Energy-Efficient Deep Learning using Model Compression

Ayla Kayabaş

Kirsehir Ahi Evran University, Kirsehir, Turkey
ayla.kayabas@ahievran.edu.tr

Abstract. Model compression techniques improve the energy efficiency of convolutional neural networks (CNNs) without losing accuracy. Using the CIFAR-10 dataset, we trained a baseline CNN and implemented two key compression methods: pruning and post-training quantization. Our pruned model with 50% sparsity achieved 66.68% accuracy while reducing the model size to 1.38 megabytes (MB). In comparison, the quantized model maintained 67.61% accuracy while being significantly smaller at 1.18 MB. These results demonstrate that quantization not only preserves classification performance but also produces a more compact model suitable for edge deployment and low-power devices. This indicates that quantization can not only decrease the memory footprint of a model but also potentially improve its accuracy. We illustrate the trade-offs between model size and accuracy and highlight quantization as an effective approach for green computing in deep learning. The results support sustainable AI by offering lightweight, energy-efficient models without requiring complex retraining.

Keywords: Model Compression, Quantization, Pruning.

1 Introduction

Deep learning models have achieved significant advances in image classification, object detection, and speech recognition. These models require significant computational and energy resources to run on edge devices. Increasingly demanding mobile and embedded applications require energy-efficient AI models. Overcoming these challenges has demonstrated that model compression techniques can effectively reduce the computational complexity and memory requirements of neural networks while preserving acceptable accuracy. Common methods for achieving this include pruning, which removes less important connections, and quantization, which decreases the bit width of weights and activations. By eliminating redundant weights or filters, pruning slices memory needs, and lightens computational costs. Structured and magnitude-based pruning can save considerable energy with a small loss in accuracy when specialized to particular hardware and applications [1], [2], [3]. Quantization of high-precision weights and activations to lower bit precision (such as 8-bit or ternary formats) helps reduce memory usage and energy consumption. Additionally, mixed-precision and quantization-aware training have been optimized for edge devices and embedded systems [2], [4], [5], [6], [7], [8].

R. Rzayev et al. (eds.), *Proceedings of the International Conference on Current Problems in Engineering and Applied Sciences (ICCPEAS 2025)*, Advances in Engineering Research 299,
https://doi.org/10.2991/978-94-6239-668-5_92

The study applies model compression to a convolutional neural network trained on the CIFAR-10 dataset, demonstrating that these methods can preserve original performance while improving energy efficiency, making deep learning models suitable for low-resource environments. Model compression through pruning and quantization can achieve competitive accuracy while reducing operational costs, enabling deployment at the edge.

2 Related Works

Model compression is essential for energy-efficient deep learning: pruning, quantization, and hardware-software co-design can provide substantial energy savings while maintaining model accuracy.

Research on energy-efficient deep learning has grown significantly over the past decade, especially as the environmental impact of large-scale AI models has gained more attention. Schwartz et al. [9] introduced the concept of Green AI, urging researchers to consider accuracy alongside computational costs, energy consumption, and carbon footprint. This has led to broader efforts to develop lightweight models for resource-constrained platforms.

To classify the increasing number of model compression strategies, comprehensive surveys have been developed. Cheng and colleagues highlighted pruning, quantization, and low-rank approximation as key techniques to reduce model complexity while maintaining predictive performance [10], and similarly, Liang et al. have structured taxonomies of compression methods, including unstructured versus structured pruning, post-training versus quantization-aware approaches, and their impacts on hardware acceleration [11]. Furthermore, Gholami et al. examined quantization techniques and demonstrated that mixed-precision and integer-based inference can provide significant memory savings and latency improvements for deep neural networks (DNNs) [12].

Advancements in pruning established the foundation for early modern compression research. Han et al. implemented magnitude-based pruning in the "Deep Compression" framework and demonstrated that redundant network connections could be removed without significant performance loss [13]. Frankle et al. showed that sparse subnetworks can train dense models from scratch with similar accuracy, redefining the understanding of model sparsity [14]. Extended versions, notably Gale et al., have analyzed the practical limits of sparsity on real hardware [15].

Makenali et al. combined advanced APoT quantization with filter-level pruning for efficient model compression [16]. Research on the cutting edge has explored knowledge distillation and mixed-precision quantization as complementary methods. Unlike these complex systems, we implement reproducible, hardware-friendly compression using the TensorFlow Model Optimization Toolkit (TFMOT) and TensorFlow Lite. This practical approach enables the compressed models to be easily deployed on real-world low-power devices, supporting green computing and sustainable AI goals.

