IEICE TRANS. INF. & SYST., VOL.E107–D, NO.7 JULY 2024

LETTER

Channel Pruning via Improved Grey Wolf Optimizer Pruner

Xueying WANG[†], **Yuan HUANG**[†], **Xin LONG**[†a)], ***Nonmembers***, ***and*** **Ziji MA**[††], ***Member***

SUMMARY In recent years, the increasing complexity of deep network structures has hindered their application in small resource constrained hardware. Therefore, we urgently need to compress and accelerate deep network models. Channel pruning is an effective method to compress deep neural networks. However, most existing channel pruning methods are prone to falling into local optima. In this paper, we propose a channel pruning method via Improved Grey Wolf Optimizer Pruner which called IGWO-Pruner to prune redundant channels of convolutional neural networks. It identifies pruning ratio of each layer by using Improved Grey Wolf algorithm, and then fine-tuning the new pruned network model. In experimental section, we evaluate the proposed method in CIFAR datasets and ILSVRC-2012 with several classical networks, including VGGNet, GoogLeNet and ResNet-18/34/56/152, and experimental results demonstrate the proposed method is able to prune a large number of redundant channels and parameters with rare performance loss.

key words: *channel pruning, convolutional neural networks, Grey Wolf algorithm, fitness*

1. Introduction

In recent years, deep learning, especially deep neural networks, has played an important role in various aspects of society [1], [2]. However, the increasing demands in computing power and memory footprint of deep network has hindered their application in small resource constrained hardware. Considerable efforts have been proposed to address this problem, including compact architecture designment [3]–[5], parameter decomposition [6], knowledge distillation [7]–[9], quantization [10]–[12], pruning [13]–[16]. Among them, channel pruning has been considered as one of the most effective methods for model compression and is easy to implement for convolutional neural networks while other approaches not.

The goal of channel pruning is to compress the number of channels in each layer of the original structure, ultimately minimizing the accuracy degradation or even achieving better accuracy of the overall network structure. In this paper, we propose a novel channel pruning method via Improved Grey Wolf algorithm [17] which called IGWO-Pruner to prune redundant channels of convolutional neural networks. It identifies pruning ratio of each layer by using Improved Grey Wolf algorithm, and then fine-tuning the new pruned network to obtain final compact model. The flow chart of the proposed algorithm is shown in Fig. 1.

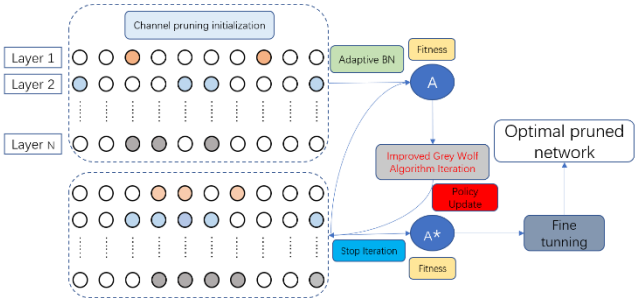

Fig. 1 Pipeline of IGWO-Pruner.

In Fig. 1, Adaptive Batch Normalization (Adaptive BN) is sourced from Ref. [18], [18] proposed a method that can quickly measure the performance of pruned models through Adaptive BN. In Fig. 1, the circle represents different elements of channels. Different colors mean that the pruning ratio of the channels' elements are different, which will affect whether they are pruned and thus affect the pruning strategy.

2. The Proposed IGWO-Pruner

In this section, we propose our method to achieve channel pruning, which called Improved Grey Wolf Optimizer Pruner (IGWO-Pruner). The pipeline is shown in Fig. 1. Assuming a general deep convolutional network S containing n layers, its original structure is represented as $S = [c_1, c_2, \cdots, c_n]$, where c_i (i = 1, . . . , n) represents the number of channels in the n-th layer of the network.

