Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/52/20241292

An analysis of different methods for deep neural network pruning

Longxiang Gou[1,4,5], Ziyi Han[2], Zhimeng Yuan[3]

[1]Faculty of Science, University of Hong Kong, Hong Kong, 999077, China
[2]School of Mathematics, Northwest University, Xian, Shaanxi, 710127, China
[3]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China

[4]u3563712@hku.hk
[5]corresponding author

Abstract. Neural network pruning, the process of removing unnecessary weights or neurons from a neural network model, has become an essential technique for reducing computational cost and increasing processing speed, thereby improving overall performance. This article has grouped current pruning methods into three classes—channel pruning, filter pruning, and parameter sparsification—and discussed how each method works. Each approach has its own strengths: channel pruning is particularly useful for reducing model depth and width, filter pruning is more suitable for maintaining model depth while decreasing storage requirements, and parameter sparsification can be applied across various network architectures to achieve both storage and computational efficiency. This work will delve into how each method works and highlight key related works of each category. In the future, it is expected that future research in neural network pruning could focus on developing more sophisticated techniques that can automatically identify important weights or neurons within a network.

Keywords: Neural network pruning, Deep learning, Parameter sparsification.

1. Introduction

Neural network pruning is a crucial model optimization technique aimed at enhancing the efficiency and practicality of neural networks through careful parameter reduction. This technique finds widespread applications across various domains, including computer vision and natural language processing.

During the process of neural network pruning, both the architecture and weights of the model may undergo changes, encompassing structural pruning and weight reduction. To achieve this objective, several methods and strategies are available. Among them, a common approach is importance-based pruning. This method initially assesses the connections, neurons, or parameters within the network, typically by measuring their impact on the model's output to determine their importance. Subsequently, based on these evaluations, selectively removing less influential components can reduce the model's size, saving storage space, and reducing computational requirements for inference, thus improving the model's performance in resource-constrained environments.

However, it is essential to note that pruning is not merely about deleting elements of the model but requires a delicate balance while maintaining overall performance. Inappropriate pruning can lead to a

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/52/20241292

decline in model performance. Therefore, after pruning operations, it is typically necessary to conduct retraining or fine-tuning to restore model accuracy. Furthermore, as pruning operations progress, the model's redundancy gradually decreases, making subsequent pruning operations more challenging.

In summary, neural network pruning, as a technology aimed at optimizing model performance and efficiency, holds significant importance for deploying deep learning models in resource-constrained environments. Through appropriate pruning operations, it is possible to achieve model lightweighting, accelerated inference, and resource savings, providing more efficient and practical solutions across various application domains. Therefore, this work will delve into the progress and evolution of neural network pruning by grouping them into three overall architectures based on different approaches to neural network pruning. In the second part of this paper, channel pruning will be elaborated, filter pruning will be introduced in the third part, parameter sparsification will be explored in the fourth part, and finally a succinct summary and future outlook will be provided.

2. Channel Pruning

When it comes to lightweighting and optimizing Convolutional Neural Networks (CNNs), channel pruning emerges as a critical technique. Its core idea is to reduce model complexity, save computational resources, and, when possible, maintain model performance by removing some channels (also known as feature maps or convolutional kernels) within the convolutional layers. Here are the detailed steps involved in channel pruning:

2.1. Feature Map Importance Evaluation

Before pruning, it is essential to evaluate the feature maps in each convolutional layer to understand their contributions to network performance. Typically, it is common to use a metric to measure the importance of feature maps, such as the average activation values or gradients of feature maps. Less active feature maps may indicate their minor impact on the model's output. He Y, Zhang X, and Sun J achieved this by building a linear least squares (least squares OR least squares) based implementation [1]. In contrast to traditional approaches, another article proposes a joint dynamic pruning algorithm to evaluate the features of the convolution kernel and the input [2]. On the one hand, some of the convolutional kernels are zeroed and allowed to update the kernel state during training until the network converges and then the zeroed convolutional kernels are permanently removed. On the other hand, features of the input image are sampled and then these features are analysed using a channel importance prediction network to identify channels that can be skipped in the convolution operation.

