

Towards Input-Distribution-Aware Approximate Multiplier Generation for CNNs

Alessandro Buccolini

Faculty of Informatics
Università della Svizzera italiana
Lugano, Switzerland
alessandro.buccolini@usi.ch

Marco Biasion

Faculty of Informatics
Università della Svizzera italiana
Lugano, Switzerland
marco.biasion@usi.ch

Rodrigo Otoni

Faculty of Science and Engineering
University of Groningen
Groningen, Netherlands
r.b.otoni@rug.nl

George A. Constantinides

Faculty of Engineering
Imperial College London
London, UK
g.constantinides@imperial.ac.uk

Laura Pozzi

Faculty of Informatics
Università della Svizzera italiana
Lugano, Switzerland
laura.pozzi@usi.ch

Abstract—Convolutional Neural Networks (CNNs) are widely used in vision-related tasks and require intensive computation, due to the large number of multiplications in their convolutional layers. Their inherent tolerance to small numerical perturbations makes them well-suited for approximate computing, which can significantly reduce circuit area and energy consumption while having a limited impact on accuracy. We present an approach for generating approximate multipliers tailored to CNN input distributions. By using multiple complementary constraints and integrating them into an SMT-based design framework, our method effectively explores the approximation design space, producing multipliers that achieve an effective accuracy–efficiency tradeoff. Compared to five state-of-the-art CNN-oriented design techniques, our approach reduces PDA (Power-Delay-Area product) by an average of 17.45% (up to 25.73%) at equivalent accuracy.

Index Terms—neural networks, approximate arithmetic, circuit design, energy saving, SMT solving

I. INTRODUCTION

Convolutional Neural Networks (CNNs) are widely used in vision tasks due to their ability to efficiently process grid-structured data. Their convolutional layers rely heavily on multiply-accumulate operations, making inference computationally expensive and dominated by multiplications.

CNNs naturally tolerate small numerical perturbations thanks to learning and redundancy across layers. This property makes them suitable for approximate computing, where using approximate operators can yield significant area and energy savings with limited accuracy degradation.

In this work, we introduce an approach for generating approximate multipliers tailored to CNN workloads. We exploit the characteristic input distribution of multipliers in CNNs to guide the approximation process, using multiple complementary constraints by partitioning the input space into regions to be approximated more or less aggressively. We then integrate our

method into a framework for approximate circuit generation [1] based on Satisfiability Modulo Theories (SMT) solving.

II. RELATED WORK

Approximate multipliers for neural networks have been explored along several directions. EvoApprox8b [2] provided a library of approximate arithmetic units and benchmarks, while Mrazek et al. [3] proposed for approximate multipliers to be co-designed with neural workloads. Truncation- and rounding-based designs were advanced in TOSAM [4], and efficiency improvements via segmentation and reduction trees were proposed in ARTS [5]. In PAM [6] an automated approach based on partial-product compression was presented. Adaptable architectures with configurable accuracy were introduced in AMCAL [7], while in Li et al. [8] CNN operand distribution and polarity were exploited. Building on this prior work, our SMT-based approach constrains multiple error metrics while leveraging CNN operand distribution, achieving superior accuracy–area tradeoffs compared with designs obtained by five state-of-the-art approaches [2] [4] [5] [6] [8].

III. MOTIVATION

CNN workloads exhibit characteristics that can be exploited when designing approximate multipliers. As shown in Fig. 1 (left), the input pairs of CNN multipliers follow a highly skewed probability distribution, due to ReLU [9] activation and Gaussian weight initialisation. This non-uniformity can significantly influence the performance of approximate multipliers within the network.

