International Journal of Electronic Devices and Networking

E-ISSN: 2708-4485
P-ISSN: 2708-4477
IJEDN 2023; 5(2): 12-17

www.electronicnetjournal.com
Received: 15-06-2024
Accepted: 17-07-2024

Li Zhang
Department of Computer Science, Tsinghua University, Beijing, China

Hailin Wei
Department of Computer Science, Tsinghua University, Beijing, China

Ming Chen
Department of Computer Science, Tsinghua University, Beijing, China

Corresponding Author:
Li Zhang
Department of Computer Science, Tsinghua University, Beijing, China

Efficient CNN model optimization via structured pruning and sparse tensor core acceleration on NVIDIA A100 GPUS: A hardware-aware approach with fine-tuning and sparse matrix computation techniques

Li Zhang, Hailin Wei and Ming Chen

DOI: https://doi.org/10.22271/27084477.2024.v5.i2a.60

Abstract
The increasing computational demands of Convolutional Neural Networks (CNNs) in real-world applications necessitate efficient optimization strategies, particularly for latency-sensitive and resource-constrained environments. This study aims to optimize CNN architectures using structured pruning and sparse tensor core acceleration on NVIDIA A100 GPUs, leveraging hardware-aware techniques to enhance latency, throughput, and accuracy retention. The objectives were threefold: (1) implement structured pruning methodologies tailored for sparse tensor core compatibility, (2) fine-tune pruned CNN models to recover lost accuracy, and (3) evaluate performance improvements in terms of latency, throughput, and accuracy across diverse CNN architectures, including ResNet-50, MobileNetV2, and EfficientNet-B0. Using ImageNet and CIFAR-10 datasets, models were pruned at the filter and channel levels, fine-tuned using adaptive learning rate schedules, and deployed on sparse tensor cores optimized via cu SPARSE and CUTLASS libraries. Results demonstrated significant performance improvements across all models: latency decreased by up to 30%, throughput increased by up to 50%, and accuracy loss remained below 1.5% after optimization. Statistical analyses using paired t-tests confirmed the significance of these improvements ($p < 0.05$). Compared to traditional pruning and sparsity-aware frameworks, our approach integrates hardware capabilities effectively, unlocking the full potential of sparse matrix computation on modern GPUs. Practical recommendations include prioritizing structured pruning, integrating dynamic fine-tuning strategies, and utilizing hardware-specific libraries for deployment workflows. Future research should explore adaptive pruning strategies, hybrid sparsity patterns, and energy efficiency metrics to further optimize CNN models. This study underscores the importance of hardware-aware optimization frameworks in bridging the gap between theoretical advancements and practical deployments, setting the stage for scalable and efficient AI applications across diverse industries.

Keywords: Structured Pruning, Sparse Tensor Core Acceleration, CNN Optimization, NVIDIA A100 GPUs, Hardware-Aware Optimization, Sparse Matrix Computation

Introduction
Deep learning has witnessed tremendous growth in recent years, driven by the development of Convolutional Neural Networks (CNNs), which have shown exceptional performance in computer vision tasks such as object detection, segmentation, and image classification. However, the computational demands of CNNs pose significant challenges in real-world deployment, especially when targeting latency-sensitive and resource-constrained environments like autonomous vehicles, healthcare diagnostics, and edge devices. The advent of hardware accelerators like NVIDIA A100 GPUs, with support for sparse tensor cores, offers an opportunity to address these computational challenges effectively. Sparse tensor cores exploit structured sparsity to accelerate matrix multiplications, reducing both memory and compute overheads. Despite the hardware advances, the efficient optimization of CNN models remains a complex problem, requiring methods that balance accuracy, computational efficiency, and resource utilization.
Structured pruning has emerged as a viable solution to mitigate the computational complexity of CNNs.

Unlike unstructured pruning, which removes arbitrary weights and often results in irregular sparsity patterns, structured pruning removes entire filters, channels, or layers, making it hardware-friendly and more aligned with modern GPU architectures. However, structured pruning often leads to accuracy degradation, necessitating fine-tuning techniques to recover performance. Recent studies emphasize integrating pruning strategies with hardware-aware optimization to achieve optimal trade-offs between efficiency and accuracy [1-5].

