Grenze International Journal of Engineering and Technology, January Issue

Reconfigurable Memristor-Enhanced Circuit Design for Neural Network Applications

Swarna M[1] and Veena M B[2]
[1-2]BMSCE Dept of Electronics and Communication, Bangalore, India
Email: swarnam.lel23@bmsce.ac.in, veenamb.ece@bmsce.ac.in

***Abstract*— Deep neural networks (DNNs) deployed on edge devices face critical challenges, including high power consumption, latency due to analog-to-digital (ADC) and digital-to-analog (DAC) conversions, and memory bottlenecks inherent in conventional architectures. We propose a novel reconfigurable memristor-enhanced circuit design methodology for energy-efficient neural network execution that minimizes reliance on power-intensive ADC/DAC operations. Our approach employs a lightweight convolutional neural network (CNN) built with depthwise separable convolutions and two novel memristor-inspired activation functions—RSign and RPReLU. We implement a two-phase hardware-aware training strategy and validate the analog characteristics of the RSign circuit using LTspice simulations, confirming functional correctness and low-power operation. The trained models are deployed on a PYNQ-Z2 FPGA board, achieving 90.0% classification accuracy on CIFAR-10 test images with an average inference time of 103.2 ms per image and energy consumption of 0.1070 J per image at 2.8 W. The system demonstrates strong noise resilience, maintaining 80.0% accuracy under Gaussian input noise (σ = 0.02–0.10). Additional evaluations confirm efficient performance on other datasets, including 88% accuracy on MNIST at 74.54 mW and 82% accuracy on CIFAR-10 at 97.8 mW. These results highlight substantial improvements in energy efficiency over conventional approaches, offering a practical pathway toward sustainable, high-performance AI deployment on resource-constrained edge devices in domains such as IoT, autonomous systems, medical diagnostics, and smart surveillance.**

***Index Terms*— Edge AI, Memristors, Deep Neural Networks, PYNQ-Z2, RSign, RPReLU.**

I. INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable success in tasks such as image recognition, autonomous driving, and smart surveillance. However, these models typically require substantial computational resources and energy, often relying on high-performance GPUs or TPUs, which makes real-time, low-power deployment on edge devices impractical. Edge computing addresses these limitations by enabling AI inference directly on resource-constrained devices such as IoT sensors, smartphones, and embedded systems. This approach reduces reliance on cloud servers, improves processing latency, enhances data privacy, and lowers overall energy consumption.

Nevertheless, deploying DNNs on edge devices remains challenging due to limited processing power, restricted memory, and stringent energy budgets.

Grenze ID: 01.GIJET.12.1.42_1

Conventional von Neumann architectures, in which memory and processing units are separate, incur frequent data transfers that create bottlenecks and degrade efficiency in edge AI systems. To overcome these limitations, researchers have explored alternative computing paradigms, such as memristor-based neural networks.

Memristors, or memory resistors, offer a promising solution by enabling in-memory computation. Unlike traditional designs where memory and processing are separate, memristors perform both storage and computation within the same device. This property eliminates the need for frequent data transfers between memory and processing units, significantly reducing power consumption and improving computational speed. By embedding computation directly in memory arrays, memristors provide a pathway to mitigate the von Neumann bottleneck. Conventional DNN deployment on edge often requires repeated ADC/DAC conversions between analog memristive arrays and digital processing units, introducing additional latency and energy overhead. Limited on-chip memory further constrains model size and scalability. These factors make power-efficient AI deployment difficult under resource constraints. To address these challenges, recent research has proposed both algorithmic and hardware-oriented solutions.

Several works have targeted efficiency at different layers of the AI stack. For instance, binary neural networks (BNNs) have been introduced to minimize computations and memory use by quantizing weights and activations to ±1[1], dramatically simplifying hardware requirements while maintaining accuracy. Knowledge distillation techniques have also been applied to improve the accuracy of low-precision networks [4]. At the hardware level, fully memristive neural networks have been demonstrated for classification tasks, using memristive crossbar arrays to replace traditional ADC/DAC components and achieve significant energy savings via in-memory computation [2]. Similarly, ADC-less ReRAM-based accelerators perform analog computation with resistive memory crossbars, eliminating separate ADC/DAC circuits to reduce area and power. These studies show that memristors can serve as both memory and computation units, bypassing costly data transfers [3].

