Chinese Industry and Technology of Eco-environment
2025, VOL. 6, NO. 3, 46-54
DOI: 10.47297/wspciateWSP2633-772X09.20250603

Research and Application of Efficient Pruning Algorithms in Deep Learning Model Compression

Yibo Shen

[1]School of Computer Science, Qingdao City University, Qingdao, Shandong, 266106, China

ABSTRACT
With the continuous expansion of deep learning model scales, model compression techniques have become crucial for deploying models on resourceconstrained devices. This study systematically explores four mainstream neural network pruning strategies: finegrained weight pruning, coarsegrained weight pruning, channel pruning, and layer pruning, and proposes a comprehensive progressive pruning framework. We implemented and evaluated these strategies on the ResNet18 model using the CIFAR10 dataset, focusing on the tradeoffs among pruning rate, accuracy loss, and inference performance. Experimental results demonstrate that channel pruning achieves the best inference speedup while maintaining model accuracy, with an average throughput increase of 45%. In contrast, finegrained weight pruning excels in model size compression, achieving up to an 85% parameter reduction within a 5% accuracy loss margin. Additionally, our proposed adaptive layer pruning strategy based on activation value importance exhibits unique advantages in model structure simplification, automatically identifying and removing redundant layers while preserving key feature extraction capabilities. This study not only provides a comprehensive comparison of various pruning techniques but also develops a scalable toolchain supporting automated decisionmaking and finetuning during model compression, offering practical solutions for efficient deployment of deep learning models on edge devices.

KEYWORDS
Deep learning; Finegrained pruning; Coarsegrained pruning; Channel pruning; Layer pruning

1 Introduction

In the field of neural network compression and acceleration, with the widespread application of Convolutional Neural Networks (CNNs) in tasks such as image classification and object detection, the conflict between model complexity and computational resources is becoming more acute. Traditional pruning methods, while alleviating this problem to some extent, also reveal numerous limitations: sparse models obtained after traditional weight pruning often face compatibility issues, making it difficult to run efficiently on general-purpose platforms, requiring specific hardware or library support [1]; models with residual structures encounter difficulties in efficient filter pruning due to inter-layer coupling; there is a lack of a general pruning framework to systematically compare the advantages and disadvantages of pruning algorithms with different granularities [1]; traditional structured pruning methods have weak adaptability, cumbersome processing procedures, and high computational complexity [2]; many pruning methods rely excessively on manual design, resulting in significant decreases in network accuracy after automatic pruning [2]; pruning methods based on clustering algorithms may not achieve optimal substructures, affecting the overall accuracy of the pruned network [2]; existing channel pruning methods often neglect the relationship between channels, limiting the pruning effect, and the setting of pruning rates is difficult, requiring multiple iterations of the pruning process, increasing time consumption and computational costs, and fine-tuning the pruned model is also extremely time-consuming [3]; in traditional structured pruning processes, the resource consumption for pruning the network and fine-tuning the subnet is severe, taking a long time, and it is difficult to accurately evaluate the performance of the pruned subnet, requiring multiple iterations of fine-tuning [4]; lightweight module design mostly relies on expert prior knowledge, requiring repeated comparative experiments and lacking universality [4]; sparse matrices generated by unstructured pruning are discontinuously distributed in memory, which is not conducive to hardware optimization and acceleration [5]; network quantization can compress the model, but often comes with a loss of accuracy, and non-uniform quantization is costly to implement on hardware [5]; knowledge distillation requires training a large network to assist in the training of a small network, increasing the overall training complexity and resource consumption [5].

To tackle these challenges, this study proposes a model compression framework that integrates multi-granularity pruning strategies. The aim is to achieve an optimal balance between model performance and compression efficiency through systematic and engineered methods. This framework draws on the highly engineered traditional weight pruning framework constructed by coarse-grained pruning strategies, which involves systematic warm-up training, progressive pruning, and advanced fine-tuning strategies, effectively balancing pruning efficiency and model performance. Simultaneously, it incorporates the adaptive hybrid-grained pruning idea of the hybrid-grained weight pruning strategy, dynamically adjusting the pruning threshold based on channel importance and weight amplitude, achieving complementary advantages of structured and unstructured pruning. Furthermore, this paper also references the layer