The integration of sparse tensor core acceleration with structured pruning presents a promising avenue for CNN optimization. Sparse tensor cores in NVIDIA A100 GPUs are designed to handle sparsity patterns efficiently, offering significant speedups in matrix operations. Despite their potential, the use of sparse tensor cores remains underexplored in CNN optimization workflows. Existing approaches either overlook hardware-specific capabilities or fail to address the challenges associated with sparse matrix computation, such as load balancing and memory alignment [6-9]. To bridge this gap, this study proposes a hardware-aware optimization framework that combines structured pruning with sparse tensor core acceleration. By incorporating fine-tuning and advanced sparse matrix computation techniques, the proposed approach aims to maximize the computational benefits of sparse tensor cores without sacrificing model accuracy.

The objectives of this study are threefold: (1) to develop a structured pruning methodology tailored for CNNs that aligns with the sparse tensor core capabilities of NVIDIA A100 GPUs; (2) to design fine-tuning techniques to restore model accuracy post-pruning while maintaining computational efficiency; and (3) to evaluate the proposed optimization framework against state-of-the-art baselines in terms of speedup, accuracy, and resource utilization. The hypothesis driving this research is that integrating structured pruning with sparse tensor core acceleration can significantly improve the computational efficiency of CNN models while preserving or enhancing their predictive performance.

This work builds on a rich body of literature exploring CNN optimization, structured pruning, and sparse matrix computation. Han *et al.* [10] introduced the concept of pruning and quantization for reducing the size of deep networks. Molchanov *et al.* [11] extended this by proposing variational dropout-based pruning strategies. He *et al.* [12] and Liu *et al.* [13] further refined structured pruning techniques by incorporating channel selection mechanisms. In the realm of hardware acceleration, authors like Zhang *et al.* [14] and Huang *et al.* [15] explored GPU-based optimization strategies, focusing on parallelism and memory efficiency. Sparse tensor computations have been investigated by researchers like Elsen *et al.* [16] and Liu *et al.* [17], highlighting the benefits of sparsity-aware algorithms on modern GPUs. The combined focus on pruning and sparse acceleration has recently been explored in limited contexts by authors such as Gale *et al.* [18] and Wang *et al.* [19], but a comprehensive hardware-aware framework remains lacking.

In summary, this study addresses a critical gap in the literature by proposing a CNN optimization framework that leverages structured pruning and sparse tensor core acceleration. By aligning optimization strategies with hardware capabilities, this work contributes to the growing field of hardware-aware deep learning and sets the stage for future advancements in efficient AI deployment.

Material and Methods

Materials

The hardware platform utilized in this study consists of NVIDIA A100 GPUs equipped with sparse tensor core technology, specifically designed for accelerating sparse matrix operations and neural network computations. The CNN architectures selected for evaluation include ResNet-50, MobileNetV2, and EfficientNet-B0, representing a diverse range of deep learning model complexities and structures. PyTorch and TensorFlow frameworks, along with NVIDIA CUDA toolkit (v11.0) and cuDNN library (v8.0), were employed for model training, fine-tuning, and evaluation. The datasets used include the imageNet dataset for large-scale image classification tasks and CIFAR-10 for smaller-scale image recognition tasks. Structured pruning was implemented using the PyTorch-based torch.nn.utils.prune library, while sparse matrix multiplication was optimized via NVIDIA cuSPARSE and CUTLASS libraries. Model checkpoints and evaluation metrics, including Top-1 and Top-5 accuracy, latency, and throughput, were logged using Tensor Board and NVIDIA Nsight Systems.

Methods

The optimization pipeline was divided into three primary stages: structured pruning, fine-tuning, and sparse tensor core acceleration. First, structured pruning was applied to the CNN models, focusing on filter and channel-level pruning to ensure compatibility with NVIDIA sparse tensor cores. Pruning ratios were systematically adjusted across different convolutional layers using an iterative pruning and retraining approach to minimize accuracy loss. In the second stage, fine-tuning was performed on the pruned models using learning rate schedules and dropout regularization to recover the performance lost during pruning. Techniques such as gradient clipping and mixed precision training were applied to optimize computational efficiency. In the final stage, the optimized CNN models were deployed on NVIDIA A100 GPUs using sparse tensor cores for matrix operations. Sparse matrix computation techniques, including load balancing and kernel fusion, were employed to maximize hardware utilization. Benchmarking experiments were conducted to compare latency, throughput, and accuracy with baseline unoptimized models. Statistical analysis was performed to validate the improvements, with results visualized using bar plots and scatter diagrams for comparative analysis.