To study this, we generate 1000 approximate 8-bit multipliers by injecting random errors, within a limited range, into a 256×256 multiplication table, and compute their mean absolute error:

$$\text{MeanAE} = \sum p(x,y) \text{AE}(x,y),$$

with $p(x,y)$ being the input pair probability, and $\text{AE}(x,y)$ the absolute error between approximate and exact outputs for that

This work was supported by the Swiss National Science Foundation (grants 2000-1-240058 and 10006773) and Hasler Foundation (grant 2025-03-18-473). This paper has been published in the proceedings of DATE'26 and is available at <https://doi.org/10.23919/DATE69613.2026.11539581>.

pair. Each multiplier is then tested in a pre-trained ResNet-8 [10] on CIFAR-10 [11], and their accuracy is measured.

We analyse the accuracy of each multiplier versus MeanAE under the uniform distribution, and versus MeanAE under the CNN distribution, and show results in Fig. 2. As can be noticed, accuracy is far more correlated with MeanAE under the CNN distribution (right), then under uniform (left), particularly in the high-accuracy regime. This observation motivates a multi-constrained methodology that uses SMT solving to enforce bounds on MeanAE under a *specific* input distribution.

IV. APPROXIMATE MULTIPLIER GENERATION

We introduce a practical strategy for generating approximate multipliers that exploits the statistical properties of CNN inputs. The input space is partitioned into i zones, each assigned a distinct absolute error threshold, AET_i . One extreme would be $N = (2^b)^2$ single-element zones, with b the multiplier bitwidth; this, however, would result in a number of constraints which is exponential in b and which might present scalability issues. Instead, we adopt a **parametric partitioning** that captures key characteristics of CNN operand distributions while keeping the constraint set compact. A function groups input pairs into *generalised zones* and assigns each zone an error budget proportional to its statistical relevance:

$$AE(x, y) \leq \left\lceil \frac{|x - x_{\text{mid}}| \cdot \alpha + y}{\beta} \right\rceil \cdot AET \quad (1)$$

where x and y denote activations and weights, $x_{\text{mid}} = 127$ is the midpoint of the 8-bit unsigned range, and parameters α and β control respectively the slope and the granularity of the scaling factor. Fig. 1 (right) visualizes the partition of the input space into zones obtained by applying function (1).

Our formulation achieves a **balanced compromise** between accuracy and tractability, with two key benefits. (i) The constraint system is no longer exponential in the bit-width; instead, it is expressed through a compact set of piecewise-linear constraints controlled by a handful of parameters. (ii) The constraints naturally adapt to the statistical structure of CNN operands – densely concentrated near small values with long-tailed distributions – assigning tighter bounds to frequently occurring pairs and looser bounds elsewhere.

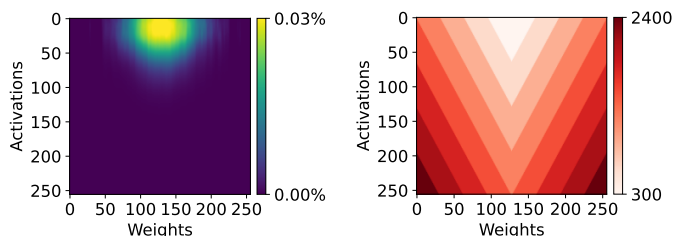


Fig. 1: (left) Distribution of input pairs in a quantized CNN. (right) Division of the input space into zones from the constraint function in (1), for $\alpha = 2$, $\beta = 64$, $x_{\text{mid}} = 127$, $AET = 300$.

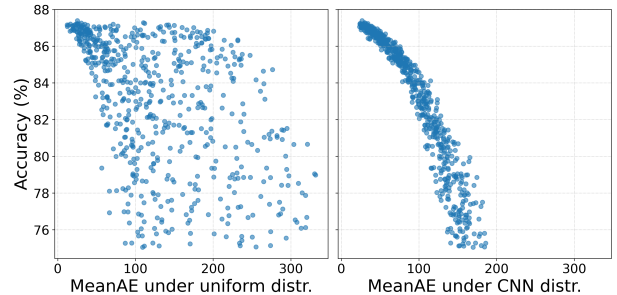


Fig. 2: Correlation between Accuracy and MeanAE under two different distributions

We integrate this parametric constraint model within SubXPAT [1], an SMT-based framework for the design of approximate circuits. The impact of the proposed methodology on circuit efficiency is assessed in the next section.