2.2. Pruning Decision

Based on the importance assessment of feature maps, selectively prune less important channels. Often, less important channels are marked as pruning targets. This can be achieved by setting a threshold or retaining a fixed number of channels. The key to pruning decisions is to strike a balance to reduce parameters while preserving the network's representational capacity as much as possible. He Y, Zhang X, and Sun J uses Least Absolute Shrinkage and Selection Operator (LASSO) regression, that is, adding L1 paradigms to the loss function to constrain the weights to complete the loss function to the objective function optimisation point of view, the L1 paradigm can make the majority of the values in the weights to be 0, so that the weights in the channel have a sparsity, which can be the coefficients of the channel clipping [1]. In another work, the authors are not only using L1 regularisation, but also applying L1 regularisation to the scaling factor gamma of the BN layer, i.e., considering that the closer the gamma is to 0, the lower the importance of the corresponding output to the result [3]. Certainly, Lin M et al. introduces a novel channel pruning approach which is based on the Artificial Bee Colony (ABC) algorithm, known as ABCPruner [4]. Its aim is to effectively determine the optimal pruning structure, i.e., the quantity of channels in each layer, rather than selecting "important" channels as done in prior research. To address the vast number of possible pruning structure combinations in deep networks, the authors propose initially constraining the search space for retained channels, significantly reducing the combinations of pruning structures. Subsequently, the search for the optimal pruning structure is

DOI: 10.54254/2755-2721/52/20241292

transformed into an optimization problem and automated using the ABC algorithm to minimize human intervention.

2.3. Model Pruning

After determining the pruning targets, actual pruning operations are carried out, involving the removal of selected channels from the convolutional layer. This results in a reduction in the number of convolutional kernels, consequently reducing the parameter count and computational load. It is important to note that the pruning strategy required for different models seriously affects their effectiveness, for example, as mentioned previously in Liu Z et al.'s article, when dealing with networks such as ResNet and DenseNet that are connected across layers, its strategy does not demonstrate good results because the output of each layer is used as input for multiple subsequent layers and its BN is done before convolution [3]. In this case, the sparsification is obtained at the end of the layer's input, and a layer selectively accepts a subset of all channels to do the next convolution operation. To save parameters and computation during testing, the work need to place a channel selection layer identifying the important channels.

2.4. Fine-tuning and Re-training

Pruned models typically experience some performance degradation because removing channels can lead to information loss. To recover performance, fine-tuning or re-training is necessary for the pruned model. Fine-tuning involves training the pruned model for several rounds to adapt it to the changes introduced by pruning.

Of course, today's projects often use ease of implementation and effectiveness as the goals of choice when completing channel pruning tasks, so Liu Z et al.'s work is being used more broadly [3].

3. Filter Pruning

Filter pruning is a crucial part of the pruning methods in the neural networks pruning. The main idea is to improve the weights in the convolution process so that the model complexity can be reduced, and computation speed can be accelerated [5]. In order to reach this purpose, some filters, usually the ones with relatively less importance, will be pruned. After the number of filters is reduced, the number of channels will decrease as well. In the end, the number of final output parameters will also drop. This method may be a little similar to channel pruning since they both reduce the number of channels in the pruning process. However, the starting points of these two methods are different.

Although filter pruning has a lot of branches, most of the pruning methods follow the following three steps as stated in He Y et al.'s work [6]. They are Training, Pruning and Retraining. In the first step, a large model is fitted on a given large dataset. Then, the filters with lower importance level are dropped according to a specific criterion during the pruning step. In the last step, the model after pruning is retrained so that the original performance can be recovered.

3.1. Filter Pruning via Geometric Median (FPGM)

The study of filter pruning has been a quite popular subject. Therefore, there are a lot of articles which seek to develop improvement from the previous methods. Among them, the method of filter pruning via geometric median (FPGM) is recently proposed to solve the problem which usually rise from the norm-based criterion [7]. This problem comes from the two requirements, the level of norm deviation of the filters stays high, and the minimum norm of the filters remains low, which sometimes cannot be met at the same time. The method of FPGM is developed so that this problem can be solved. It does not prune the filters with less importance level as many previous methods might do. Instead, it prunes the filters with less redundancy. This method allows the pruning process to be less computationally expensive and the model to be less complex [7].

