Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/19/20231025

Research on pruning optimization techniques for neural networks

Jiajun Wang

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, 264200, China

202100800049@mail.sdu.edu.cn

Abstract. Large deep neural networks have been deploying in more and more application scenarios due to their success in multiple application scenarios. However, deep neural networks are difficult to apply to devices with fewer resources, as the large models and huge demand for computing resources make this difficult. Pruning optimization, as a critical model compression method, has become an essential part of the deployment process of deep neural networks and has extreme significance. This article summarizes the methods of deep neural network pruning optimization technology, sorts out the current research status of pruning optimization technology, analyzes different fine-grained pruning optimization technologies based on the different fine-grained levels of pruning optimization technology, and comparing the characteristics of different fine-grained pruning optimization techniques. This article also introduces the development process and current development direction of different fine-grained pruning optimization technologies, compares the effectiveness differences between different fine-grained pruning optimization technologies and looks forward to the combination of pruning optimization technology and model quantification technology. The end of the paper summarizes pruning optimization techniques and provides prospects.

Keywords: pruning optimization, neural network model, compression, deep learning.

1. Introduction

Due to its superior performance, CNN have recently undergone rapid development and are now widely employed in natural language processing, object Detection, face recognition，and other fields. The performance optimization of convolutional neural networks mainly improves the accuracy by increasing the number of parameters and continuously deepening the number of neural network layers with the help of a high-performance Image Processing Unit (GPU) to accelerate the operation. However, this also enormously increases the convolutional neural network's computational complexity and memory space requirements. As the quantity of layers in the deep learning network model increases, the number of parameters also increases, and the computational complexity of the model also increases with the number of parameters so that each calculation process requires millions of parameters for multiple iterations to complete, which has high requirements for the storage and computing resources of the computing platform. However, at the same time, the computing performance of hardware devices and the read and write performance of storage are growing relatively slowly, which makes it hard to ensure the real-time performance and power consumption limit when the model is deployed on devices with

DOI: 10.54254/2755-2721/19/20231025

low computing power and memory space such as embedded devices. Deep neural network models, according to studies, frequently feature a lot of duplicate parameters [1]. On the one hand, these parameters increase the amount of calculation and induce overfitting; on the other hand, they contribute little to the accuracy improvement and may even reduce the accuracy. To solve the contradiction between the expanding scale of the CNN model and the limited resources of deployed devices, the pruning optimization technology inspired by biological synaptic pruning emerged as The Times requires, as shown in Figure 1. By deleting unimportant neurons to reduce the amount of calculation and parameters of the model, the optimized network is obtained, and the lightweight network is realized while the accuracy is reduced slightly and the generalization ability is improved. It can help large networks to be deployed on resource-limited hardware.

With increasing attention to pruning optimization techniques, many results have emerged. However, a more relevant summary and comparative literature are required. Therefore, this paper combs these results and summarizes the research results of pruning optimization.

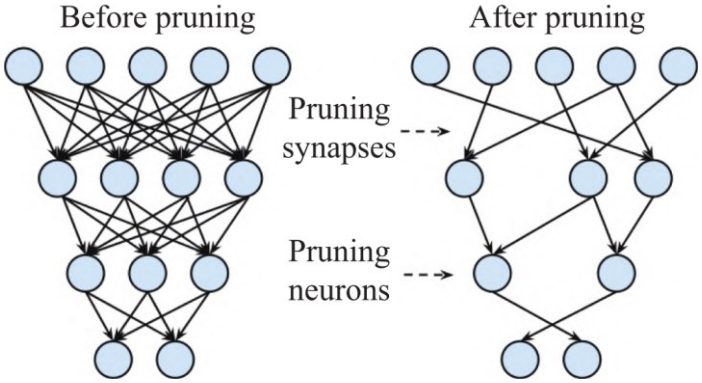

Figure 1. Neurons before and after pruning [2].