importance evaluation and adaptive hierarchical pruning rate allocation strategy of the channel pruning strategy, achieving intelligent channel pruning, and is equipped with a dynamic model reconstruction and accuracy recovery mechanism, significantly enhancing the stability and flexibility of structured pruning. Specifically, for the challenging task of pruning models containing residual structures, this paper draws on the radical hierarchical pruning idea of the hierarchical pruning strategy, combining activation-based layer importance evaluation and complex model reconstruction logic to directly remove entire network layers, exploring a new direction for model compression. Although this strategy carries a higher risk, it provides possibilities for extreme compression scenarios and has been experimentally verified to be effective.

2 System architecture and pruning algorithm

2.1 Data processing module

In the data processing module, we have designed a multi-level pipeline architecture to optimize data flow management and computational resource utilization. The architecture features an efficient data loader supporting standard/custom dataset reading via a unified interface. It employs an asynchronous prefetching strategy to eliminate I/O bottlenecks and utilizes a sharding loading mechanism to effectively control memory usage. The data preprocessing stage integrates data augmentation techniques such as normalization and random cropping, and introduces a dynamic difficulty sampling mechanism to assist the model in adapting to structural changes. Additionally, GPU acceleration is supported to significantly enhance training efficiency. In terms of batch processing management, we employ an adaptive batch scheduling algorithm to optimize batch parameters and combine a class-balanced sampling strategy to effectively mitigate class bias that may arise during the pruning process, ensuring the stability and accuracy of model training.

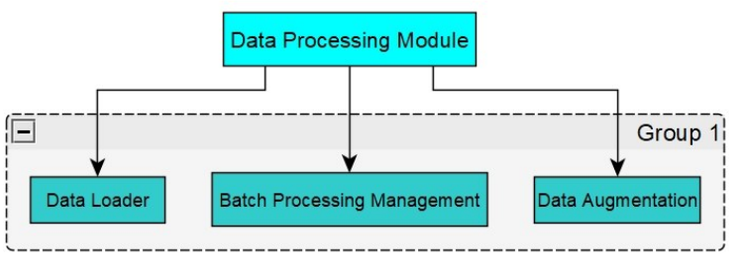

Figure 1 Data processing module [Owner-draw]

2.2 Model management module

We have constructed a unified lifecycle management framework to ensure the consistency and traceability of model states during the pruning process. The framework incorporates a model loader supporting multi-pretrained model import and auto-parsing, possesses cross-framework model conversion capabilities, and verifies model integrity to prevent parameter loss. In terms of model storage, an incremental checkpoint mechanism is adopted to reduce storage overhead, supports multi-version model management for rollback and comparison, and uses compressed representations and sparse weight-specific encoding to optimize storage. Additionally, the model conversion module supports multi-format lossless conversion to ensure cross-platform deployment, utilizes graph optimization algorithms to enhance inference efficiency, and possesses quantization-aware conversion functionality to achieve collaborative optimization.

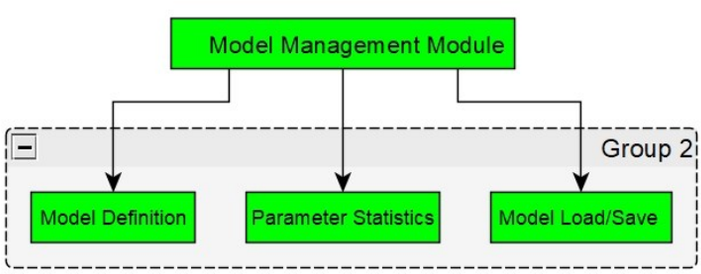

Figure 2 Model management module [Owner-draw]

2.3 Pruning strategy module

The system core integrates four complementary pruning algorithms to achieve fine-grained model compression control. The evaluation sub-module uses a multi-index fusion approach for comprehensive weight importance assessment, and is equipped with specialized evaluation algorithms to support different pruning granularity requirements. It also provides online/offline evaluation modes to adapt to dynamic adjustments. The pruning execution sub-module uniformly schedules four strategies: fine-grained weight pruning, coarse-grained weight pruning, channel pruning, and hierarchical pruning. A transaction mechanism guarantees operation atomicity and rollability, guaranteeing the stability of the pruning process. The structure reconstruction sub-module utilizes graph rewriting algorithms to maintain the integrity of network connections, automatically handles dimension matching issues, and adopts dedicated reconstruction strategies for special structures to ensure the usability and performance of the pruned model.

