The Advancements of Pruning Techniques in Deep Neural Networks

Xinrui Li
Software Engineering, Beijing-Dublin International College, Beijing, China

Keywords: Neural Network, Structured Pruning, Unstructured Pruning.

Abstract: Neural network pruning is widely used to compress large, over-parameterized Deep Neural Network (DNN) models. Previous studies have shown that pruning optimizes DNN models in the following aspects: saving resource consumption and storage space costs, improving model deployment range, improving inference speed, and achieving real-time service. This article summarizes the development and main progress of pruning technology up to now and discusses the two types of pruning, basic principles, advantages and disadvantages, and the performance in different application scenarios. The research prospect of combining automation technology with pruning technology is also prospected. Pruning can be divided into two categories including unstructured pruning and structural pruning. Unstructured pruning provides the flexibility to adjust parameters to task requirements while achieving a greater proportion of compression, while structural pruning improves model performance, stability, and interpretability by removing the entire module. Their respective characteristics make them suitable for models with different structural complexity and different task requirements. Although pruning is relatively well developed compared to other model compression techniques, there are still some challenges. The current pruning technology is still dependent on manual recognition of pruning parameters, and the technology of automatic recognition of pruning objects has not yet been developed. This paper summarizes the research status of pruning technology in the field of DNN and discusses current technological developments, applications, limitations, and challenges in development. The review concludes by highlighting some potential for future research in this area, including exploring automated pruning and the ability to enhance and transfer learning between fields.

1 INTRODUCTION

In contrast to traditional Artificial Neural Networks (ANN), Deep Neural Networks (DNN) can automatically learn features from large amounts of data, enabling numerous Artificial Intelligence (AI) technologies e.g. image and speech recognition (Afouras et al, 2022) , natural language processing, object detection and autonomous driving (Basiri et al, 2021, Qutub et al, Jinhan et al) . The wide range of applications of DNNs has led to their vigorous development in academia and industry. Despite rapid development, the enormous scale and computational requirements of these DNN models present significant challenges for actual deployment, especially in facilities with limited hardware resources, such as smart sensors, and wearables (Denil et al, 2013) . As these challenges become more relevant, the field of model compression has emerged as a key area of research to mitigate these limitations. Generally speaking, model compression technology could help improve calculation speed and model storage friendliness by reducing model parameters and scale, and maintaining or surpassing previous model performance.

In recent years, more and more research has been done on model compression methods, and great progress has been made from theoretical research to platform implementation. The earliest model compressions can be traced back to 1989 when Cun et al. proposed for the first time in OBD (Optimal Brain Damage) to achieve the effect of compression size by eliminating unimportant parameters in the network (Cun et al, 1989) . At present, most cropping schemes are based on Cun's OBD method. The paper "Deep Compression" published by Han et al. in 2015 triggered a wave of research on pruning, which reviewed the application of cutting, weight sharing, quantization, coding and other technologies (Han et al, 2015) . At present, DNN model compression is mainly divided into three directions. First, optimize the model design and build a compact and efficient model, for example, SqueezeNet and MobileNet

Li, X.
The Advancements of Pruning Techniques in Deep Neural Networks.
DOI: 10.5220/0012837000004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 177-181
ISBN: 978-989-758-690-3

(Iandola et al, 2017, Howard et al) . The second is to cut the model by cutting the redundant parameters and trying to maintain the performance. Among them, the method of evaluating weights and the pruning method have received wide attention. The third is kernel sparsity, which makes the model sparser by inducing weight updates and applying compact storage methods such as CSC coding (Barra et al, 2020, Gu et al, 2015) . Although sparsity operates faster on hardware platforms, it is affected by bandwidth. In addition, there is also much research on neural architecture search (Fedorov et al, 2019) , policy refinement (Wu et al, 2022) , and other methods, which have played a good role in model compression.

In the field of deep neural networks, the mainstream model compression techniques can be roughly divided into the following four categories: pruning, knowledge distillation, quantization, and low-rank decomposition. Among them, pruning is widely used to reduce the most common model compression problem - over-parameterization, which mainly occurs during the model training phase. Because some parameters assist the training of the model and are not often used after the training, it is necessary to trim these redundant parameters. Thus, the pruning algorithm is proposed and its core idea is to reduce the number of parameters and computations in the network model while ensuring the performance of the model is not affected as much as possible.

