An In-Depth Overview of Existing Quantization Strategies for Neural Networks

Vishakha Agrawal

vishakha.research.id@gmail.com

Abstract

Neural network quantization has emerged as a crucial technique for efficient deployment of deep learning models on resource-constrained devices. This paper provides a detailed survey of existing quantization strategies, analyzing their theoretical foundations, algorithmic details, and empirical performance. We compare and contrast various quantization techniques, including post-training quantization, quantization- aware training, and knowledge distillation-based methods, to provide insights into their strengths, limitations, and applications.

Keywords: Quantization, QAT, PTQ, Dynamic Quantization, Fixed-Point Quantization, Mixed-Precision Quantization

I. INTRODUCTION

Neural network deployment on resource-constrained devices has become increasingly important in recent years. Quantiza- tion, the process of reducing the numerical precision of model weights and activations, has emerged as a crucial technique for model compression and acceleration. This paper examines the landscape of quantization strategies, their mathematical foundations, and their practical applications.

II. MOTIVATION

The deployment of deep neural networks in edge devices and mobile applications faces several challenges:

- Limited memory and storage capacity
- Power consumption constraints
- Real-time processing requirements
- Bandwidth limitations for model distribution

Quantization [8], [5], [6] addresses these challenges by reduc- ing model size and computational complexity while striving to maintain model performance.

III. FUNDAMENTAL CONCEPTS

1) Numerical Precision in Neural Networks: Traditional neural networks typically use 32-bit floating-point (FP32) representation for weights and activations. This high precision, while beneficial during training, often exceeds the requirements for inference. Extensive re- search has demonstrated that neural networks can maintain acceptable performance with significantly reduced numerical precision. The key lies in understanding the trade-offs between precision and accuracy, and how different parts of the network respond to reduced precision operations.

2) Quantization Process: The quantization process involves mapping continuous, high-precision values to a discrete set of lower-precision values. This mapping can be

represented mathematically as: Q(x) = round(x/S) * S Where Q(x) is the quantized value, x is the original value, and S is the scaling factor. This fundamental equation underlies most quantization schemes, though various methods may modify or extend it to achieve specific objectives. The scaling factor S plays a crucial role in determining the range and granularity of the quantized values, directly impacting the model's final performance.

IV. Types of Quantization Strategies

1) Post-Training Quantization (PTQ) : Post-training quantization applies quantization to a pre-trained model with- out requiring retraining[12]. This approach offers minimal additional training overhead and preserves the original training pipeline, making it particularly attractive for rapid deployment scenarios[1]. However, the simplicity of PTQ comes with potential drawbacks. Models may experience significant accuracy degradation, particularly with aggressive quantization schemes. The success of PTQ largely depends on the model architecture, task complexity, and the chosen quantization parameters.
2) Quantization-Aware Training (QAT) : Quantization- aware training [9] incorporates quantization effects during the training process, allowing the network to adapt to reduced precision. This method simulates quantization in the forward pass while maintaining full precision in the backward pass, enabling finer optimization of quantized weights. QAT typically achieves better accuracy than PTQ, especially for more aggressive quantization schemes. The trade-off comes in the form of increased training time and computational resources required for the training process.
3) Dynamic Quantization : Dynamic quantization [11] determines quantization parameters at runtime:
   - Adapts to changing activation distributions
   - Reduces storage requirements
   - May increase runtime computational overhead

V. Precision Schemes

1) Fixed-Point Quantization : Fixed-point quantization rep- resents a fundamental approach to reducing model precision by using integer arithmetic for computation. The most widely adopted format is INT8, which has be- come the de facto standard for inference due to its excellent balance between precision and efficiency. More aggressive quantization approaches include INT4, which has emerged as a promising ultra-low precision format for scenarios requiring extreme model compression. At the extreme end of the spectrum lie binary and ternary quantization schemes, which reduce weights to just one or two bits. While these extreme approaches achieve maximum compression, they often require specialized training techniques and architectural modifications to maintain acceptable accuracy.
2) Mixed-Precision Quantization: Mixed-precision quantization represents a more nuanced approach that recognizes the varying sensitivity of different network components to precision reduction [10]. This method assigns different bit-widths to different layers or operations based on their impact on model performance [4]. The allocation of precision can be determined through sensitivity analysis, hardware constraints, or optimization- based approaches that consider both accuracy and efficiency objectives. This flexible approach often achieves better results than uniform quantization, though it introduces additional complexity in both implementation and deployment.

VI. IMPLEMENTATION CONSIDERATIONS

1) Hardware Compatibility: Hardware compatibility plays a crucial role in determining the success of quantization strategies in real-world applications. Different hardware platforms offer varying levels of support for quantized operations. Modern CPUs typically provide optimized instructions for INT8 operations, while GPUs may offer specialized acceleration capabilities for different precision levels. Custom accelerators and edge devices often impose specific constraints on the types of operations and precisions they can efficiently handle. Understanding these hardware constraints is essential for developing practical quantization strategies that deliver real-world performance benefits.
2) Calibration Methods: Calibration represents a critical step in the quantization process that significantly impacts the final model performance. The selection of calibration data must carefully balance representation of the tar- get distribution with practical constraints on calibration time and resources. Statistical analysis of activations helps determine optimal quantization parameters, while range optimization techniques ensure efficient use of the available numerical precision. Advanced calibration approaches may employ error minimization strategies that consider the entire network's behavior rather than optimizing each layer in isolation. The development of robust calibration methods remains an active area of re- search, particularly for challenging cases such as outlier- heavy distributions and dynamic range requirements.