2.1 Description of Deep Network Pruning Problem

The network structure obtained after pruning the original network is $S' = [c'_1, c'_2, \cdots, c'_n]$, where $c'_n < c_n$. The pruning rate for each layer of the network is set to $R = [r_1, r_2, \cdots, r_n]$, where $r_i = c'_i/c_i$, the optimization problem is to find the optimal pruning rate r while obtaining the optimal network inference accuracy under a given test and training set. Therefore, this problem can be summarized as follows:

$$R^* = arg_r \max acc(S'(r)) \tag{1}$$

where $acc(S'(r))$ represents the inference accuracy obtained after pruning and adaptive batch standardization processing

Manuscript received January 18, 2024.
Manuscript publicized March 7, 2024.

[†]The authors are with College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China.
[††]The author is with College of Electronical and Information Engineering, Hunan University, Changsha 410073, China.
a) E-mail: longxin14@nudt.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2024EDL8007

using pruning rate r for each layer of the model. However, in order to obtain the optimal solution of the above equation, it involves high-dimensional optimization and the process is very complex. In order to simplify the computational cost of search, the method proposed in this article constrains the pruning rate r_i (where r_i is in %, represents the i_{th} channel's pruning rate) within {0,10%, 20%, . . . , 100%}, making the above solving problem a search optimal combination problem. This constraint condition can greatly reduce the combination of pruning structures, and the final number of search set elements becomes 11^n, making the solution of Eq. (1) more efficient. In order to further solve the above optimization problem, we use an Improved Grey Wolf algorithm to search for the optimal pruning rate.

2.2 Pruning Algorithm Process

To avoid previous channel pruning algorithms based on channel importance falling into local optima, we propose an automatic search pruning algorithm that considers network structure pruning from the perspective of the entire network. We use the Improved Grey Wolf algorithm to search for the optimal pruning rate for each convolutional layer. The search process of this algorithm is shown in the Fig. 2.

Search initialization: Due to the fact that the Improved Grey Wolf algorithm is less affected by initial values, we use the random number method to generate the initial search population:

$$X_{i,j} \sim U(lb, ub) \tag{2}$$

where $X_{i,j}$ represents the gray wolf population involved in the search, i is the number of gray wolf population and $\in \{1, 2, \ldots, N\}$, j is the population dimension (which represents the number of network layers), U is a random function, lb and ub are the upper and lower bounds of the search interval (the search space is {0%, 10%, 20%, . . . , 100%}).

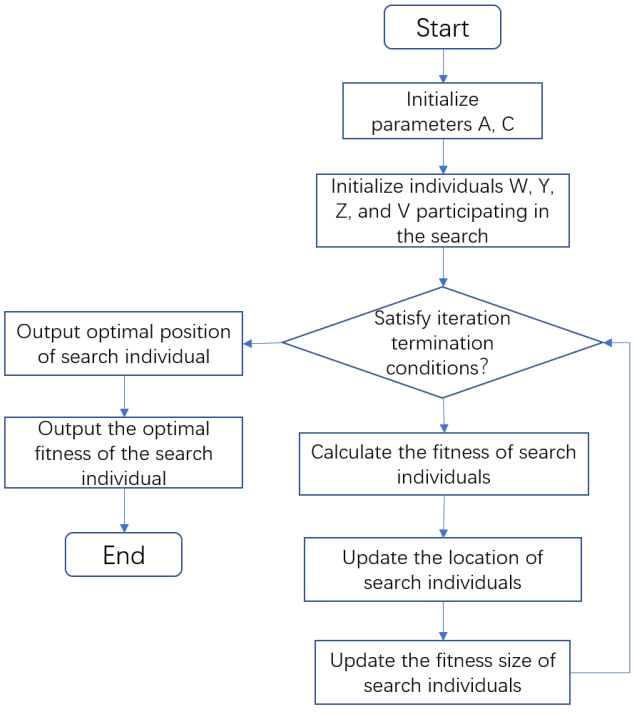

Fig. 2 Diagram of the Improved Grey Wolf algorithm search process.