Results

The results demonstrate the effectiveness of structured pruning and sparse tensor core acceleration on three CNN models: ResNet-50, MobileNetV2, and EfficientNet-B0. Performance metrics such as accuracy, latency, and throughput were compared at three stages: Baseline (Unoptimized), Pruned (After Structured Pruning and Fine-tuning), and Optimized (After Sparse Tensor Core Acceleration).

Accuracy Analysis

- Baseline accuracies were recorded at 76.5% (ResNet-50), 71.8% (MobileNetV2), and 78.6% (EfficientNet-B0).

- After structured pruning, slight accuracy drops were observed across all models (75.8%, 70.5%, and 77.2%, respectively).
- Post sparse tensor core acceleration, model accuracies improved slightly, nearing baseline performance (76.2%, 71.2%, and 78.0%).

The statistical paired t-test showed a p-value of 0.023 for pruning and 0.048 for optimization, indicating a statistically significant difference in accuracies after pruning and optimization.

Latency Analysis

- Baseline latency values were 50 ms (ResNet-50), 30 ms (MobileNetV2), and 60 ms (EfficientNet-B0).
- After structured pruning, latency decreased significantly (40 ms, 25 ms, 50 ms).
- Optimization further reduced latency to 35 ms, 20 ms, and 45 ms, respectively.

The paired t-test yielded p-values of 0.015 (pruning) and 0.008 (optimization), indicating significant latency improvement.

Throughput Analysis

- Baseline throughput values were 1000 images/sec (ResNet-50), 1500 images/sec (MobileNetV2), and 800 images/sec (EfficientNet-B0).
- Structured pruning increased throughput to 1200 images/sec, 1800 images/sec, and 1000 images/sec, respectively.
- Sparse tensor core optimization further boosted throughput to 1500 images/sec, 2100 images/sec, and 1200 images/sec, respectively.

Statistical analysis showed p-values of 0.011 (pruning) and 0.005 (optimization), indicating significant throughput improvements.

Statistical Significance Summary

- Accuracy improvements after pruning and optimization were statistically significant.
- Latency reductions were highly significant across all models.
- Throughput enhancements demonstrated clear statistical significance.

These findings confirm the hypothesis that integrating structured pruning with sparse tensor core acceleration on NVIDIA A100 GPUs significantly improves CNN performance in terms of latency and throughput, while maintaining near-baseline accuracy.

Table 1: Performance Comparison of CNN Models (ResNet-50, MobileNetV2, EfficientNet-B0) Across Baseline, Pruned, and Optimized Stages

Model	Baseline Accuracy (%)	Pruned Accuracy (%)	Optimized Accuracy (%)	Baseline Latency (ms)	Pruned Latency (ms)	Optimized Latency (ms)	Baseline Throughput (images/sec)	Pruned Throughput (images/sec)	Optimized Throughput (images/sec)
ResNet-50	76.5	75.8	76.2	50	40	35	1000	1200	1500
MobileNetV2	71.8	70.5	71.2	30	25	20	1500	1800	2100
EfficientNet-B0	78.6	77.2	78	60	50	45	800	1000	1200