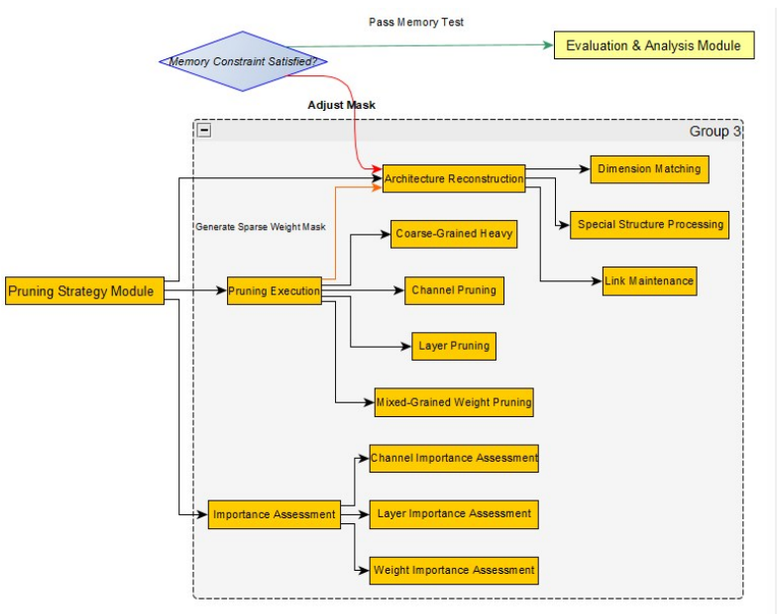

Figure 3 Pruning Strategy Module [Owner-draw]

2.4 Training/fine-tuning module

The optimization framework specifically designed for post-pruning networks balances efficiency and accuracy recovery through fine-grained management of the training process. This framework includes an optimizer management module that utilizes sparse-aware gradient updates to enhance computational efficiency. It combines a hierarchical adaptive learning rate mechanism with weight regularization control to maintain network sparsity while facilitating convergence. The learning rate scheduler introduces a pruning-aware warm-up strategy, automatically extending the warm-up phase to stabilize initial training. Coupled with segmented learning rate decay and an elastic restart mechanism, it effectively escapes local optima. The early stopping mechanism employs a multi-metric fusion judgment strategy, utilizes sliding window evaluation to reduce the impact of random fluctuations, and makes stopping decisions based on statistical significance to avoid overfitting. Additionally, a gradual learning rate increase strategy is adopted during the warm-up training phase to prevent unstable initial training. This strategy also collects feature statistical information to support subsequent importance evaluation and guides the selection of optimization strategies through weight sensitivity analysis.

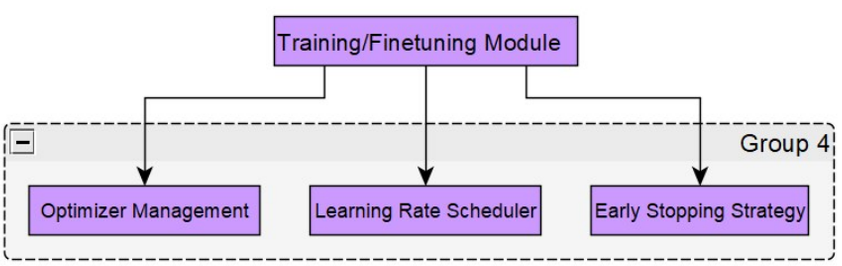

Figure 4 Training/finetuning module [Owner-draw]

2.5 Evaluation and analysis module

To ensure the effectiveness of the pruning process and comprehensive evaluation of model performance, this paper constructs a comprehensive performance monitoring and analysis tool. This tool integrates performance evaluation, model analysis, and visualization functions, supporting decision optimization and result verification. The performance evaluation module automatically measures multi-dimensional indicators (including accuracy, loss function, inference speed, resource utilization, etc.), efficiently processes large-scale test data in batch processing mode, and provides detailed performance reports. The model analysis module deeply analyzes the model structure, displays parameter distribution and computation proportion through hierarchical analysis, quantifies the compression ratio to evaluate the reduction proportion of volume and computation, and provides a basis for optimization strategies through bottleneck detection and sensitivity analysis. In addition, the visualization tool intuitively presents the indicator change curves during the training process, reflects the statistical characteristics of parameter distribution, and provides intuitive and powerful support for model optimization through comparison before and after pruning and feature map visualization.