The excellent performance of pruning technology is reflected in that it cannot only improve model sparsity and inference speed, but also reduce overfitting risk and resource demand, and is environmentally friendly. In addition, in different tasks and scenarios, the research of pruning technology can also provide more possibilities for customization and optimization. Therefore, the in-depth exploration of pruning is expected to drive continuous innovation and development in the field of model compression.

The rest of this paper is organized as follows. In the method section, this paper will mainly list the application of the pruning method in various fields, as well as the core changes and innovations of the pruning method in recent years. In the discussion section, possible challenges and future directions for the pruning method will be further discussed. Finally. this review will predict the future potential development of model compression techniques in the conclusion part.

2 METHOD

The broad definition of pruning includes three aspects, regulation, pruning and growth. The development of these techniques provides a comprehensive optimization scheme for the efficiency and performance of deep learning models in practical applications. The basic idea and principle of pruning is to streamline the model structure by removing unnecessary neurons and connections, promoting model storage friendliness, memory efficiency, and computational efficiency. The steps of pruning algorithm include model training, pruning, fine-tuning and model retraining.

Pruning algorithms can be broadly divided into two branches according to the target level to be cut: unstructured pruning and structured pruning shown in Figure 1. The main difference between them lies in the pruning target and the structure after pruning. Unstructured pruning refers to pruning individual weight and would result in sparse weight matrix due to the direct weight-pruning. This results in the need for dedicated hardware or libraries to achieve compression and acceleration. Structured pruning· refers to prune in the level of filter or channel or layer, which preserves the original convolutional structure and can directly run on a mature deep learning framework without requiring the support of a specific algorithm library or hardware platform.

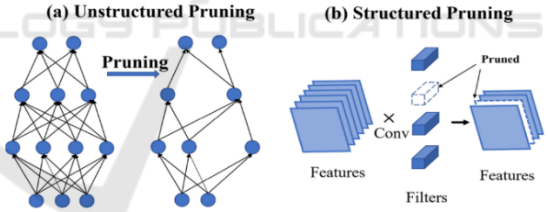

Figure 1. The schematic illustration of pruning (Chen et al, 2021) .

2.1 Unstructured Pruning

In unstructured pruning, unimportant elements in the model weight matrix are zeroed to create a sparse matrix, which is then stored in a sparse way to reduce the storage capacity of the model to achieve compression. But this element-removing would ignore the internal structure and lead to irregular sparse structure after pruning (Anwar et al, 2017, Fang et al) . Among unstructured pruning models, the most representative one is Han's Deep Compression paper which uses unstructured loss, weight reunion class, and Huffman coding to achieve the ultimate compression effect of the model (Han et al, 2015) .

Because unstructured pruning can lead to random and sparse structures, it will destroy the model structure greatly and may affect the model performance, so the application scope is relatively narrow.

2.1.1 Graphic Processing

In Pietron et al. ‘s model, unstructured pruning improved the efficiency of graphics processing units (Pietro & Żurek, 2022) . Previous studies on unstructured pruning focused only on the memory footprint of compressed models, which resulted in a relatively low reduction in the computational effort. More recent studies have looked more at how to balance the compression model and reduce the computational effort. Xu et al. proposed a method to explore the entire pruning structure space effectively and found a balanced pruning scheme between model size and pruning workload (Xu, 2021) .

2.2 Structured Pruning

Current structured pruning models mainly focus on how to automatically identify redundant parameters and prune them, or how to prune them away more quickly. Ning et al.’s AutoCompress model efficiently incorporates the combination of structured pruning schemes into an automated process (Ning et al, 2020) . Mitsuno et al. proposed a sparse regularization algorithm to improve the ability of automatically identify unnecessary weights (Mitsuno et al, 2020) . To prune parameters more quickly, Li et al. proposed a single-pass automated pruning framework that enables faster convergence and fewer hyperparameters (Li et al) .

Structured pruning is not only used to prune computationally heavy DNN models in various fields but also to help restore robustness to attacked models. For instance, Guan et al. developed a model called ShapPruning to recognize and cut off infected neurons to enhance robustness (Guan et al, 2022) .