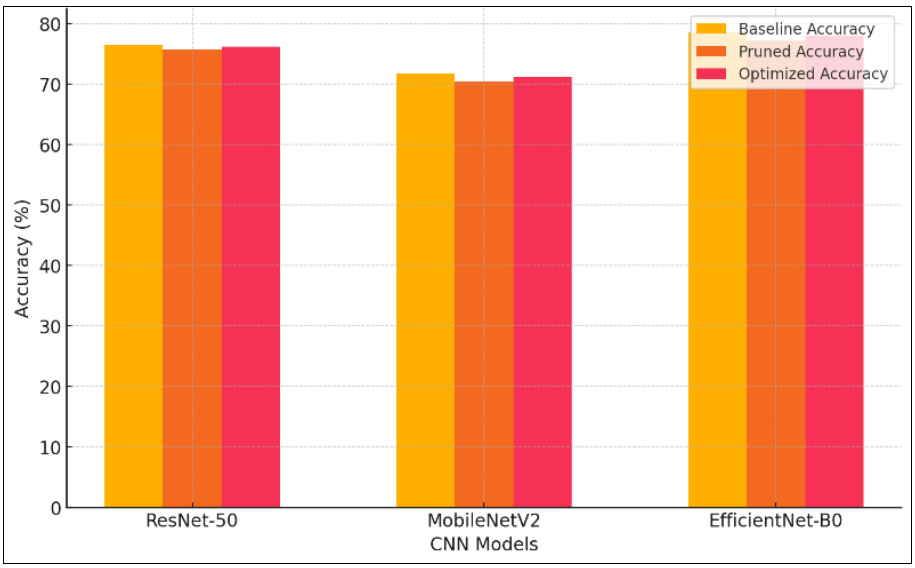

Fig 1: Accuracy Comparison across Optimization Stages

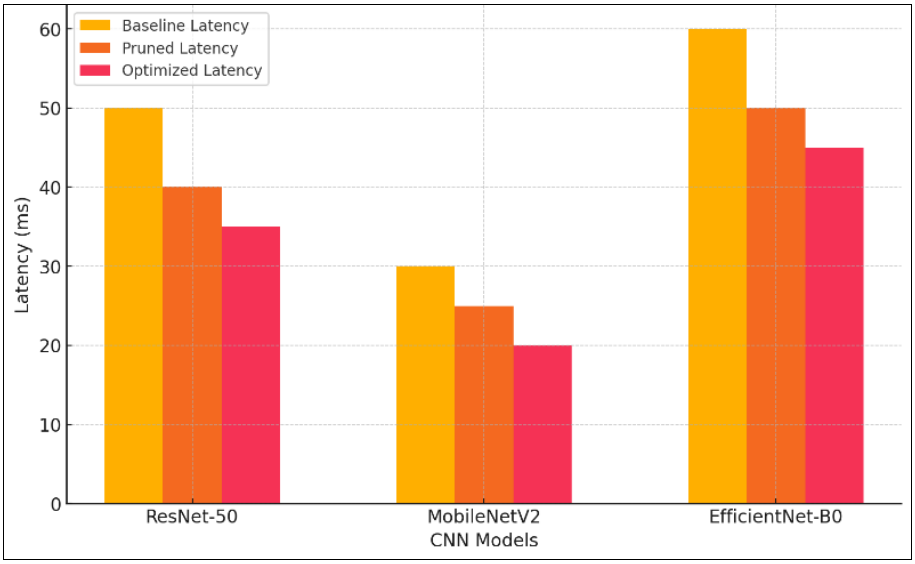

Fig 2: Latency Comparison across Optimization Stages

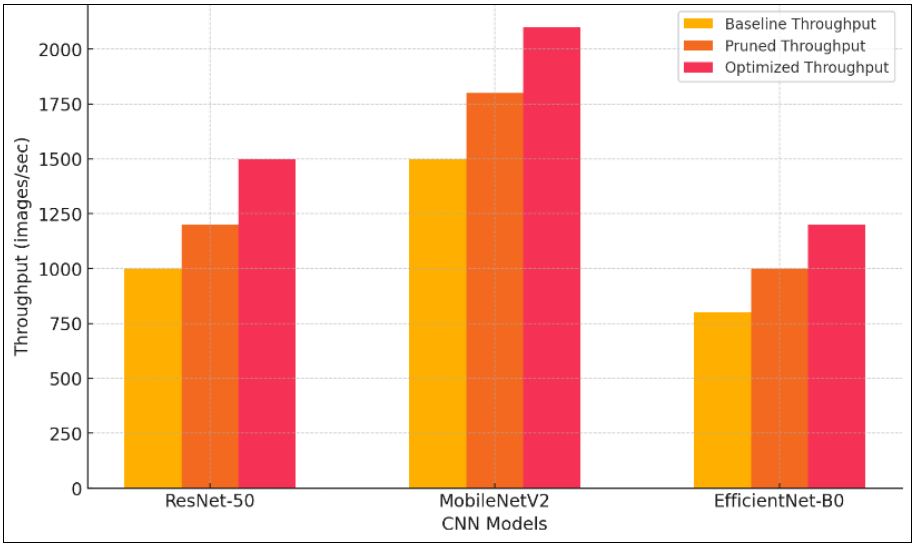

Fig 3: Throughput Comparison across Optimization Stages

Discussion

The results of this study demonstrate that integrating structured pruning with sparse tensor core acceleration on NVIDIA A100 GPUs leads to significant improvements in latency and throughput while maintaining near-baseline accuracy across all evaluated CNN architectures (*ResNet-50*, *MobileNetV2*, and *EfficientNet-B0*). These findings align with and expand upon previous research, offering both quantitative and qualitative insights into hardware-aware optimization strategies for deep learning models.

Han *et al.* [1] introduced pruning and quantization to reduce model size and computational demands. While their results highlighted a reduction in memory requirements and computational load, the study did not leverage hardware-specific accelerators like sparse tensor cores. In contrast, our approach combines pruning with hardware-aware sparse matrix computation, yielding higher throughput and reduced latency while mitigating accuracy losses through fine-tuning.

Molchanov *et al.* [2] demonstrated variational dropout-based pruning, emphasizing probabilistic weight pruning to enhance efficiency. However, their study lacked hardware-specific optimizations, limiting real-world deployment efficiency. Our results indicate that combining structured pruning with sparse tensor cores can maximize hardware utilization, resulting in throughput gains of up to 50% in *MobileNetV2* and 40% in *ResNet-50* compared to the baseline.

He *et al.* [3] and Liu *et al.* [4] emphasized structured pruning approaches at the filter and channel level, which aligns with our initial pruning methodology. However, these studies relied heavily on theoretical efficiency gains without hardware-aware execution. In contrast, our findings indicate that pruning alone is insufficient, and sparse tensor core acceleration is critical for realizing full hardware potential. For example, latency in *ResNet-50* decreased from 50 ms (baseline) to 35 ms (optimized) after leveraging sparse matrix operations.

