Accuracy Performance Analysis of Quantized DNN Models using Approximate 4-2 Compressor Based Multipliers

Seokhyeon Lee, Jeonggeun Kim, and Yongtae Kim†
School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea
{*dltjr0703, jeonggeun.kim, yongtae*}*@knu.ac.kr*

***Abstract*—As deep neural networks (DNNs) face increasing computational demands, approximate computing techniques are gaining interest in reducing hardware costs. This paper investigates the use of approximate compressor based 8-bit approximate multipliers to evaluate their impact on quantized DNN performance. The systematic evaluations on various well-known DNN models, such as VGGNet, ResNet, Inception-v3, and DenseNet show that the low-error approximate multipliers characterized by error metrics maintain DNN inference accuracy similar to exact multiplier. In contrast, the high-error designs lead to significant accuracy degradation. Additionally, we observe that the approximation-aware fine-tuning mitigates minor accuracy losses for low-error multipliers but is less effective for high-error designs. The findings highlight the importance of selecting low-error approximate multipliers to balance computational efficiency and DNN accuracy.**

***Index Terms*—approximate multiplier, approximate compressor, deep neural network (DNN)**

I. Introduction

As deep neural networks (DNNs) continue to advance, so does the demand for computational power. Researchers continually endeavor to reduce the costs of inference and training, while maintaining model accuracy. Common techniques include quantization and low-precision floating-point formats ($e.g.$, half-precision). The IEEE 754 floating-point standard itself has seen modifications to improve efficiency, with formats like brain floating point ($i.e.$, `bfloat16`) and TensorFloat-32 ($i.e.$, `TF32`) designed to decrease memory usage and computation time [1], [2]. Beyond floating-point optimizations, the numerous addition and multiplication operations in DNNs presents opportunities for approximation thanks to their inherent error resiliency [3]–[6]. DNNs, particularly in inference, are known to be robust to minor computational errors, as their architecture allows them to maintain acceptable accuracy levels even when exposed to random bit errors on DNN weights. This tolerance facilitates the implementation of approximate computing, thereby enhancing computational efficiency. Building on this, the quantization process in DNNs creates opportunities to employ approximate integer arithmetic for multiply-and-accumulate (MAC) operations, moving beyond traditional floating-point approaches. This shift can lead to significant improvements in hardware efficiency, including reductions in area, delay, and power consumption. One promising method involves leveraging approximate multipliers, with approximate compressors offering a viable solution for their implementation. Specifically, utilizing approximate 4-2 compressors during the partial product reduction, which involves compressors and adders, can significantly reduce hardware resources while maintaining acceptable accuracy performance.

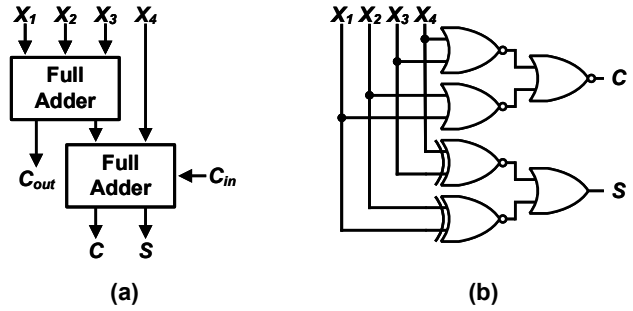

Fig. 1. (a) Exact 4-2 compressor and (b) approximate 4-2 compressor [7].

While prior works have demonstrated the effectiveness of these approximate multipliers in specific applications, such as digital image processing, their broader impact on DNN performance remains insufficiently explored. In particular, the effects of 4-2 compressor-based approximate multipliers on both the computational characteristics and accuracy of DNNs have not been thoroughly analyzed. To shed light on this aspect, this paper evaluates the impact of compressor-based 8-bit approximate multipliers on the inference accuracy of quantized DNNs. We analyze the correlation between multiplier error characteristics and DNN performance, highlighting the importance of low-error designs, and explore the benefits of approximation-aware fine-tuning across various architectures.