2. Pruning optimization algorithm

Pruning optimization makes the model lightweight by removing redundant parameters and making the parameter sparsity more significant. Pruning optimization methods can be roughly divided into four types according to the granularity of pruning: filter (channel) pruning kernel granularity pruning, kernel vector pruning, and fine-grained pruning. Next, this article will provide a detailed introduction to these four pruning optimization methods

2.1. Fine-grained pruning

As a type of unstructured pruning, fine-grained pruning directly deletes the parameters not always needed in the convolution kernel. Because the pruning grain size of this method is the smallest, the compression ratio of fine-grained pruning is relatively highest. In the early stage of fine-grained pruning, using the Taylor series approximate loss function to determine the contribution of parameters to the model accuracy through the quadratic term of the Taylor series, and the parameters with fewer contributions were removed to make the model lightweight [3]. However, this way of approximating the loss function by the Taylor series is computationally expensive and time-consuming. HanSong combines quantization with Huffman coding and fine-grained pruning [4]. The weights in the model are compressed significantly by quantization and Huffman coding, and the precision loss is reduced by retraining. The proposed compression method can achieve up to 35 times the compression ratio in many usual neural network models with little loss of model accuracy. However, the method proposed by HanSong is static pruning, and the parameters subtracted cannot be added again in the subsequent network, which may cause some crucial parameters to be missing in the following network, resulting in reduced accuracy. In response to reference [5], a dynamic network pruning method is proposed, which allows the resplicing previously subtracted parameters to restore meaningful connections while pruning (Figure 2).

DOI: 10.54254/2755-2721/19/20231025

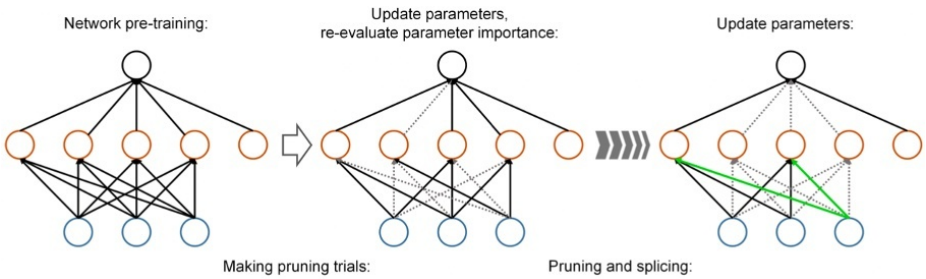

Figure 2. Dynamic pruning of redundant networks [5].

Because this method can dynamically judge the weights of parameters, compared with reference [4], this method has less error pruning and can further reduce the training time and get a larger compression ratio. Fine-grained pruning has a minor loss of model accuracy, but because it is unstructured pruning, the pruned model cannot be directly trained on GPU, and it usually requires additional hardware design to adapt.

2.2. Vector-level pruning

In-kernel vector granularity pruning belongs to unstructured pruning. The former is mainly aimed at pruning vectors in the convolution kernel. For intra-kernel vector pruning, Anwar proposed the intra-kernel strided sparsity algorithm, which pruned the vectors in the kernel by a certain step size [6]. The initial length was randomly assigned, but a larger step size could be selected to obtain the best pruning effect if the pruning was successful. This method prunes the vectors in the core into a pruning method with more regular connections between networks. Therefore, in sparse networks, since the same index can be used in the core, this method means that the pruning parameters need fewer indexes and therefore require less storage space than fine-grained pruning, as shown in Figure 3.

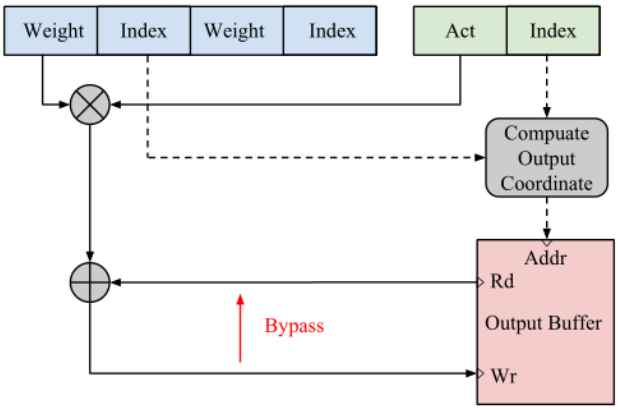

Figure 3. Sparse format storage Simplified storage [7].

In addition, Mao points out that under the same sparsity condition, the accuracy of intra-kernel vector pruning and fine-grained pruning in various standard models (VGG-16, Google Net) is less than 1% and can reach more than 80%. Compared with fine-grained pruning, the number of memories reads and writes of in-core vector pruning is significantly lower, only 60%~70% of the latter [7]. Compared with fine-grained pruning, in-core vector pruning requires less memory space, has less accuracy degradation, and requires less hardware, and its ASIC acceleration is easier to develop.