Search process: All gray wolves approach their prey in the following ways, which leads to an optimal solution:

$$D = |C \cdot X_p(t) - X(t)| \tag{3}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{4}$$

Where D represents the distance between the gray wolf and its prey, t is the current number of iterations, $X_p(t)$ and $X(t)$ are the prey position and the wolf position, A and C are the adjustment coefficients, can be calculated as follows:

$$A = 2a \cdot r_1 - a \tag{5}$$

$$C = 2 \cdot r_2 \tag{6}$$

where r_1 and r_2 is a random vector between 0 and 1, $a = 2e^{-t/T}$, and T is the maximum number of iterations set by the algorithm.

Parameter update: By obtaining the pruning rates of the network layers corresponding to the position vectors of each gray wolf, the L1 norm of each channel in each layer is calculated. Channels with lower L1 norm parameter values in each layer of the network are pruned. Then, the pruning model is updated with adaptive batch standardization methods to update the batch standardization layer. The obtained inference accuracy is used as the fitness of gray wolf individuals, with the optimal individual marked as W, the optimal individual marked as Y, and the suboptimal individual marked as Z, the rest are V.

During the update, due to the individual's optimality, candidate gray wolves update their position by calculating the movement distances D_W, D_Y, and D_Z with gray wolves W, Y, and Z, respectively. The relevant calculation formula is as follows:

$$D_W = |C_1 \cdot X_W - X(t)| \tag{7}$$

$$D_Y = |C_2 \cdot X_Y - X(t)| \tag{8}$$

$$D_Z = |C_3 \cdot X_Z - X(t)| \tag{9}$$

$$\begin{aligned} X(t+1) = (1/3)((X_W - A_1 D_W) + (X_Y - A_2 D_Y) \\ + (X_Z - A_3 D_Z))(1 - t/T) \\ + (X_W - A_1 D_W) \cdot (t/T) \end{aligned} \tag{10}$$

Where $X(t)$ represents the current candidate gray wolf position, $X(t+1)$ is the next candidate gray wolf position, and X_W, X_Y and X_Z represents the current positions of W wolf, Y wolf, and Z wolf respectively. A_1, A_2, A_3 and C_1, C_2, C_3, like A and C in above search process, are random variables.

2.3 Channel Pruning and Fine-Tuning

We add L1 norm to the Loss function to constrain the weight.

From the perspective of optimizing the objective function, L1 norm can make most of the weights 0, which makes the weights of network channels have a certain sparsity, so that related channels can be pruned. The objective function is:

$$Loss = Loss + \gamma \sum_{w \in K} \|w\|_1 \tag{11}$$

Where Loss is the standard loss function of deep network, K is the network weight set, w is the element in the set K, and $\|\bullet\|_1$ is the L1 norm, γ is the penalty factor. Besides, pruning extra channel would lead to some accuracy loss when the pruning percentile is pretty high. In experimental sections, this can be largely compensated by fine-tuning process which needs less training epochs and time.

3. Experiments

In this section, we mainly conduct the effectiveness of proposed method on several representative network and datasets. We implement our method based on Pytorch.

3.1 Implementation Details

In this paper, we empirically conduct experiments on CIFAR-10 and ILSVRC-2012 [19]. The same standard data augmentation strategy in [20] is adopted by this paper. For network architectures, we evaluate proposed method on some frequently-used network: VGGNet, GoogLeNet and ResNet-18/34/56/152. During training process, we use the Stochastic Gradient Descent algorithm (SGD) for fine-tuning with momentum 0.9 and the batch size is set to 256. We also bring several evaluation terms which will be used in the following parts, like Channel number, FLOPs (floating point operations) and parameters, which are used to measure the network pruning and compression.

3.2 Results and Discussions

CIFAR-10: We conduct our experiments on CIFAR-10 with three deep networks including VGGNet, GoogLeNet and ResNet-56. The results are shown in Table 1.