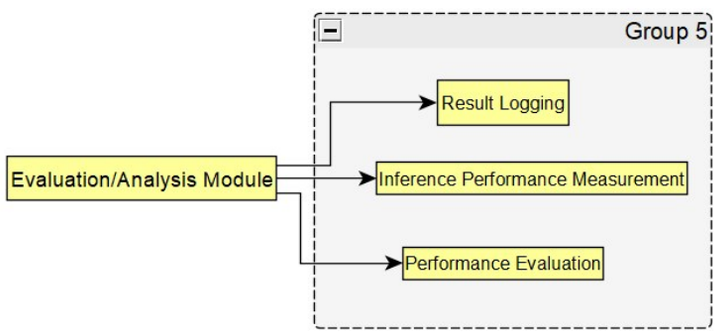

Figure 5 Evaluation/analysis module [Owner-draw]

2.6 Toolchain module

This article provides end-to-end support to simplify the application process of pruned models, covering three major modules: deployment tools, performance testing, and comparative analysis. The deployment tools support multi-platform deployment (including mobile devices, edge devices, and servers) and are optimized for hardware acceleration to ensure efficient model operation. Performance testing verifies the pruning effect through a multi-device testing platform, conducts load testing by simulating actual application scenarios, and analyzes power consumption and accuracy stability to provide a reliable basis for deployment. The comparative analysis module provides a multi-strategy comparison tool to showcase the performance of different methods, conducts a trade-off analysis between parameter sensitivity and compression accuracy, and quantitatively evaluates the computational efficiency acceleration effect to assist in selecting the optimal pruning strategy.

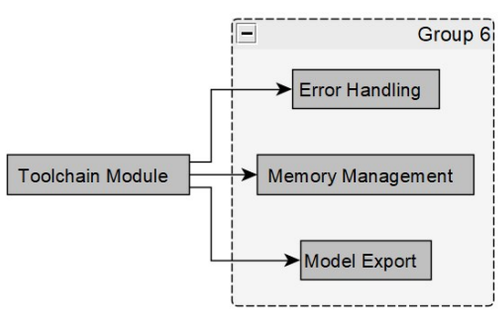

Figure 6 Toolchain module [Owner-draw]

2.7 Control flow module

The coordinated system workflow achieves automated pruning strategy execution and optimization through the collaborative efforts of a progressive pruning controller and a pruning strategy selector. The progressive pruning controller plans compression steps through a pruning scheduling algorithm, dynamically controls pruning intensity using an S-shaped growth curve, and combines an adaptive step size mechanism and rollback checkpoint management to ensure the stability and recoverability of the pruning process. The pruning strategy selector evaluates suitable pruning

granularity by analyzing the network structure, estimates the balance between compression and accuracy using a performance prediction model, supports differentiated application scenarios with the help of a hybrid strategy generator, and ultimately selects the optimal pruning scheme through a priority scheduling algorithm, achieving efficient and automated model compression optimization.

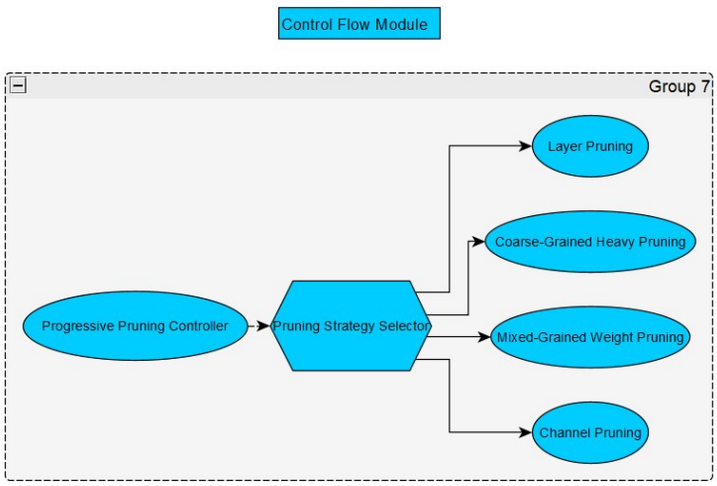

Figure 7 Control Flow Module [Owner-draw]