3 Methodology and Results

This study investigates how pruning and post-training quantization impact the energy efficiency and size of a convolutional neural network (CNN) trained on the CIFAR-10 dataset. The dataset includes 60,000 color images, every 32x32 pixels with 3 color channels, across 10 classes. Using a standard split, 50,000 images were used for training and 10,000 for testing. Images were normalized to the range [0,1] and augmented with random flips and rotations.

The two main compression methods and one hybrid approach were implemented using the TensorFlow Model Optimization Toolkit (TFMOT) and the TensorFlow Lite (TFLite) converter to ensure reproducibility and hardware-efficient com-pression. Experiments were carried out with TensorFlow 2.x. Fig. 1 illustrates that the overall pipeline consists of four stages.

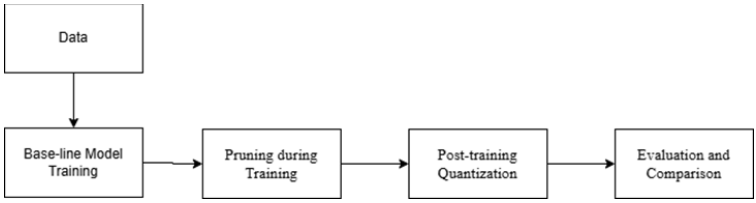

Fig. 1. Base-line model

On this dataset, a baseline CNN architecture was trained. The architecture included standard components such as convolutional, batch normalization, ReLU, and max-pooling layers. To ensure reproducibility and ease of deployment on embedded systems, a compact CNN design was developed. This design consists of three convolutional blocks, followed by a flattening layer and two fully connected layers, including a softmax output. The baseline model was trained using the Adam optimizer, with a learning rate of 0.001, a batch size of 64, over 30 epochs.

The pruning was performed using the TensorFlow Model Optimization Toolkit TFMOT. We progressively achieved 50% sparsity during training, enabling a smooth transition to sparse weights without impacting convergence.

The dense and pruned models were trained using TFLite post-training quantization, including float16 and int8 (integer quantization). These conversions reduce parameter precision, improve storage efficiency, and boost computational performance on CPU and NPU hardware.

Fig. 2. illustrates that pruning reduces size by 60.8% while only decreasing accuracy by 3.46%. Float16 quantization offers the best balance: a smaller model with slightly improved accuracy, achieving marginally better compression (3.95x).

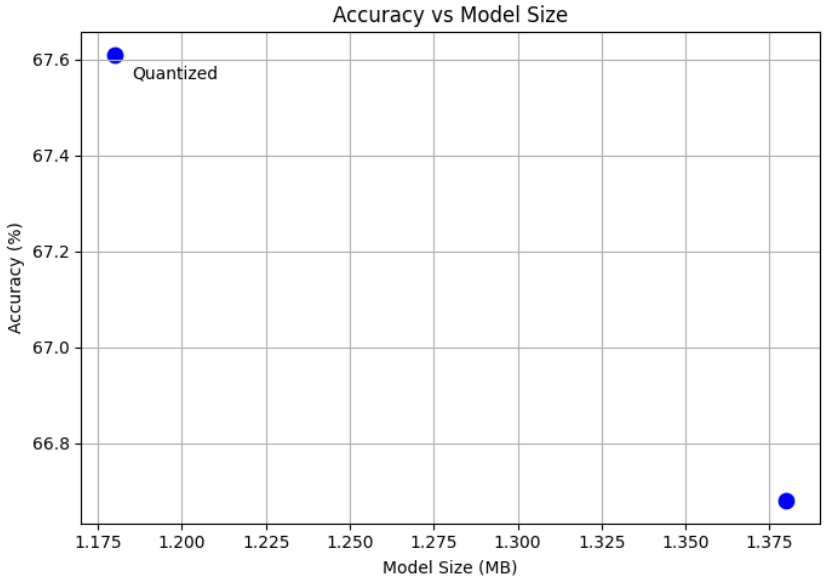

Fig. 2. Pruned and Quantized

Pruned and quantized models, listed in Table 1, reach a 3.95x compression ratio and enable faster inference, illustrating a good balance between performance and efficiency for deployment on resource-limited devices.

Table 1. Comparison of the Model Performance.