2.3. Kernel granularity pruning

Kernel granularity pruning is a coarse-grained pruning method, which usually directly removes the convolution kernels that contribute little to the result to make the number of parameters less and multiply and add operations. However, Lebed proposed a method to build the original convolution kernel tensor through sparse sub-convolution kernels to accelerate the operation speed of matrix multiplication and

DOI: 10.54254/2755-2721/19/20231025

reduce the parameters [8]. This approach achieves an 8.5x speedup over partial convolutional layers of Alexnet with only a 1% drop in accuracy. In pruning optimization, it is crucial to judge the pruning target's contribution to the model's accuracy. The commonly used norm-based method to determine the contribution to the accuracy may mistakenly delete the convolution kernels that have a considerable contribution to the model accuracy when small and negligible differences are between norms of different convolution kernels, or the minimum norm is still significant. A convolution kernel pruning method based on the geometric median is pointed out, which filters out the redundant and repeated parts of the convolution kernel according to the geometric median to maintain accuracy better while removing unnecessary parameters from the model [9]. Considering that the local contribution of the same convolution kernel may not be the same as the overall contribution of the model, Jiang proposed a new pruning idea: only the kernels in the filter are pruned at first, and the kernels with the most vital feature extraction ability are retained, and then the kernels in the convolution layer are pruned after reaching a certain compression ratio [2]. This method can further enhance the compression ratio of kernel granularity pruning. In general, the key to kernel granularity pruning is identifying the convolution kernel with the most limited contribution to the model accuracy. In general, kernel granular convolution is a type of structural pruning that makes it possible to use GPU acceleration while maintaining high compression rates directly. This more friendly way of reading and computing data allows granular kernel pruning to be deployed on more devices with limited performance.

2.4. Filter (channel) level pruning

Channel pruning removes the redundant channel directly and thus removes all operations on the convolution kernel associated with the channel. This effectively reduces model size and is one of the most popular pruning methods. A channel pruning method based on channel contribution is proposed, simplifying the calculation of all convolution kernels connected with the channel by deleting channels with a low contribution [10]. However, the paper [11] simplified the grid by extracting features from the original channels and reconstructing new channels. The most representative channels were retained through repeated iterations of LASSO regression, and the least square method was used to rebuild the remaining grids. In this way, the error of grid reconstruction is reduced, and the redundancy of shallow networks is significantly reduced. This method achieves five times acceleration on the VGG16 network without any loss of accuracy. However, literature [12] points out that literature [11] attaches too much importance to local areas while ignoring the whole grid, which may increase the error propagation in the grid. Therefore, a method of joint pruning of the entire neural network based on the same objective is proposed (Figure 4). To preserve the predictive power of the clipped network to the maximum degree, this method uses the NISP algorithm to determine which interneurons to delete by importance scores.

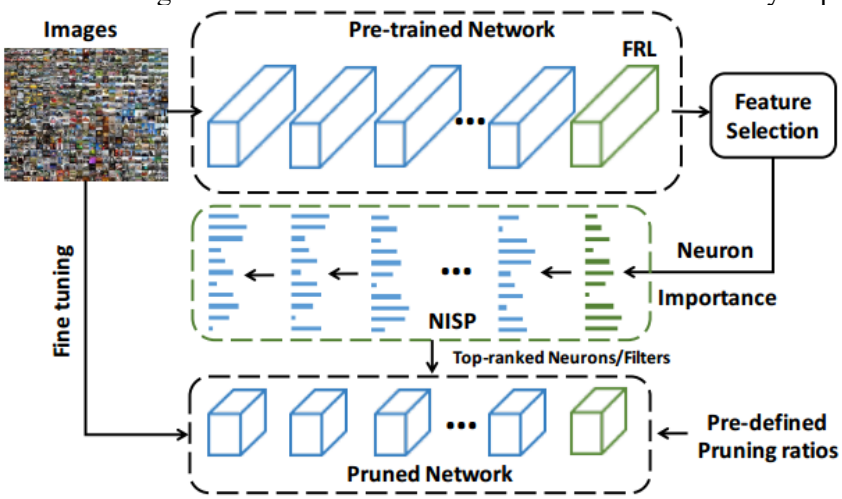

Figure 4. Schematic diagram of NISP algorithm [12].