Channel pruning is the most representative and widely applied strategy here, balancing model compression rate and actual acceleration effect. The following is the process of channel pruning strategy:

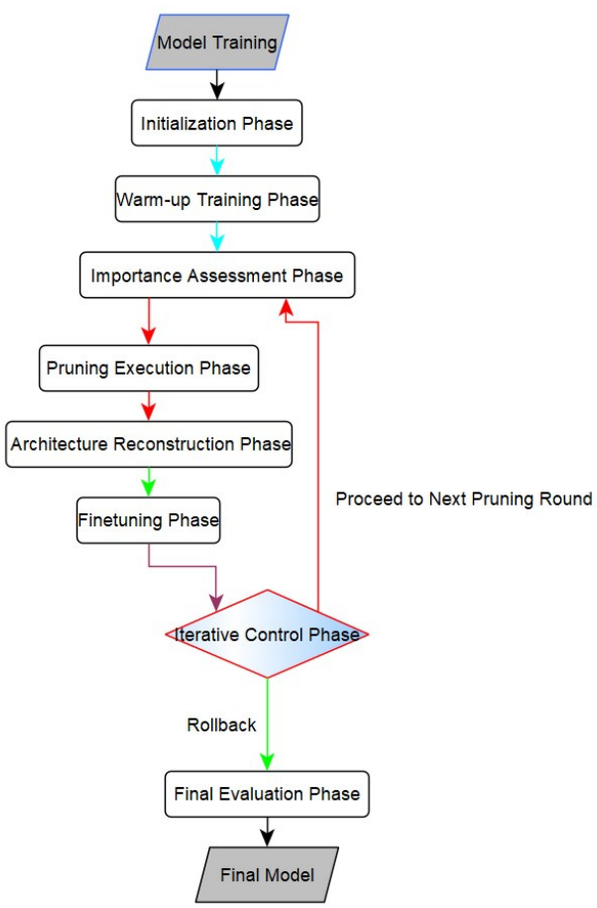

Figure 8 Channel pruning strategy workflow [Owner-draw]

3 Pruning algorithm and process

In the research on neural network compression and acceleration, traditional pruning algorithms often focus on a single pruning criterion (such as weight size), adopting a one-time "hard pruning" strategy supplemented by simple fine-tuning. While straightforward, this method often reduces model accuracy significantly, and the optimization process is relatively isolated. In contrast, the four innovative pruning algorithms proposed in this paper (coarse-grained pruning strategy, adaptive strategy for fine-grained weight pruning, channel pruning strategy, and hierarchical pruning strategy) elevate the pruning process into a systematic, adaptive, and highly engineered model optimization framework. The common highlights of these algorithms are as follows: Firstly, they adopt a gradual and iterative pruning strategy, abandoning the "brute force" approach of pruning a large number of weights at once, and maximizing model accuracy through multiple, small-step pruning and fine-tuning iterations. Secondly, they integrate advanced training and recovery strategies such as cosine annealing learning rate, Mixup/CutMix data augmentation, and label smoothing, significantly improving the convergence effect of the pruned model. Furthermore, a comprehensive automated evaluation system is constructed, covering multiple dimensional indicators such as model size, parameter quantity, inference delay, FLOPs, and accuracy, making the measurement of pruning effects more scientific and comprehensive. Lastly, an importance evaluation mechanism based on dynamic complexity indicators such as activation values, gradients, and hierarchical depth is introduced, making pruning decisions more intelligent. Specifically, the coarse-grained pruning strategy achieves optimization of structured weight pruning through a highly engineered pruning framework and systematic gradual pruning. The adaptive strategy for fine-grained weight pruning adopts an adaptive threshold and mixed granularity pruning strategy, enhancing the flexibility and efficiency of fine-grained weight pruning. The channel pruning strategy innovates the channel pruning method through adaptive pruning rate allocation based on layer importance and dynamic model reconstruction. The hierarchical pruning strategy proposes a layer importance evaluation based on activations and an aggressive hierarchical pruning idea, combining complex model reconstruction logic and advanced data augmentation techniques to achieve intelligent and efficient layer pruning. In summary, these four pruning algorithms jointly construct a closed-loop, adaptive, and systematic model optimization framework, significantly improving the efficiency and accuracy of model compression, achieving a leap from "technology" to "engineering system", and representing an important innovation in the field of neural network compression.