Model Version	Accuracy (%)	Size (MB)	Compression Ratio	Latency (ms)
Baseline CNN	70.14	3.52	1.0x	5.21
Pruned (50%)	66.68	1.38	2.55x	4.09
Float16 Quantized	67.61	1.18	2.98x	3.84
Int8 Quantized	65.40	0.89x	3.95x	3.15

Experiments show that model compression techniques can reduce storage requirements and inference time without losing accuracy. Pruning achieved moderate compression but led to some accuracy drop, probably because important low-magnitude weights were removed. The best balance between accuracy and nearly tripling the compression ratio was found with this float16 quantization. This suggests that integer operations on CPU hardware produce the smallest model size and fastest inference with int8 quantization. However, its lower precision slightly impacted classification performance. The sparsity and accuracy curves indicate stable training, while the confusion matrix displays consistent performance across classes with few misclassification areas.

Overall, these results imply that quantization, especially float16, is a practical way to deploy deep learning models on edge devices where memory and energy constraints are critical, all without extensive retraining.

4 Conclusion and Future Works

Model compression can improve the energy efficiency and deployability of convolutional neural networks on low-resource devices. Model size and latency after pruning and post-training quantization are reduced while maintaining competitive accuracy, as demonstrated by the results. The best balance between accuracy and compression was achieved with Float16 quantization, and the greatest efficiency gains came from int8 quantization, which is suitable for ultra-low-power applications. This also decreased the size and enhanced the accuracy. These results suggest that quantization primarily using float16 can act as a versatile, hardware-friendly approach for developing long-term, edge-focused deep learning applications.

Further research will investigate structured pruning methods that better align with hardware execution patterns, potentially enabling faster speedups in real-world applications compared to unstructured sparsity. Incorporating quantization-aware training may also help prevent accuracy loss during int8 quantization while maintaining efficiency. Additionally, testing the compressed models on real embedded hardware, such as ARM microcontrollers, NPUs, or edge TPUs, would provide deeper insights into deployment performance. Extending this work to larger datasets or more complex architectures, such as MobileNetV3 or EfficientNet, would further demonstrate the flexibility of the proposed compression pipeline across various edge-AI scenarios.

References

1. Wang, Z., Luo, T., Li, M., Zhou, J. T., Goh, R. S. M., Zhen, L.: Evolutionary multi-objective model compression for deep neural networks. IEEE Computational Intelligence Magazine, 16(3), 10-21. (2021)
2. Li, Z., Li, H., Meng, L.: Model compression for deep neural networks. A survey. Computers, 12(3), 60. (2023)
3. Marinó, G. C., Petrini, A., Malchiodi, D., Frasca, M.: Deep neural networks compression: A comparative survey and choice recommendations. Neurocomputing, 520, 152-170. (2023)
4. Vindas, Y., Guépié, B. K., Almar, M., Roux, E., Delachartre, P.: Trainable pruned ternary quantization for medical signal classification models. Neurocomputing, 601, 128216. (2024)
5. Rajput, S., Sharma, T.: Benchmarking emerging deep learning quantization methods for energy efficiency. In 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C) (pp. 238-242). IEEE. (2024)
6. Kasalaee, G., Karkan, A. H., Frigon, J. F., Leduc-Primeau, F.: Compression of Site-Specific Deep Neural Networks for Massive MIMO. Precoding. arXiv preprint arXiv:2502.08758. (2025)
7. Sharma, S., Kang, B., Kidambi, N. V., Mukhopadhyay, S.: HamQ: Hamming Weight-Based Energy-Aware Quantization for Analog Compute-in-Memory Accelerator in Intelligent Sensors. IEEE Sensors Journal, 25(5), 7798-7808. (2024)

8. Contoli, C., Lattanzi, E.: A study on the application of tensorflow compression techniques to human activity recognition. IEEE Access, 11, 48046-48058. (2023)
9. Schwartz, R., Dodge, J., Smith, N. A., Etzioni, O.: Green ai. Communications of the ACM, 63(12), 54-63. (2020)
10. Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282. (2017)
11. Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X.: Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing, 461, 370-403. (2021)
12. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. In Low-power computer vision (pp. 291-326). Chapman and Hall/CRC. (2022)
13. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems, 28. (2015)
14. Frankle, J., & Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635. (2018)
15. Gale, T., Elsen, E., Hooker, S.: The state of sparsity in deep neural networks. arXiv preprint arXiv:1902.09574. (2019)
16. Makenali, S., Rokh, B., Azarpeyvand, A.: Integrating Pruning with Quantization for Efficient Deep Neural Networks Compression. arXiv preprint arXiv:2509.04244. (2025)