DOI: 10.54254/2755-2721/19/20231025

In reference [13], a network slimming method is proposed for the pruning of large networks, which utilizes the objective function：

$$L = \sum_{(x,y)} l(f(x,\omega),y) + \lambda \sum_{\gamma\in\Gamma} g(\gamma) \quad (1)$$

The former term is used to represent the loss of regular training in CNN, as shown in Figure 5, while the latter term is the penalty term. γ is how important the channel is; when it's small, it will be deleted.

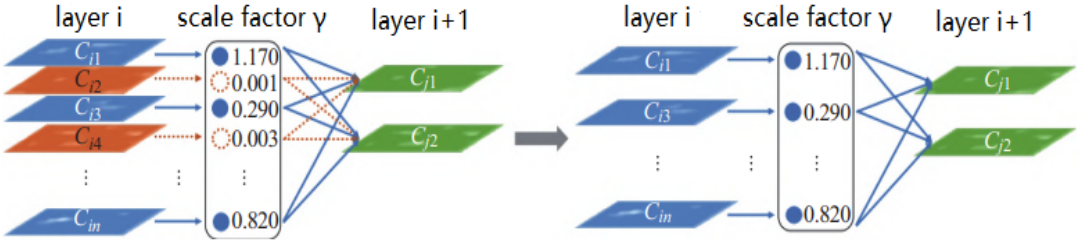

Figure 5. Before (left) and after (right) network slimming [2].

The method proposed in the literature [13] can automatically identify and trim unimportant channels, significantly reducing the amount of computation and without significant loss of accuracy. For the defects of the document [13], which requires a large quantity of computation to retrain the network, is difficult to converge, and retains redundant channels when minimizing reconstruction errors, the document proposes a channel pruning method with identification power perception [14]. The channel with the most substantial extraction capability is retained in each layer by introducing additional channels to assist the feature extraction capability of loss identification channels. The greedy algorithm was used for channel pruning and parameter optimization. In general, channel pruning involves pruning large models with many redundant parameters to minimize the number of parameters while maintaining accuracy. However, literature [15] points out that, in the case of the target network structure being determined, direct training of a small model may achieve higher precision than trimming and fine-tuning a large network with the same amount of computation. In general, channel pruning, as a coarse-grained method, can significantly compress model parameters. However, when the compression rate is high, the model's accuracy will suddenly drop sharply after compression, and it is not easy to recover the accuracy through fine-tuning. However, as a structured pruning, channel pruning is more hardware-friendly, which enables the network to train directly using GPUs.

3. Compare

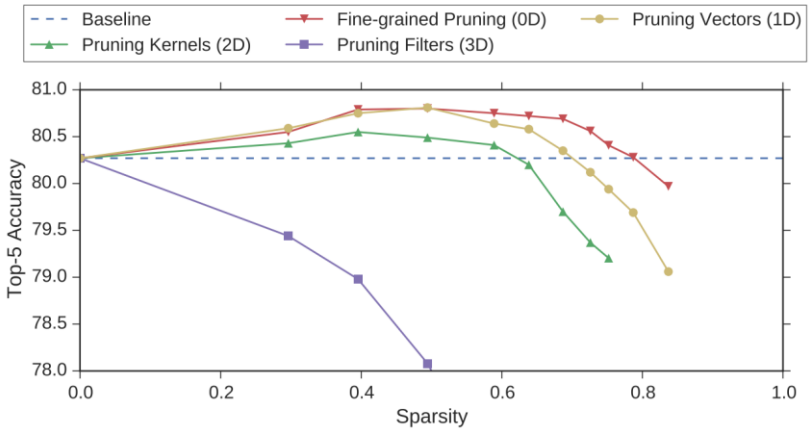

Figure 6. Accuracy-Sparsity Curve of AlexNet obtained by iterative pruning. [7].

DOI: 10.54254/2755-2721/19/20231025

The accuracy of filter pruning significantly decreases with the increase of compression rate, while other fine-grained pruning methods maintain better accuracy as the compression rate gradually increases. When the compression rate increases, the accuracy of fine-grained pruning methods first increases and then decreases, and the overall accuracy is relatively smooth. Because sparsity reduces the number of parameters while also limiting their positions, accuracy increases instead of decreasing when the compression rate is low. And the finer the pruning granularity, the higher the model accuracy under the same compression ratio.