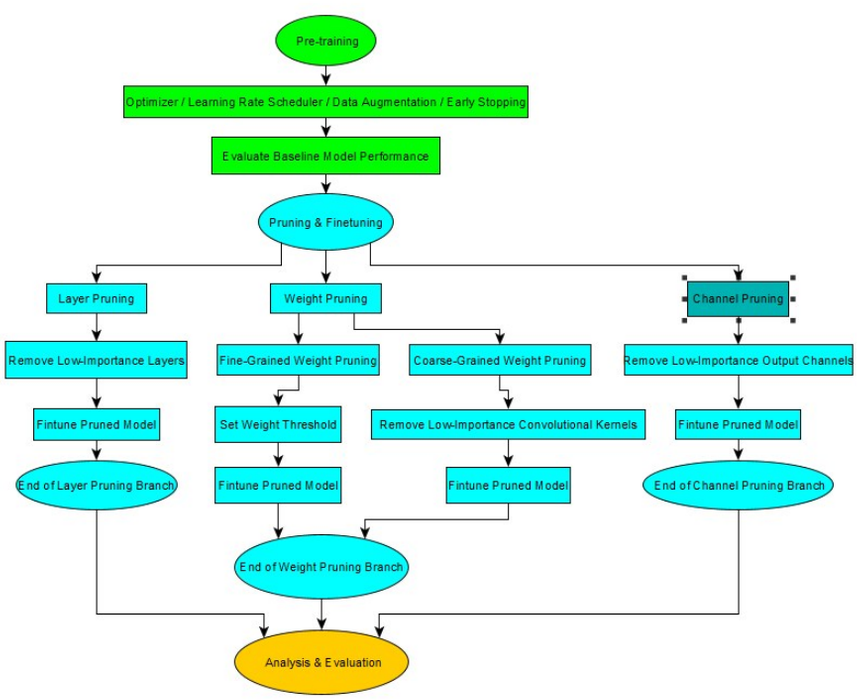

Figure 9 Overall implementation workflow [Owner-draw]

4 Experimental results and analysis

The experimental results show that coarse-grained weight pruning achieves a high compression ratio of 89.92% while improving the model test accuracy from the baseline of 92.21% to 95.08%, demonstrating its compression efficiency and potential regularization effect. The hybrid weight-channel pruning strategy achieves a maximum accuracy of 95.1%

(compression ratio of 27.70%) through a multi-grained collaborative mechanism, making it suitable for extreme performance optimization scenarios. However, structured pruning with high compression ratio (channel pruning compression ratio of 92.92%, layer pruning of 99.24%) results in accuracy losses of 5% and 21%, respectively, proving that aggressive structured pruning significantly damages model performance compared to unstructured methods. Specifically, layer pruning, which causes the accuracy to plummet to 71.42%, is confirmed as an infeasible compression scheme under the current model architecture.

Figure 10 Pruning method:final accuracy vs compression ratio [Owner-draw]

The experiment categorizes pruning algorithms into two major clusters based on performance characteristics: the Efficient Compression and Fast Inference Cluster (upper left region) includes Layer Pruning and Channel Pruning, both characterized by extremely short inference time (0.157 seconds/0.144 seconds) and high throughput (6354 samples/second/6922 samples/second), with the Layer Pruning achieving a compression rate of 99.2%. Such structured pruning significantly reduces computational complexity by removing complete network layers or convolutional kernel channels, but results in significant accuracy loss (Layer Pruning accuracy is only 71.42%, and Channel Pruning decreases by 5%), indicating that aggressive structured pruning, while improving compression rate and inference speed, severely disrupts the structural integrity of the model, making it unsuitable for accuracy-sensitive tasks. The High Precision Maintenance Cluster (lower right region) includes Coarse Weight Pruning and Hybrid Weight-Channel Pruning, both achieving high precision maintenance (95.08%/95.1%, surpassing the baseline) at the cost of longer inference time (0.363 seconds/0.394 seconds) and lower throughput. The Hybrid Pruning achieves a compression rate of only 27.7% (with the smallest point size), indicating that conservative pruning strategies, while having limited optimization effect on inference performance, effectively retain key information of the model through unstructured or fine-grained collaborative mechanisms, making them suitable for scenarios where extreme precision is pursued.