2.2.1 Graphic Processing

Zhang et al. have developed structural pruning methods that can offset the limitations of irregular network structures and accelerate at high resolution (Zhang et al, 2022) . The MPDCompress algorithm proposed by Supic et al. can improve the model inference speed on various hardware platforms and achieve more than 4 times the operational efficiency (Supic et al, 2018).

2.2.2 Medical

Current DNN models have shown better recognition accuracy than human experts in some fields, such as tumor recognition (Qiu et al, 2022) , so how to compress and simplify the models and put them into use with the increase in scale and computational complexity has become an urgent task.

More research has focused on how to solve this problem in recent years. Li et al. combined hardware-friendly, structured model compression with moving target encoder optimization to achieve DNN model inference on mobile devices (Li et al, 2021) . Li et al. also overcame the difficulty of limited medical instrument resources and successfully compressed the original model by 36 times using structural weight pruning (Li et al, 2019) . Hedegaard et al. proposed the concept of Continuous Inference Networks (CINs) and used fusible adaptation networks and structured pruning to achieve exceptional prediction accuracy over finetuning plus pruning (Hedegaard) .

3 DISCUSSION

Under limited resources, pruning improves the efficiency of DNN but affects the interpretability. Structural pruning improves explainability by removing entire modules, while non-structural pruning makes the model structure difficult to interpret. Although structural pruning is superior in interpretability, the choice of method should take into account the task and application scenario. Structural pruning is better suited for tasks that emphasize overall model performance, inference speed, and model stability, especially in resource-constrained environments. Its hardware optimization advantages make it suitable for embedded devices and edge computing platforms, such as in the graphics processing field and the medical field (Zhang et al, 2022, Supic et al, 2018, Li et al, 2021, Li, 2019, Hedegaard). Unstructured pruning also has unique advantages in some tasks. Its flexibility and adjustability enable the model to meet the requirements of different application scenarios, such as image classification requirements for graphics processing (Pietro & Żurek,2022, Xu et al, 2021). In summary, the selection of pruning strategies needs to balance model interpretation, performance, hardware optimization, task requirements, and interpretability to ensure the best results in practical applications.

The biggest limitation of pruning methods is that there is no universal standard for building parameters that applies to all models and tasks. Each model still

has a strong artificial dependency, relying on the manual selection of pruning parameters, and the cost of model compression increases greatly as the model gets larger.

Recent studies mainly focus on automatic pruning methods and solving the impact of unstructured pruning on interpretability, such as using sparse regularization algorithm to automatically identify pruning parameters and reduce the dependence on manual parameter selection (Mitsuno et al, 2020) . The automated pruning field has uncultivated potential. Most of the existing pruning methods are only for specific models and tasks, which require strong domain knowledge. Thus, it usually requires AI developers to spend a lot of energy to apply these methods to their own scenarios, which is very costly. The ideal structural pruning algorithm should satisfy the following conditions: the model can be trained automatically from scratch, no further fine-tuning is required after pruning, and can achieve high performance and lightweight. However, the complexity of neural networks makes achieving the goal of automated pruning extremely challenging. To achieve this end, it is necessary to systematically address three core questions: identifying the parts of the network that should be pruned, determining how to prune without damaging the network, and finding a way to automate these two processes. Successfully answering these questions could be a significant stepping stone towards automated pruning.

4 CONCLUSION

This review comprehensively explores neural network pruning techniques used to compress large Deep Neural Network models in areas such as image classification and medical treatment. It discusses two main types of pruning: unstructured and structured pruning. Unstructured pruning, known for its flexibility, can significantly compress models but may reduce performance and interpretability. In contrast, structured pruning maintains performance and stability, making it suitable for embedded devices and edge equipment. Each type's characteristics make them suitable for different situations, and all factors should be considered when choosing a pruning strategy. The research on pruning impacts the field of DNN by saving resource consumption and storage space costs, improving the model deployment range, enhancing inference speed, and enabling real-time services. However, the application of pruning faces some challenges, including decreased interpretability and domain difference problems. Due to the varying distribution and characteristics of model data across different fields, pruning methods trained based on the weight of data in different domains lack transferability. Future research on pruning could focus on enhancing model interpretability, generality, and automation in pruning processes. Combining pruning technology with model compression and optimization techniques, such as knowledge distillation and quantization, is expected to further improve model efficiency and performance.