4. Conclusion

This scholarly article focuses on the optimization of deep neural network models through the process of pruning. Specifically, this work explores various techniques ranging from fine-grained pruning to coarse filter (channel) pruning, including kernel vector pruning and kernel granularity pruning. According to different pruning granularity, the compression rate of pruning optimization usually decreases with the increase of pruning granularity at the same accuracy. For example, in channel pruning, when the compression ratio exceeds a certain value, its accuracy will significantly decrease. However, for structural pruning with coarse pruning granularity, since this method does not directly modify specific parameters, it is unnecessary to consider how to accelerate the calculation of sparse matrix, and most hardware training can be used directly. For the problem of static pruning not being able to recover only temporarily useless parameters, this article introduces the method of. To achieve the goal of improving the overall accuracy of the network, dynamic pruning recalculates the contribution of parameters and restores the parts that contribute more to the current network.

The further development of deep neural networks will promote pruning optimization techniques to be further optimized for the compression of large deep networks and further applied in conjunction with other model compression techniques. For example, how to combine pruning optimization with parameter quantization to improve the compression rate of the model can be regarded as the content of subsequent research. The pruning compression technology of the model will help deploy the deep neural network model to edge scenarios, such as embedded devices and intelligent driving vehicles. Take the innovative vehicle as an example. As intelligent driving needs to deal with the changing environment in real-time, such as high-speed driving and complex road conditions in urban areas, intelligent vehicles put extremely high requirements on the computing speed of the model to ensure safety. In this scenario, the pruning optimization technology compresses the neural network model, compressing the model parameters, improving the model calculation speed, and reducing the model power consumption. Pruning optimization techniques enable deploying of deep CNN models on devices with low computing power and memory space.

References

[1] Denil M,Shakibi B, Dinh L, et al. Predicting Parameters in Deep Learning. Inter. Conf. NIPS - Volume 2.USA: Curran Associates Inc., 2013: 2148-2156.

[2] Jiang X., Li Z., Huang L., Peng M., Xu S. Review of Neural Network Pruning Techniques.2022, J. App. Sci., **40(5)**: 838-849.

[3] Le Cun Y, Denker J S, Solla S A. Optimal Brain Damage. 1989, *Inter. Conf. Nips*: 598-605.

[4] Han S, Mao H, Dally W J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, 2015. arXiv:1510.00149.

[5] Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient dnns. 2016, arXiv preprint arXiv:1608.04493.

[6] Anwar S, Hwang K, Sung W. Structured pruning of deep convolutional neural networks. 2017, *J. Emerg. Tech. Comput. Sys.* **13(3)**: 1-18.

[7] Mao H, Han S, Pool J, Li W, Liu X. Exploring the regularity of sparse structure in convolutional neural networks. arXiv preprint arXiv:1705.08922, 2017

[8] Lebedev V, LempiTsky V. Fast ConvNets using group-wise brain damage, 2016 *IEEE Conf. Comp. Vis. Patt. Rec.*, 2016: 2554-2564.

DOI: 10.54254/2755-2721/19/20231025

[9] He Y, Liu P, Wang Z W, et al. Filter pruning via geometric Median for deep convolutional neural networks acceleration, 2019 *IEEE Conf. Comp. Vis. Patt. Rec.*: 4335-4344.
[10] Polyak A, Wolf L. Channel-level acceleration of deep face representations. 2015, *IEEE Access*, 3: 2163-2175.
[11] He Y H, Zhang X Y, Sun J. Channel pruning for accelerating very deep neural networks, 2017 *Inter. Conf. Compu.Vis*, 2017: 1398-1406.
[12] Yu R C, Li A, Chen C F, et al. NISP: pruning networks using neuron importance score propagation 2018 *IEEE Conf. Comp. Vis. Patt. Rec.* 9194-9203.
[13] Liu Z, Li J G, ShEn Z Q, et al. Learning efficient convolutional networks through network slimming, 2017, *Inter. Conf. Compu.Vis*, 2755-2763.
[14] Zhuang Z W, Tan M K, Zhuang B H, et al. Discrimination-aware channel pruning for deep neural networks. 2018 https://arxiv.org/abs/1810.11809.
[15] Liu Z, Sun M J, Zhou T H, et al. Rethinking the value of network pruning. 2019 https://arxiv.org/abs/1810.05270.