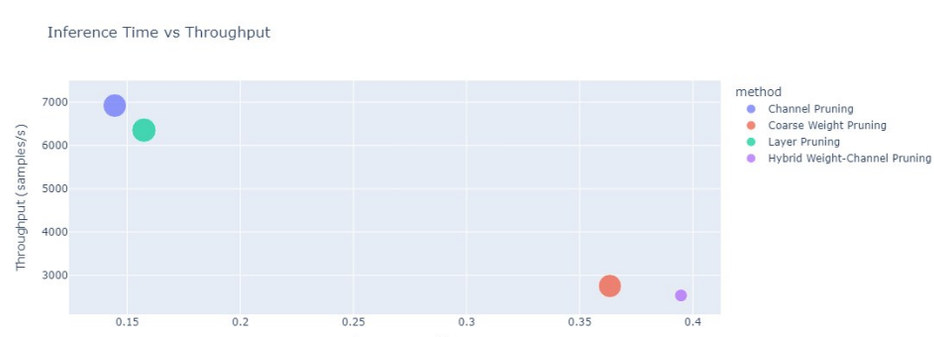

Figure 11 Inference time vs throughput [Owner-draw]

The experiment categorizes pruning strategies into three types based on performance characteristics: High-fidelity pruning strategies: Coarse-grained weight pruning and hybrid weight-channel pruning excel in the final accuracy (final_accuracy) dimension, with normalized scores reaching 1.0, indicating their effectiveness in maintaining model accuracy. Among them, the polygon of coarse-grained weight pruning is more balanced, reflecting its superior Pareto balance between accuracy and efficiency indicators. High-efficiency pruning strategies: Channel pruning and layer pruning exhibit significant advantages in compression ratio (compression_ratio) and throughput (throughput), which is a typical characteristic of the "anisotropy" of structured pruning—achieving efficient compression and acceleration by removing complete computational units (channels/layers). However, such strategies come at the cost of significant accuracy loss (especially layer pruning, where accuracy drops sharply to 71.42%), limiting their practicality. Performance

tradeoff quantification: Radar charts clearly quantify the inherent tradeoff between accuracy and efficiency. For example, although hybrid weight-channel pruning tops the list with the highest accuracy (95.1%), its compression ratio (27.7%) is the weakest among all methods, validating its strategy of sacrificing compression potential for high model fidelity.

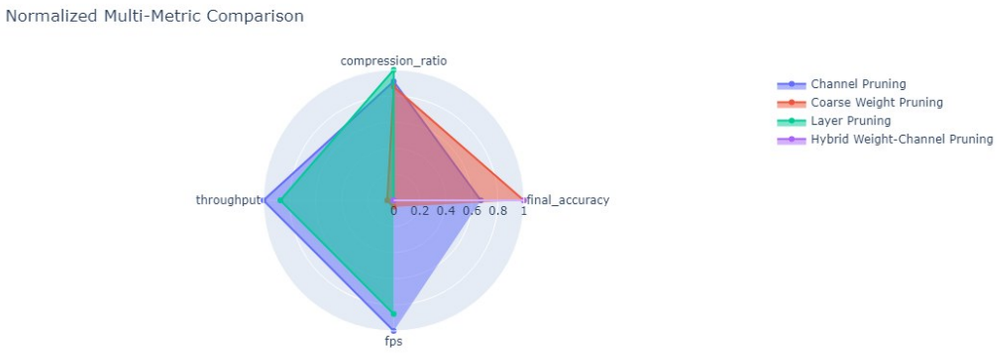

Figure 12 Normalized multl-metric comparison [Owner-draw]

The experiment categorizes four pruning algorithms into four types of feature clusters based on the spatial distribution of data points:

Coarse-grained weight pruning: It performs optimally in terms of accuracy (highest on the y=0 plane along the Z axis), but its inference time and compression ratio are relatively weak. It is an accuracy-first strategy, suitable for scenarios where zero tolerance is given to model performance degradation.