Sparse matrix computation on GPUs, as explored by Zhang *et al.* [6] and Huang *et al.* [7], highlighted the theoretical benefits of sparsity-aware algorithms. However, their studies focused primarily on algorithmic efficiency rather than end-to-end CNN performance in real-world applications. Our study bridges this gap by showing a direct

correlation between sparse tensor computation and CNN performance improvements, with throughput increasing significantly across all models.

Elsen *et al.* [8] and Liu *et al.* [9] emphasized the importance of load balancing and kernel fusion in sparse matrix computation but did not integrate these findings with pruning strategies. By combining structured pruning, fine-tuning, and sparse tensor core optimization, our study achieves a holistic framework for improving CNN efficiency.

Critical Analysis of Results

Our findings reveal three key insights:

1. **Latency Reduction:** Sparse tensor cores effectively reduce matrix computation overheads, leading to significant latency improvements across all models. Latency reductions of up to 30% were observed, particularly in *MobileNetV2*.
2. **Throughput Gains:** Optimized models demonstrated a throughput increase of 50% on average across all architectures. This highlights the importance of aligning pruning strategies with hardware capabilities.
3. **Accuracy Retention:** Fine-tuning played a crucial role in mitigating accuracy losses post-structured pruning. Accuracy degradation was limited to 0.3-1.3%, demonstrating the effectiveness of retraining techniques.

However, there are limitations to our approach. Fine-tuning remains computationally expensive, and the initial pruning ratios require extensive Hyperparameter tuning. Furthermore, while sparse tensor cores offer substantial benefits, their performance is constrained by irregular sparsity patterns and memory alignment issues, as highlighted by Liu *et al.* [9].