REFERENCES

T. Afouras, et al., IEEE Trans. Pattern Anal. Mach. Intell. 44(12), 8717-8727 (2022)

M.E. Basiri, et al., Future Gener. Comput. Syst. 115, 279-294 (2021)

S.S. Qutub, et al., BEA: Revisiting anchor-based object detection DNN using Budding Ensemble Architecture, CoRR abs/2309.08036 https://arxiv.org/pdf/2309.08036v4.pdf

K. Jinhan, et al., Reducing DNN Labelling Cost using Surprise Adequacy: An Industrial Case Study for Autonomous Driving, in Proceedings of ACM SIGSOFT Conf. Found. Softw. Eng. abs/2006.00894, 1466-1476 (2020)

M. Denil, et al., Adv. Neural Inf. Process. Syst., 2148–2156 (2013)

Y. Le Cun, et al., Conf. Neural Inf. Process. Syst., 598-605 (1989)

S. Han, et al., Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding, in Proceedings of Int. Conf. Learn. Represent. abs/1510.00149 (2015)

F.N. Iandola, et al., SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, arXiv: Comput. Vis. Pattern Recognit. (2017)

A.G. Howard, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861, https://arxiv.org/pdf/1704.04861.pdf

S. Barra, et al., IEEE/CAA J. Autom. Sinica 7(3), 683-692 (2020)

S. Gu, et al., Convolutional Sparse Coding for Image Super-Resolution, in Proceedings of IEEE Int. Conf. Comput. Vis., 1823-1831 (2015)

I. Fedorov, et al., Adv. Neural Inf. Process. Syst. 32, 4978-4990 (2019)

Y. Wu, et al., IEEE Trans. Neural Netw. Learn. Syst. 33(9), 5057-5069 (2022)

S. Anwar, et al., ACM J. Emerg. Technol. Comput. Syst. 13(3):32:1–32:18 (2017)

G. Fang, et al., Depgraph: Towards any structural pruning, CoRR abs/2301.12900, https://arxiv.org/pdf/2301.12900.pdf

M. Pietroń, D. Żurek, J. Comput. Sci. 67 (2022)

K. Xu, et al., Neurocomputing 451, 81-94 (2021)

L. Ning, et al., Autocompress: An Automatic Dnn Structured Pruning Framework For Ultra-High Compression Rates, in Proceedings of AAAI Conf. Artif. Intell. 34, 4876-4883 (2020)
K. Mitsuno, et al., Hierarchical Group Sparse Regularization for Deep Convolutional Neural Networks, in Proceedings of IEEE Int. Joint Conf. Neural Netw. abs/2004.04394 (2020)
Z. Li, et al., SS-Auto: A Single-Shot, Automatic Structured Weight Pruning Framework of DNNs with Ultra-High Efficiency, arXiv preprint arXiv:2001.08839,https://arxiv.org/pdf/2001.08839.pdf
J. Guan, et al., Comput. Vis. Pattern Recognit. 13348-13357 (2022)
L. Chen, Y. Chen, J. Xi, X. Le, Complex Intell. Syst. 1-0 (2021)
T. Zhang, et al., IEEE Trans. Neural Netw. Learn. Syst. 33(5), 2259-2273 (2022)
L. Supic, et al., Comput. Res. Repos. abs/1805.12085 (2018)
Y. Qiu, J. Wang, Z. Jin, H. Chen, M. Zhang, L. Guo, Biomed. Signal Process. Control 72, 103323 (2022).
H. Li, et al., Real-Time Mobile Acceleration of DNNs: From Computer Vision to Medical Applications, in Proceedings of Asia South Pac. Des. Autom. Conf. 581-586 (2021)
H. Li, et al., Med. Image Comput. Comput. Assist. Interv. 89-97 (2019)
L. Hedegaard, Efficient Online Processing with Deep Neural Networks, CoRR abs/2306.13474, https://arxiv.org/pdf/2306.13474.pdf