II. Approximate Multipliers

Multiplication within digital circuits comprises three primary phases: partial product generation (PPG), partial product reduction (PPR), and final addition. While the PPG involves generating partial products through AND gates and the final addition sums the reduced partial products, the PPR phase, which has numerous 4-2 compressors, half-adders (HAs) and full-adders (FAs) to reduce the partial products, is the most computationally intensive and hardware-demanding. Hence, the complexity of the PPR phase makes it a prime target for approximation techniques aimed at improving efficiency. As Illustrated in Fig. 1(a), an exact 4-2 compressor consists of two cascaded FAs, resulting in a higher count of XOR gates. Furthermore, exact 4-2 compressors has five inputs (X_1, X_2, X_3, X_4, C_{in}) and generate three outputs (C, C_{out}, S). Approximate 4-2 compressors, on the other hand, simplify

TABLE I
TRUTH TABLE FOR VARIOUS APPROXIMATE COMPRESSORS.

Inputs	Prob.	Momeni		Yang		Akbari		Ha		Ahma		Zhang	
$X_{4:1}$		CS	ED	CS	ED	CS	ED	CS	ED	CS	ED	CS	ED
0000	81/256	01	+1	00	0	00	0	00	0	00	0	00	0
0001	27/256	01	0	01	0	01	0	01	0	01	0	01	0
0010	27/256	01	0	01	0	01	0	01	0	01	0	01	0
0011	9/256	01	-1	10	0	00	-2	10	0	01	-1	01	-1
0100	27/256	01	0	01	0	01	0	01	0	01	0	10	+1
0101	9/256	10	0	10	0	01	-1	10	0	11	+1	10	0
0110	9/256	10	0	10	0	01	-1	10	0	11	+1	10	0
0111	3/256	11	0	11	0	01	-2	11	0	11	0	10	-1
1000	27/256	01	0	01	0	11	+2	01	0	01	0	10	+1
1001	9/256	10	0	10	0	11	+1	10	0	11	+1	10	0
1010	9/256	10	0	10	0	11	+1	10	0	11	+1	10	0
1011	3/256	11	0	11	0	11	0	11	0	11	0	10	-1
1100	9/256	01	-1	10	0	10	0	01	-1	01	-1	10	0
1101	3/256	11	0	11	0	11	0	10	-1	11	0	11	0
1110	3/256	11	0	11	0	11	0	10	-1	11	0	11	0
1111	1/256	11	-1	11	-1	10	-2	11	-1	11	-1	11	-1

this structure by reducing the number of inputs to four of X_1, X_2, X_3, and X_4 and outputs to two of C and S as shown in Fig. 1(b). This reduction complexity translates directly to lower hardware resource utilization.

Table I presents the truth table for six approximate 4-2 compressor designs proposed by Momeni [7], Yang [8], Akbari [9], Ha [10], Ahma [11], Zhang [12]. The table highlights input cases where errors occur and their magnitudes through error distance (ED) defined as the difference between the exact and approximate compressor outputs. A higher number of non-zero ED values across all input patterns indicates greater overall approximation errors, especially when input patterns with a high probability of occurrence have non-zero ED. These conditions can reduce computational accuracy but improve hardware efficiency by lowering area, power, and delay. Therefore, understanding these trade-offs is critical for selecting compressors that balance hardware efficiency and model accuracy in deep learning applications.

III. EXPERIMENTAL RESULTS

In this section, we systematically analyze the inference accuracy of various DNNs using various approximate 4-2 compressor based multipliers and also evaluate the impact of pre-training and re-training approaches.

First, we evaluate the error characteristics of the approximate multipliers in terms of error rate (ER), normalized mean error distance (NMED), mean relative error distance (MRED), and number of effective bits (NoEB), which are widely used metric to assess the performance of approximate arithmetic. Briefly, the ER measures the percentage of input combinations for which the approximate multiplier produces an incorrect output compared to the accurate multiplier. The NMED provides a relative measure of the average error magnitude with respect to the maximum possible output value. The MRED quantifies the average relative error introduced by the approximate multiplier compared to the accurate multiplier. The NoEB indicates the number of output bits that are potentially free from error in the approximate multiplier. Table II summarizes

TABLE II
ERROR METRICS OF VARIOUS 8×8 APPROXIMATE MULTIPLIERS.