As shown in Table 1, it could be seen that the proposed method can achieve significant pruning of channel numbers, parameters, and computational complexity with minimal accuracy degradation.

Table 1 Accuracy and pruning results on CIFAR-10.

Model	Acc	Channel Pruned	FLOPs Pruned	Parameters Pruned
VGGNet-16	93.13%	0.00%	0.00%	0.00%
VGGNet-16(Ours)	93.04%	65.32%	76.24%	89.56%
GoogLeNet	95.18%	0.00%	0.00%	0.00%
GoogLeNet(Ours)	95.31%	26.76%	69.29%	64.47%
ResNet-56	93.22%	0.00%	0.00%	0.00%
ResNet-56(Ours)	93.09%	25.42%	49.08%	50.39%

ILSVRC-2012: We further conduct our experiments on ILSVRC-2012 with some deep networks including ResNet-18/34/152. The results are shown in Table 2.

As shown in Table 2, ILSVRC-2012 is a large-scale dataset and contains 1,000 categories, which is much complex than the CIFAR-10 with only 10 categories, so it could be seen that the performance drops on ILSVRC-2012 are more than these on CIFAR-10. On the other hand, it comes that the proposed method obtains higher pruning rates and less accuracy drops as the depth of network increases.

Comparison with Other Methods: Reference [21] provides a good review and summary of Pruning Deep Neural Networks, we have selected from [21] several methods (in [22]–[24]) that are similar to our application field and pruning approach for algorithm performance comparison. Besides, we also selected some the state-of-art methods (in [25]–[28]) for algorithm performance comparison. The results in Table 3 show that IGWO-Pruner could obtain better FLOPs reduction or less accuracy loss, it seems that proposed channel pruning method would provide a great tradeoff be-

Table 2 Accuracy and pruning results on ILSVRC-2012.

Model	Acc-Top1	Acc-Top5	Channel Pruned	FLOPs Pruned	Parameters Pruned
ResNet-18	68.97%	88.23%	0.00%	0.00%	0.00%
ResNet-18(Ours)	66.45%	86.83%	14.27%	49.52%	21.03%
ResNet-34	72.57%	90.49%	0.00%	0.00%	0.00%
ResNet-34(Ours)	70.97%	89.34%	20.24%	39.83%	51.02%
ResNet-152	77.30%	93.05%	0.00%	0.00%	0.00%
ResNet-152(Ours)	76.88%	92.47%	15.98%	58.74%	57.45%

Table 3 Performance comparison with the state-of-art methods.

Method	Acc-baseline	Acc-pruned	FLOPs
CP[22]	76.45%	72.88%	2.73G
SSS[23]	76.45%	74.73%	2.82G
MP[24]	76.45%	75.49%	2.92G
CHEX[25]	76.45%	75.68%	2.72G
ABC-Pruner[26]	76.45%	74.84%	2.56G
LPSR[27]	76.45%	74.62%	2.98G
ELC[28]	76.45%	74.95%	2.89G
IGWO(Ours)	76.45%	75.55%	2.65G

tween model size and performance.

4. Conclusions

In this paper, we propose a novel channel pruning method based on improved Grey Wolf algorithm which called IGWO-Pruner to prune redundant channels of convolutional neural networks. It identifies proper pruning ratio of each layer by using intelligent search algorithm, and then fine-tuning the new pruned network model so as to compensate accuracy loss. Experimental results show that the proposed method can achieve great pruning results than existing pruning methods.

Acknowledgments

This work was supported in part by project funded by China National Natural Science Foundation of China, NSFC (62207030).