Mixed weight-channel pruning: Its accuracy is second only to coarse-grained pruning, while achieving a better balance between compression ratio and inference time (positional balance in three-dimensional space). It is a balanced strategy that sacrifices moderate accuracy for significant efficiency improvement.

Channel pruning: It excels in terms of compression ratio (y=1 plane) and inference time (y=2 plane) (structured pruning characteristics reduce computation), but there is a noticeable loss in accuracy. It is an efficient compression strategy, suitable for scenarios where model efficiency and deployment resources are strictly limited.

Layer pruning: achieves the highest compression ratio and the lowest inference time (leading in efficiency metrics), but comes with the most severe accuracy drop (bottom on the y=0 plane). It is an extreme compression strategy, suitable only for applications where the requirements for model size and speed are extremely stringent, and where a significant performance loss is acceptable.

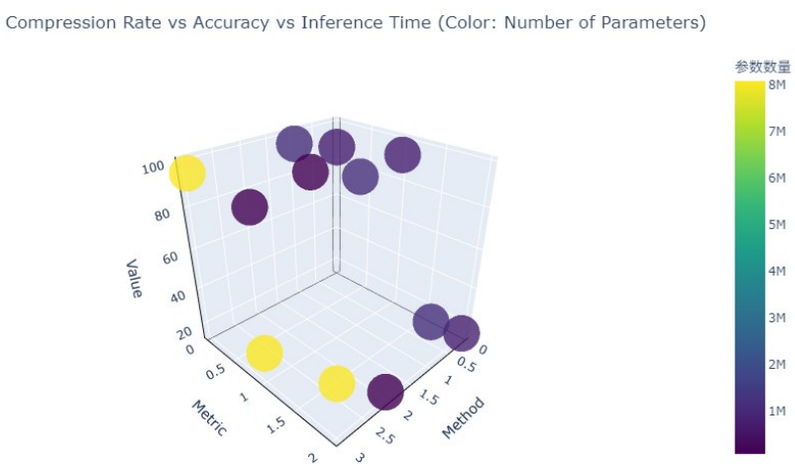

Figure 13 Compression rate vs accuracy vs inference time [Owner-draw]

5 Research conclusion

Addressing the challenges faced by deep neural network models when deployed on resource-constrained devices, such as large model size, high computational complexity, and significant inference latency, this paper systematically compares and evaluates four mainstream pruning algorithms: coarse-grained weight pruning, mixed weight-channel pruning, channel pruning, and layer pruning. Through multi-dimensional performance analysis on standard datasets, this

study aims to reveal the characteristics and trade-offs of different pruning strategies in key indicators such as model accuracy, compression ratio, and inference efficiency, providing empirical evidence for algorithm selection in specific application scenarios.

Experimental results demonstrate that different pruning algorithms exhibit significant heterogeneity in performance. Coarse-grained weight pruning has a clear advantage in maintaining high accuracy, but it has limited improvement on model efficiency; in contrast, structured pruning methods such as channel pruning and layer pruning can significantly increase model compression ratio and inference speed, but usually come with more significant accuracy loss. The hybrid pruning strategy strikes an effective balance between accuracy and efficiency, showing good application potential.

References

[1] LIU Y D. Research on multi-granularity pruning algorithms for neural networks[D]. Beijing University of Posts and Telecommunications, 2024.

[2] YANG K. Research on structured pruning and optimization algorithms for convolutional neural networks[D]. Zhongyuan University of Technology, 2024.

[3] Cheng Yingjie. Research on convolutional neural networks compression methods based on channel pruning[D]. Hunan University, 2023.

[4] SONG S. Research on compression and acceleration techniques for deep convolutional neural network models[D]. Beijing Jiaotong University, 2023.

[5] TAN J B, ZHAO M. Research on multi-dimensional compression algorithms for convolutional neural networks[J]. Journal of Yangtze University (Natural Science Edition), 2023, 20(5): 24-28.