V. EVALUATION

We evaluate our methodology, and five state-of-the-art approaches [2] [4] [5] [6] [8], on the ResNet-20 architecture, trained and tested on CIFAR-10 [11]. For our methodology as well as for all state-of-the-art methods, the network is first trained in full precision; then, the model is quantised and briefly fine-tuned. Finally, approximate multipliers are integrated into the network as Look-Up Tables (LUTs) encoding the approximate outputs for each input pair. After insertion, the model is retrained for three epochs using the Straight-Through Estimator (STE) to enable gradient propagation through the LUT-based multipliers. The final accuracy is then measured and reported.

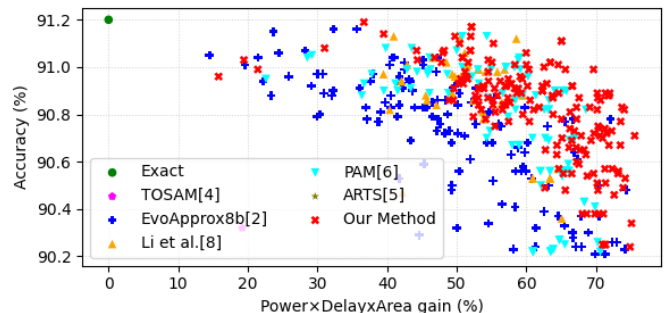


Fig. 3: Accuracy–PDA trade-off for our method compared to state-of-the-art approaches [2] [4] [5] [6] [8].

Fig. 3 illustrates results for the 806 designs tested, and shows how our approximate multipliers consistently achieve superior PDA gains while preserving accuracy compared to the other approaches. Specifically, they achieve an average relative PDA improvement of 17.45% (up to 25.73%) at comparable accuracy levels.

These results confirm that our method is effective in designing approximate multipliers for CNN workloads. Future work will concentrate on a wider range of constraint functions, and of alternative CNN models.

REFERENCES

- [1] M. Rezaalipour, M. Biasion, F. Costa, C. Tirelli, L. Ferretti, R. Otoni, G. A. Constantinides, and L. Pozzi, "Approximate Logic Synthesis via Iterative SMT-based Subcircuit Rewriting," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–14, 2026.
- [2] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "EvoApprox8b: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2017, pp. 258–261.
- [3] V. Mrazek, S. S. Sarwar, L. Sekanina, Z. Vasicek, and K. Roy, "Design of Power-Efficient Approximate Multipliers for Approximate Artificial Neural Networks," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 2016, pp. 1–7.
- [4] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "TOSAM: An Energy-Efficient Truncation- and Rounding-Based Scalable Approximate Multiplier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 5, pp. 1161–1173, 2019.
- [5] M. S. S. Kelayeh, S. Divsalar, S. Vahdat, and N. TaheriNejad, "ARTS: An Approximate Reduced Tree and Segmentation-based Multiplier," *Future Generation Computer Systems*, vol. 175, pp. 1–12, 2026.
- [6] K. Li, Y. Dai, Z. Li, and L. Wang, "Permutation-Based Approximate Multiplier with High Accuracy," in *Proceedings of the IEEE International Conference on ASIC*, 2023, pp. 1–4.
- [7] R. Zendegani and S. Safari, "AMCAL: Approximate Multiplier With the Configurable Accuracy Levels for Image Processing and Convolutional Neural Network," *IEEE Access*, vol. 12, pp. 94 135–94 151, 2024.
- [8] Z. Li, S. Zheng, J. Zhang, Y. Lu, J. Gao, J. Tao, and L. Wang, "Adaptable Approximate Multiplier Design Based on Input Distribution and Polarity," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 12, pp. 1813–1826, 2022.
- [9] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *Computing Research Repository (CoRR)*, pp. 1–7, Aug. 2018.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [11] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," pp. 1–60, 2009.