References

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.
[2] L. Yang, Y. Wang, X. Xiong, J. Yang, and A.K. Katsaggelos, "Efficient video object segmentation via network modulation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp.6499–6507, 2018.
[3] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," Computer Vision – ECCV 2018, ed. V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, pp.122–138, Springer International Publishing, Cham, 2018.
[4] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp.6848–6856, 2018.
[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp.4510–4520, 2018.
[6] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp.67–76, 2017.
[7] S. Zhou, Y. Wang, D. Chen, J. Chen, X. Wang, C. Wang, and J. Bu, "Distilling holistic knowledge with graph neural networks," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, pp.10367–10376, 2021.
[8] P. Luo, Z. Zhu, Z. Liu, X. Wang, and X. Tang, "Face model compression by distilling knowledge from neurons," Proc. Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, vol.30, no.1, pp.3560–3566, 2016.
[9] L. Lu, M. Guo, and S. Renals, " Knowledge distillation for small-footprint highway networks," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, pp.4820–4824, 2017.
[10] M. Courbariaux, Y. Bengio, and J. David, "Binary connect: Training deep neural networks with binary weights during propagations," I Proc. 28th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, pp.3123–3131, 2015.
[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," Computer Vision – ECCV 2016, ed. B. Leibe, J. Matas, N. Sebe, and M. Welling, pp.525–542, Springer International Publishing, Cham, 2016.
[12] Z. Li, B. Ni, W. Zhang, X. Yang, and W. Gao, "Performance guaranteed network acceleration via high-order residual quantization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp.2603–2611, 2017.
[13] S. Han, J. Pool, J. Tran, and W.J. Dally, "Learning both weights and connections for efficient neural networks," Proc. 28th International Conference on Neural Information Processing Systems - Volume 1, Cambridge, MA, USA, pp.1135–1143, 2015.
[14] J. Guo and M. Potkonjak, "Pruning ConvNets online for efficient specialist models," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, pp.430–437, 2017.
[15] X. Long, Z. Ben, X. Zeng, Y. Liu, M. Zhang, and D. Zhou, "Learning sparse convolutional neural networks via quantization with low rank regularization," IEEE Access, vol.7, pp.51866–51876, 2019.
[16] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp.2755–2763, 2017.
[17] S. Mirjalili, S.M. Mirjalili, and A. Lewis, "Grey Wolf optimizer," Adv. Eng. Softw., vol.69, pp.46–61, 2014.
[18] B. Li, B. Wu, J. Su, and G. Wang, "EagleEye: Fast sub-net evaluation for efficient neural network pruning," Computer Vision – ECCV 2020, ed. A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, pp.639–654, Springer International Publishing, Cham, 2020.
[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, vol.115, no.3, pp.211–252, 2015.
[20] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H.P. Graf, "Pruning filters for efficient ConvNets," arXiv preprint arXiv:1608.08710, 2016.
[21] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," IEEE Access, vol.10, pp.63280–63300, 2022.
[22] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp.1398–1406, 2017.
[23] Z. Huang and N. Wang, "Data-driven sparse structure selection for deep neural networks," Computer Vision – ECCV 2018, ed. V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, pp.317–334, Springer International Publishing, Cham, 2018.
[24] Z. Liu, H. Mu, X. Zhang, Z, Guo, X. Yang, K.-T. Cheng, and J. Sun, "Meta-pruning: Meta learning for automatic neural network channel pruning," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), pp.3295–3304, 2019.
[25] Z. Hou, M. Qin, F. Sun, X. Ma, K. Yuan, Y. Xu, Y.-K. Chen, R. Jin, Y. Xie, and S.-Y. Kung, "CHEX: Channel EXplotration for CNN model compression," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, pp.12277–12288, 2022.
[26] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, and Y. Tian, "Channel pruning via automatic structure search," Proc. 29th Int. Joint Conf. Artif. Intell., pp.673–679, July 2020.
[27] K. Zhang and G. Liu, "Layer pruning for obtaining shallower ResNets," IEEE Signal Process. Lett., vol.29, pp.1172–1176, 2022.
[28] J. Wu, D. Zhu, L. Fang, Y. Deng, and Z. Zhong, "Efficient layer compression without pruning," IEEE Trans. Image Process., vol.32, pp.4689–4699, 2023.