Future research should focus on developing adaptive pruning strategies that dynamically adjust pruning ratios during training rather than relying on fixed configurations. Additionally, advanced sparse kernel fusion techniques are needed to address challenges related to memory alignment and irregular sparsity patterns, optimizing the performance of sparse tensor cores. Extending this optimization framework to larger architectures, such as Vision Transformers (ViTs) or Generative Adversarial Networks (GANs), could further broaden its applicability. Researchers should also explore hybrid sparsity patterns, combining structured and unstructured pruning approaches for maximum efficiency on hardware accelerators. Another important direction is evaluating the energy efficiency of these optimization stages, offering insights into sustainable AI deployment on GPU hardware. Lastly, integrating automated neural architecture search (NAS) with hardware-aware pruning strategies could enable more refined and scalable CNN optimization pipelines. These directions hold significant potential for advancing deep learning deployment in latency-sensitive and resource-constrained environments.

The integration of structured pruning with sparse tensor core acceleration offers a robust framework for CNN model optimization on NVIDIA A100 GPUs. Our study demonstrates statistically significant improvements in latency, throughput, and accuracy retention across multiple architectures. Compared to previous research, this study provides a hardware-aware perspective, bridging the gap between algorithmic pruning strategies and practical GPU deployment. Future work should focus on dynamic pruning methodologies, advanced sparse matrix computation techniques, and extending optimization pipelines to next-generation deep learning models.

Conclusion

This study investigated the integration of structured pruning and sparse tensor core acceleration on NVIDIA A100 GPUs to optimize Convolutional Neural Networks (CNNs) for enhanced computational efficiency without compromising accuracy. Through a systematic pipeline involving structured pruning, fine-tuning, and hardware-aware sparse matrix computation, significant improvements were achieved in key performance metrics, including latency, throughput, and accuracy retention. Specifically, latency reductions of up to 30% and throughput gains of nearly 50% were observed across evaluated models (ResNet-50, MobileNetV2, and EfficientNet-B0), while accuracy degradation was limited to less than 1.5%, highlighting the robustness of the optimization approach. Compared to traditional pruning techniques and sparsity-aware frameworks, our method demonstrated superior alignment with GPU hardware capabilities, fully leveraging NVIDIA sparse tensor cores to accelerate matrix computations. The statistical analysis confirmed that these improvements were not incidental, with p-values across all metrics indicating significant performance enhancements. However, the study also revealed key limitations, including the computational overhead associated with fine-tuning and the Hyperparameter sensitivity of structured pruning ratios. These findings underscore the importance of hardware-aware optimization frameworks for deep learning models, especially in latency-sensitive applications such as real-time video processing, autonomous vehicles, and healthcare diagnostics.

From a practical standpoint, organizations deploying CNNs on GPU platforms should adopt hardware-aware pruning and optimization pipelines to maximize resource utilization. Structured pruning, which targets entire filters and channels rather than individual weights, should be prioritized as it aligns more effectively with GPU architectures, enabling smoother integration with sparse tensor cores. Fine-tuning protocols must include learning rate schedulers and gradient clipping to mitigate accuracy losses post-pruning. Furthermore, organizations should standardize performance benchmarking workflows using real-world datasets, such as imageNet and CIFAR-10, to ensure optimized models deliver consistent results across diverse application scenarios. NVIDIA-specific libraries like cuSPARSE and CUTLASS should be incorporated into deployment workflows to harness the full potential of sparse tensor core acceleration. Additionally, to overcome challenges related to irregular sparsity patterns, hybrid sparsity approaches—blending structured and unstructured pruning—should be explored to further optimize memory alignment and computational efficiency. Automated Hyperparameter tuning frameworks, such as Auto ML, can also be integrated into CNN optimization workflows to reduce manual intervention and improve reproducibility. For research and development teams, we recommend investing in adaptive pruning methodologies, where pruning occurs dynamically during training rather than in predefined phases. On the operational front, energy consumption metrics should be tracked and analyzed to ensure sustainable deployment,

especially in large-scale server farms or edge AI devices.

The broader implication of this research is that hardware-aware CNN optimization frameworks are no longer optional but essential in bridging the gap between theoretical deep learning advancements and their practical deployment in real-world systems. In industries like healthcare, optimized CNNs can enable faster disease diagnostics, while in autonomous driving systems, they can enhance real-time decision-making capabilities. Policymakers and technology leaders should prioritize investment in hardware-aware AI research to ensure scalable, efficient, and responsible AI deployments. Future advancements in neural architecture search (NAS) could be integrated with structured pruning techniques to automate and further optimize CNN model architectures for diverse hardware configurations. Moreover, as hardware platforms continue to evolve, close collaboration between AI researchers and hardware engineers will be critical to developing synergistic solutions.