The FPGM method is evaluated for both single-branch network as well as multiple-branch network. This is done on two popular datasets which are CIFAR-10 and ILSVRC-2012 [7]. For the single-branch network pruning, FPGM is tested on the dataset CIFAR-10 in comparison with VGGNet, which is a

DOI: 10.54254/2755-2721/52/20241292

famous single-branch network pruning model. For the multiple-branch network pruning, it is tested on two datasets which are CIFAR-10 and ILSVRC-2012. In the CIFAR-10 group, FPGM is compared with ResNet-20, 32, 56 and 110 with two different pruning rates which are 30% and 40%. As for the dataset ILSVRC-2012, it is compared with ResNet-18, 34, 50 and 101 with pruning rates 30% and 40% [7].

In the single-branch network pruning group, the accuracy and number of parameters and required Float Points Operations (FLOPs) of FPGM is higher than that of VGGNet, suggesting that FPGM gives a better performance in both prediction accuracy as well as computational cost. In the multiple-branch network pruning group, although the situation is a bit more complicated. FPGM succeeds to show a generally better performance than other methods in both datasets [7].

3.2. High Rank Feature Map

In another article which is done by Lin M et al., another way of improving the traditional filter pruning is raised [8]. This article proposes a pruning method by exploring the High rank (HRank) of feature map. The inspiration of this method is the stability of the average rank of multiple feature map generated by a single filter. The average rank which comes from a single filter does not change. A method to prune the filters with low rank feature map is then developed. Because less information is included in low-rank feature map, it makes the pruning result quite easy to be reproduced. This is the idea behind the HRank feature map method. In addition, the author of this article also shows that more information is contained in the high-rank feature map. This makes the damage to the whole model very little even when a portion is not updated in the pruning process.

In order to show that this newly developed method could improve the pruning methods, it is compared with other famous pruning model on some popular datasets. The two datasets involved in this experiment are CIFAR-10, which is already introduced in the previous part of this article, and ImageNet.

ImageNet is a large-scale dataset with more than 15,000,000 images in around 22000 classes. The method of HRank is compared with some mainstream CNN models such as VGGNet, GoogLeNet, ResNet and DenseNet on both datasets. The protocols of FLOPs, top-1 accuracy and top-5 accuracy are used to determine which model gives a better result.

The final result of the experiment is very complex since it includes many different comparisons over two datasets. Overall, the HRank pruning method gives a generally better performance than the other mainstream CNN models. It shows a higher rate of accuracy and higher level of FLOPs, which suggests that this method indeed can reduce the computational cost and model complexity and increase the prediction accuracy [8].

3.3. Model Comparison

The two pruning methods mentioned above both do a great job in reducing the output parameters and increasing the accuracy level. However, HRank method shows a higher FLOPs in the comparison with ResNet-56/110 in CIFAR-10 than FPGM, while FPGM achieves a higher accuracy level [7,8]. This may suggest that FPGM gives more accurate result than HRank and HRank may be less computationally expensive than FPGM.

4. Parameter Sparsification

Instead of removing weights one at a time, structured sparsification of the convolution kernel's parameters takes into account the tensor's overall structure, which speeds up computation and reduces data reading.

4.1. Fusion of LASSO and Singular Value Decomposition (SVD)

Wu, J et al. proposes Singular Value Decomposition (SVD) to process the structure of the tensor, where mathematically the singular value decomposition of a matrix is unique, thus allowing a large tensor to be decomposed into multiple smaller tensors thereby achieving tensor arithmetic sparsification [9]. They use the model trained on ImageNet dataset using VGG-16 as the basis for migration learning on Oxford flowers_102 dataset, and compare the resolution obtained by PCA dimensionality reduction of the

DOI: 10.54254/2755-2721/52/20241292

dataset with the resolution obtained by fusing SVD lightened model based on Lasso regression channel pruning [9]. The result is that the model fused with SVD has a higher resolution for the images, and since channel pruning inherently reduces the number of parameters in the model, the final model accuracy using SVD is higher with a reduced number of parameters, indicating that SVD has less loss of important parameters in the model. For the effectiveness of SVD in reducing the spatial complexity, Wu, J et al. proposes to take the caffemodel as the original model and compare the number of parameters of the model based on Lasso regression channel pruning with that of the model of Lasso regression channel pruning fused with SVD, which objectively and concretely embodies the effectiveness of SVD in reducing the storage requirements of the model [9].