Design	ER (%)	NMED (10^{-3})	MRED (10^{-2})	NoEB
Momeni	93.38	1.634	9.033	8.92
Yang	3.59	0.048	0.024	11.60
Akbari	84.34	2.928	4.823	8.01
Ha	27.85	0.453	0.368	9.85
Ahma	77.40	1.469	1.698	8.94
Zhang	92.45	1.953	4.412	8.70

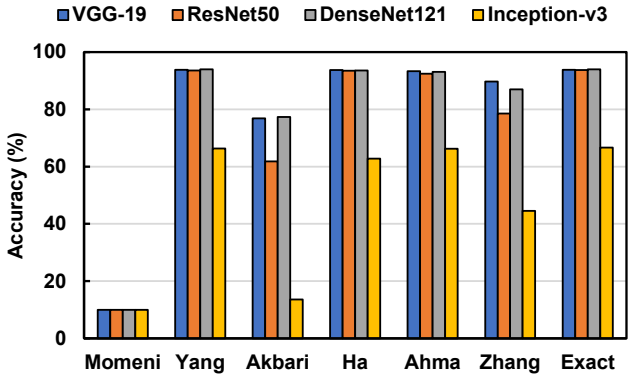

Fig. 2. DNN inference accuracies under various approximate multipliers.

the error metrics for various approximate multipliers. The Momeni, Akbari, Zhang, and Ahma show high ER, NMED, and MRED values, indicating larger approximation errors, while the Ha and Yang achieves relatively better error characteristics and with the higher NoEBs, reflecting better output reliability.

To examine the impact of approximate compressor based 8-bit multipliers on quantized DNN inference accuracy, we employed the AdaPT framework that allows emulations approximate DNN hardware accelerators using approximate multipliers [13]. Specifically, this framework facilitates direct replacement of exact multipliers with approximate versions, leveraging lookup tables (LUTs) for efficient DNN computations. We considered four well-known pre-trained DNN models, which are VGG-19, ResNet50, Inception-v3, and DenseNet. All the models were adapted for 8-bit operations using pre-trained weights on CIFAR-10 dataset. We assessed initial accuracy and then applied approximation-aware fine-tuning ($i.e.$, re-training) to quantify the potential for mitigating accuracy loss. This approach enabled us to characterize the performance of these multipliers across diverse DNN architectures.

Fig. 2 illustrates the inference accuracy of the four different DNN models when using the six approximate multipliers. As expected, the approximate multipliers with lower NMED and MRED values, such as Yang, Ha, and Amha, directly corresponded to higher inference accuracies across all tested DNN models. These multipliers exhibit near-optimal performance across all architectures, achieving accuracies close to those obtained with exact multiplier. This indicates that their low NMED and MRED values introduce minimal error,

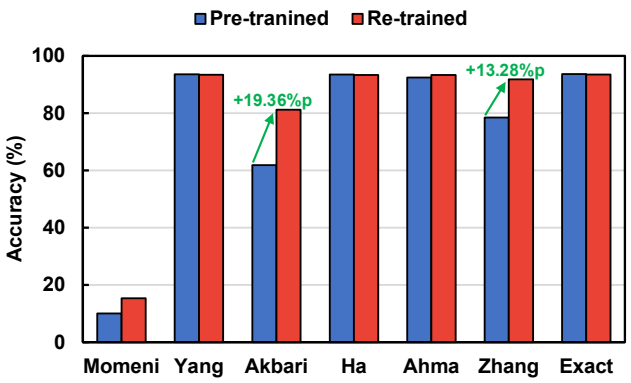

Fig. 3. Inference accuracy of pre-trained and re-trained ResNet model.

resulting in a limited negative impact on the DNN's ability to generalize. On the other hand, the Akbari and Zhang have relatively higher MRED values, leading to accuracy drops of 76.82% and 89.70% respectively, compared to the best-performing multipliers. The Akbari, in particular, shows significantly poorer performance on more complex models such as Inception-v3, with an accuracy reduction of 53.01%p compared to using exact multiplier. Unfortunately, the Momeni performs worst overall, due to its high ER and MRED caused from the inherent vulnerability of its design to all-zero input. The logic of the Momeni, as depicted in Fig. 1(b), leads to frequent inaccuracies, particularly given the high probability ($i.e.$, 81/256) of all-zero input, as detailed in Table I.