VII. PERFORMANCE ANALYSIS

Accuracy Impact: The impact of quantization on model accuracy varies significantly across different tasks and architectures. Classification tasks have demonstrated remarkable resilience to 8-bit quantization, often maintaining accuracy within 1-2 percent of full-precision models. Detection tasks, particularly those involving precise localization, typically show higher sensitivity to precision reduction and may require careful tuning or higher bit- widths in critical layers. Natural language processing tasks present unique challenges due to their diverse computational patterns and the importance of maintaining precise attention mechanisms. Understanding these task- specific considerations enables more effective quantization strategies tailored to particular applications.

1) Computational Efficiency: Quantization delivers multiple computational benefits that make it particularly attractive for resource-constrained deployments. The reduction in memory bandwidth requirements significantly decreases power consumption and improves cache utilization. Lower precision operations enable faster inference times through increased arithmetic intensity and better use of vector processing units. The reduced model size not only facilitates deployment on devices with limited storage but also improves distribution efficiency and update processes. These benefits compound in edge computing scenarios where multiple constraints must be satisfied simultaneously.

VIII. ADVANCED TECHNIQUES

1) Knowledge Distillation with Quantization : Knowledge distillation [7] has emerged as a powerful complement to quantization, enabling more effective compression through the transfer of knowledge from full-precision to quantized models. The teacher-student framework allows the quantized model to learn from the rich representations of a full-precision network, often achieving better performance than direct quantization. Feature- level distillation provides additional supervision signals that help maintain the discriminative power of inter- mediate representations. The combination of distillation and quantization represents a promising direction for achieving extreme compression while preserving model capability.

2) Neural Architecture Search for Quantization: The application of neural architecture search to quantization has opened new possibilities for automatically discovering efficient[2], quantization-friendly network architectures. Hardware-aware search strategies [3] incorporate deployment constraints directly into the architecture optimization process, resulting in models that are inherently more suitable for quantized execution. Precision- constrained optimization enables joint exploration of architectural choices and quantization parameters, while efficiency-accuracy trade-off exploration helps identify optimal operating points for specific deployment scenarios. This automated approach to quantization-aware architecture design represents a significant advance to- ward more systematic development of efficient neural networks.

IX. FUTURE DIRECTIONS

1) Research Opportunities: The field of neural network quantization continues to present numerous compelling research opportunities that warrant further investigation. Ultra-low precision techniques represent a particularly promising direction, as they push the boundaries of how efficiently neural networks can operate. The development of adaptive quantization schemes that can dynamically adjust to changing computational demands and data distributions remains an open challenge. The intersection of hardware and software design presents another fertile ground for innovation, as closer integration between these domains could yield significant improvements in quantized model performance. The the-oretical understanding of quantization effects on neural networks also remains incomplete, with opportunities to develop more robust mathematical frameworks for analyzing and predicting quantization impacts.

2) Emerging Applications: The landscape of applications driving quantization research continues to evolve rapidly with the proliferation of AI in edge computing scenarios. The deployment of AI models on edge devices presents increasingly complex challenges as applications demand more sophisticated capabilities within strict resource constraints. Internet of Things (IoT) devices represent a particularly demanding use case, requiring extremely efficient model execution while maintaining reliability across diverse operating conditions. Mobile applications continue to push the boundaries of what's possible with on-device AI, creating demand for more sophisticated quantization techniques that can enable complex models to run efficiently on mobile processors. Real-time systems present additional challenges, as they require not only efficient execution but also consistent and predictable performance under tight timing constraints. These emerging applications are driving innovation in quantization techniques and will likely continue to shape the direction of research in this field.

X. CONCLUSION

Neural network quantization has evolved from a simple compression technique to a sophisticated field that en- compasses various strategies, methodologies, and theoretical frameworks. The importance of quantization continues to grow as deep learning models become more prevalent in resource- constrained environments. While significant progress has been made in developing various quantization strategies, important challenges remain in achieving optimal trade-offs between model size, computational efficiency, and accuracy. The future of quantization research appears particularly promising, with emerging techniques in automated quantization, hardware- aware design, and theoretical understanding pointing toward more sophisticated approaches. The growing demand for efficient AI deployment in edge devices and mobile applications ensures that quantization will remain a crucial area of research and development in the coming years. As the field continues to mature, we can expect to see even more innovative solutions that better

balance the competing objectives of model performance and resource efficiency, ultimately enabling more widespread deployment of AI systems across a diverse range of applications and environments.

REFERENCES

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] Yukang Chen, Gaofeng Meng, Qian Zhang, Xinbang Zhang, Liangchen Song, Shiming Xiang, and Chunhong Pan. Joint neural architecture search and quantization. *arXiv preprint arXiv:1811.09426*, 2018.

[3] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

[4] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 293–302, 2019.

[5] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(187):1–30, 2018.

[6] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer- arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[7] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

[8] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[9] Alicja Kwasniewska, Maciej Szankin, Mateusz Ozga, Jason Wolfe, Arun Das, Adam Zajac, Jacek Ruminski, and Paul Rad. Deep learning optimization for edge devices: Analysis of training quantization param- eters. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 96–101. IEEE, 2019.

[10] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Va- jda, and Kurt Keutzer. Mixed precision quantization of convnets via dif- ferentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018.

[11] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2- NIPS)*, pages 36–39. IEEE, 2019.

[12] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *International conference on machine learning*, pages 7543–7552. PMLR, 2019.