4.2. Tensor Decomposition Compression

K. K. Kung et al. suggest using the mean and variance as the index to gauge the significance of the filter based on SVD in order to lighten the large network and increase the computing time and accuracy of the large network [10]. The edge of an image is the most basic feature that constitutes an image, and the effect of convolution kernel edge extraction directly affects image recognition and understanding; in their analysis, they found that the convolution kernel with the highest mean value extracts to the low-frequency signals, the convolution kernel with the highest variance sharpens the image and extracts to the high-frequency signals, and the filter with the highest variance is better able to extract to the edge characteristics of an image [10]. Therefore, they consider that the filter with the lower mean and variance in the convolutional network is a redundancy [10]. Thus, K. K. Kung et al. considers the filters with smaller mean and variance in the convolutional network as redundant filters, and uses clustering to separate the filters with smaller mean and variance for trimming, and then reconstructs the convolutional layer using the retained filters [10]. In order to validate the above method, the Lenet5 network pruned in Minst dataset in K. K. Kung et al.'s work, the pruning rate of this method is as high as 25%, and compared with other mainstream pruning methods, which verifies that the pruning method proposed by them can maintain high image recognition accuracy with high efficiency of pruning [10].

4.3. Model Compression

A deep network model with a large number of parameters and a significant number of redundant parameters, such as the Faster RCNN algorithm, one of the top algorithms in the field of target detection, must be pruned in order to reduce time and space complexity while maintaining model accuracy. The second article takes the VGG16-based Faster RCNN as the research object, and uses the "pruning method with SVD relying on the mean and variance as the evaluation criterion" to compress the deep convolutional network Faster RCNN, and experimentally compares it with the channel pruning fusion SVD based on Lasso regression proposed in the first article, and the outcomes of the experiments demonstrate that it is the deep convolutional network's superior technique. The experimental results show that the parameter decrease rate and acceleration ratio of the method in the second article are significantly higher than that of the method in the first one, and the decrease rate of the AP value is significantly lower than that of the first one [9,10]. The same SVD is also applied, but the mean and variance pruning criterion proposed in K. K. Kung et al.'s work is better than that of the Lasso regression-based channel pruning in the first article for the pruning of the deep network, i.e., the parameter sparsification of the structured of convolutional kernel parameter sparsification will be superior to channel pruning, and better able to improve the performance of the network under limited resources [9,10].

5. Conclusion

This paper presents an overview of the development and application of neural network structured pruning techniques across various models. Structured pruning is a powerful approach for deleting redundant parameters in neural networks with a large number of parameters while ensuring model accuracy. It has been shown to effectively reduce the storage space required by the model, and improve the model's computational efficiency leading to faster training and inference times. It has also been

DOI: 10.54254/2755-2721/52/20241292

demonstrated that structured pruning can achieve better performance than traditional unstructured pruning methods in some scenarios. Additionally, the application of structured pruning has been extended to a wide range of neural network models, including convolutional neural networks, recurrent neural networks (RNNs), and transformers. These application scenarios cover various fields, such as image classification, speech recognition, and natural language processing.

Authors contribution

All authors contributed equally to this research, and their names are listed in alphabetical order.

References

[1] He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE international conference on computer vision, 1389-1397.

[2] Zhang, M., Lu, Q., Li, W., & Song, H. (2021). Deep neural network compression algorithm based on combined dynamic pruning. Journal of Computer Applications, 41(6), 1589.

[3] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE international conference on computer vision, 2736-2744.

[4] Lin, M., Ji, R., Zhang, Y., Zhang, B., Wu, Y., & Tian, Y. (2020). Channel pruning via automatic structure search. arXiv preprint arXiv:2001.08565.

[5] Luo, J. H., Wu, J., & Lin, W. (2017). Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the IEEE international conference on computer vision, 5058-5066.

[6] He, Y., Ding, Y., Liu, P., Zhu, L., Zhang, H., & Yang, Y. (2020). Learning filter pruning criteria for deep convolutional neural networks acceleration. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2009-2018.

[7] He, Y., Liu, P., Wang, Z., Hu, Z., & Yang, Y. (2019). Filter pruning via geometric median for deep convolutional neural networks acceleration. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 4340-4349.

[8] Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., & Shao, L. (2020). Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 1529-1538.

[9] Wu, J., Wu, H. N., Liu, A., Li, C. & Li, Q. S. (2019). A deep learning model compression method based on fusion of Lasso regression and SVD. Telecommunications Technology (05), 495-500.

[10] Gong, K., Zhang, C., & Zeng, G. (2020). Convolutional neural network model pruning combined with tensor decomposition compression method. Comput. Appl, 40, 3146-3151.