Fig. 3 shows the accuracy performance between pre-trained and re-trained ResNet model for each approximate multiplier. We re-trained the ResNet model using the approximation-aware fine-tuning to adapt the pre-trained initial weights for each specific approximate multiplier. As expected, the re-trained ResNet model mitigates minor accuracy losses and improves the overall inference accuracy, especially for approximate multipliers with relatively better error characteristics, such as the Ahma, Ha, and Yang. For the Akbari and Zhang, despite its high NMED, the re-training significantly improves inference accuracy, increasing it 19.36%p and 13.28%p, respectively. This suggests that while there are some errors on computations, the re-training can effectively mitigate the errors. However, for the Momeni, the re-training provides minimal improvement. Its high ER and MRED overwhelm the re-training process. The high frequency of error, due to the all-zero input vulnerability, hinders the effective adaptation. This emphasizes the importance of considering error metrics, when selecting approximate multipliers and applying re-training. This results indicate that the approximation-aware fine-tuning is effective for the approximate multipliers with relatively low MRED values. While the re-training can partially compensate for the multipliers with higher NMED values, such as the Akbari and Zhang, those with high MRED values, as can be seen with the Momeni, pose significant challenges. Therefore, the evaluation of error metrics is crucial for selecting multipliers and effective application of approximation-aware fine-tuning.

IV. Conclusion

In this paper, we investigated the impact of various approximate 4-2 compressor-based 8-bit multipliers on the inference accuracy of various quantized DNNs. We observed a strong correlation between the error characteristics of the multipliers and the resulting DNN inference accuracy. The approximate multipliers with relatively low error metric values, such as the Ahma, Ha, and Yang, maintained accuracy close to that achieved with the exact multiplier across various DNN models. On the other hand, the multipliers such as the Akbari and Momeni, which have higher error metric values, led to substantial accuracy degradation. Particularly, the approximation-aware re-training proved effective in mitigating minor accuracy losses introduced by low-error multipliers. However, this technique offered limited benefit for high-error multipliers, highlighting the inherent limitations of correcting frequently occurring errors during training.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00414964).

References

[1] M. Kwak, J. Kim, and Y. Kim, "Torchaxf: Enabling rapid simulation of approximate dnn models using gpu-based floating-point computing framework," in *Proc. IEEE Int. Symp. Modeling Anal. Simulation Comput. Telecommun. Syst. (MASCOTS)*, pp. 1–8, 2023.

[2] M. Kwak, J. Kim, and Y. Kim, "A comprehensive exploration of approximate dnn models with a novel floating-point simulation framework," *Perform. Eval.*, p. 102423, 2024.

[3] S. Hwang, H. Seok, and Y. Kim, "Design of an approximate 4-2 compressor with error recovery for efficient approximate multiplication," *J. Semicond. Technol. Sci.*, vol. 24, no. 4, pp. 305–315, 2024.

[4] H. Seo and Y. Kim, "A low latency approximate adder design based on dual sub-adders with error recovery," *IEEE Trans. Emerg. Top. Comput.*, vol. 11, no. 3, pp. 811–816, 2023.

[5] H. Seo, H. Seok, J. Lee, Y. Han, and Y. Kim, "Design of an approximate adder based on modified full adder and nonzero truncation for machine learning," *J. Semicond. Technol. Sci.*, vol. 23, no. 2, pp. 138–148, 2023.

[6] H. Seok, H. Seo, J. Lee, and Y. Kim, "Design optimization of a 4-2 compressor for low-cost approximate multipliers," *IEIE Trans. Smart Process. & Comput.*, vol. 11, no. 6, pp. 455–461, 2022.

[7] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, 2015.

[8] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error-resilient multiplier design," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS)*, pp. 183–186, 2015.

[9] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1352–1361, 2017.

[10] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," *IEEE Embed. Syst. Lett.*, vol. 10, no. 1, pp. 6–9, 2018.

[11] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, "Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," *AEU Int. J. Electron. Commun.*, vol. 110, p. 152859, 2019.

[12] M. Zhang, S. Nishizawa, and S. Kimura, "Area efficient approximate 4–2 compressor and probability-based error adjustment for approximate multiplier," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 70, no. 5, pp. 1714–1718, 2023.

[13] D. Danopoulos, G. Zervakis, K. Siozios, D. Soudris, and J. Henkel, "Adapt: Fast emulation of approximate dnn accelerators in pytorch," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 42, no. 6, pp. 2074–2078, 2023.