arXiv preprint arXiv:1803.03635 (2018).