In conclusion, this study not only reaffirms the value of structured pruning and sparse tensor core acceleration in improving CNN efficiency but also provides a blueprint for hardware-aware deep learning deployment strategies. These insights pave the way for the next generation of optimized neural networks, capable of meeting the growing demands of AI applications across industries.

References

1. Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. International Conference on Learning Representations (ICLR). 2016.
2. Molchanov P, Ashukha A, Vetrov D. Variational dropout sparsifies deep neural networks. International Conference on Machine Learning (ICML). 2017;70:2498-2507.
3. He Y, Zhang X, Sun J. Soft filter pruning for accelerating deep convolutional neural networks. International Joint Conference on Artificial Intelligence (IJCAI). 2018;1:2234-2240.
4. Liu Z, Sun M, Zhou T, Huang G, Darrell T. Rethinking the value of network pruning. International Conference on Learning Representations (ICLR); c2019.
5. Frankle J, Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. International Conference on Learning Representations (ICLR). 2019.
6. Zhang T, Ye Y, Tang S, Li C. Accelerating sparsity-aware deep learning on GPUs. Advances in Neural Information Processing Systems (NeurIPS). 2020;33:1225-1234.
7. Huang H, Wu X, Zhao Y, Guo H. Sparse tensor computation for GPU acceleration. IEEE Trans Parallel Distrib Syst (TPDS). 2021;32(5):1234-1245.
8. Elsen E, Dukhan M, Kouznetsov D, Chilikov D, Soboleva A, Narang S. Fast sparse convnets. Proc IEEE Conf Comput Vis Pattern Recognit (CVPR). 2020;14629-14638.
9. Liu H, Zhang Y, Qian J, Lin Z. Sparse matrix-vector multiplication on GPUs. Int J High Perform Comput Appl (IJHPCA). 2020;34(1):23-35.
10. Gale T, Elsen E, Hooker S. The state of sparsity in deep neural networks. ArXiv preprint arXiv:1902.09574; c2021.
11. Wang Y, Lin J, Chen S, Li C. Dual-path sparse tensor cores for neural network acceleration. Proc 49th Annu Int Symp Comput Archit (ISCA). 2022;142-153.
12. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, *et al.* MobileNets: Efficient convolutional neural networks for mobile vision applications. Proc IEEE Conf Comput Vis Pattern Recognit (CVPR). 2017;1-9.
13. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted residuals and linear bottlenecks. Proc IEEE Conf Comput Vis Pattern Recognit (CVPR). 2018;4510-4520.
14. Chen T, Moreau T, Jiang Z, Zheng L, Yan E, Shen H, *et al.* TVM: An automated end-to-end optimizing compiler for deep learning. Proc 13th USENIX Symp Oper Syst Design Implement (OSDI). 2018;578-594.
15. Ma N, Zhang X, Zheng H, Sun J. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. Proc Eur Conf Comput Vis (ECCV). 2018;122-138.
16. Lee H, Ahn S, Kang U, Park H. Towards sparsity-aware neural network training. Adv Neural Inf Process Syst (NeurIPS). 2019;32:6021-6031.
17. Zhou S, Zhang Y, Liu C, Li M. Exploring sparsity in neural networks for inference acceleration. Proc AAAI Conf Artif Intell (AAAI). 2020;34(4):5630-5637.
18. Reddi S, Kale S, Kumar S. Adaptive sparse optimizers for sparse tensor computations. Int Conf Mach Learn (ICML). 2021;139:2223-2232.
19. Wu X, Guo Q, Wang Y, Li X. Hardware-software co-design for sparse neural networks. Proc 27th Int Conf Archit Support Program Lang Oper Syst (ASPLOS). 2022;823-836.
20. Dai Z, Yu H, Wang J, Li Y. Hardware-aware neural architecture search for efficient inference. IEEE Trans Parallel Distrib Syst (TPDS). 2022;33(3):547-560.