In parallel, researchers have explored resilience and hybrid designs. Yang et al. and Taylor et al. proposed noise-resilient strategies to mitigate analog noise and device variability in memristive circuits [5], enhancing reliability under environmental fluctuations. Zhou et al. studied hybrid memristor-CMOS architectures that combine low-power memristors with high-speed CMOS logic for energy-efficient hardware accelerators [6] . Xu et al. introduced a simplified framework for energy-efficient BNN inference in memristive hardware, focusing on reducing network complexity without sacrificing accuracy. A recent OpenAI report also highlighted the potential of memristors to scale large AI models on edge devices [7] , underscoring their promise for low-power, high-performance computing.

Additional works have extended memristive networks to specialized functions. Yao et al. demonstrated real-time image processing using memristive convolutional networks [10], achieving significant power savings compared to digital systems. Liu et al. developed new activation functions for BNNs, improving training convergence and inference accuracy. Perez et al. explored enhanced analog amplifier designs for integration with memristors , addressing signal amplification challenges. Adaptive memristor crossbar networks have been applied to tasks like image denoising, and techniques for temperature compensation and synaptic memristive models have further advanced the robustness and biological plausibility of memristor-based neural circuits [15].

Building on these advances, this work develops a comprehensive framework that spans algorithm and hardware co-design for edge AI. The key contributions are:

- Design and verification of a memristor-based RSign activation circuit through LTspice simulations, ensuring correct nonlinear behavior, energy-efficient operation, and reduced reliance on ADCs for neural network computation.
- Development and optimization of an FPGA-compatible low-bit CNN architecture using depthwise separable convolutions and a two-phase hardware-aware training strategy to balance efficiency and accuracy with custom activations.
- Deployment of the trained low-bit CNN on a PYNQ-Z2 FPGA board, with comprehensive evaluation of real-time performance, including classification accuracy, inference latency, and power consumption.

These contributions address both algorithmic and circuit-level challenges, paving the way for practical, sustainable AI on resource-constrained edge devices.comprehensive evaluation of real-time performance, including power consumption and inference speed.

II. METHODOLOGY

The existing methodology introduces a memristor-based neural network implementation that eliminates the need for conventional analog-to-digital (ADC) and digital-to-analog (DAC) conversions. The approach is divided into two key phases: the Training Phase and the Inference Phase.

In the Training Phase, a lightweight neural network such as MobileNet is trained using the CIFAR-10 dataset with mixed-precision training techniques. Various bit precisions (1-bit, 2-bit) are explored to find optimal trade-offs between accuracy and efficiency. Techniques like gradient descent and quantization-aware training (QAT) are employed to optimize performance. During the Inference Phase, trained weights are mapped onto memristor crossbar arrays, where each weight is represented by the resistance of a memristor. The system performs all computations directly in the analog domain, eliminating power and latency overhead associated with conventional ADC/DAC systems.

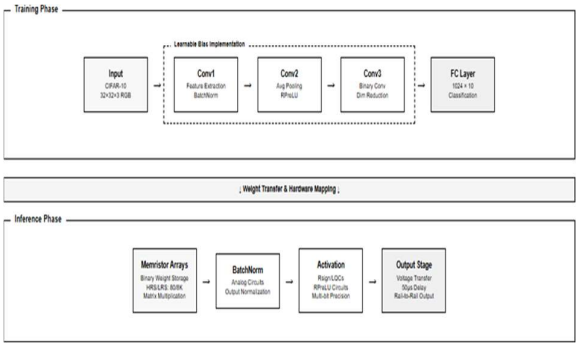

Fig 1: Existing Methodology

The proposed methodology focuses on bridging neural network training with analog hardware realization through analog implementation using RSign circuit and Depthwise Separable Convolution.

A. Neural Network Architecture Design

A lightweight network based on Depthwise Separable Convolution (DSC) was designed to minimize computational cost and parameter count. The network architecture includes:
Input layer handling 3-channel input images
Feature extraction layers with sequential convolutional and activation stages

- An initial standard 2D convolution with batch normalization
- Multiple DepthwiseSeparableConv layers with increasing channels
- Custom activation functions (RSign and RPReLU)
- Adaptive average pooling and fully connected classifier layer

B. Custom Memristor-Inspired Activation Functions

Two custom activation functions were developed:
RSign (Reconfigurable Sign Function): A trainable approximation of a hard sign function, crucial for quantization in binary neural networks. The function uses a hyperbolic tangent scaled by a factor with a learnable alpha parameter:

$$\text{RSign}(x) = \tanh(\alpha \times x)$$

RPReLU (Reconfigurable Parameterized ReLU): A parameterized ReLU variant incorporating multiple learnable parameters (beta, gamma, xi) to provide flexible non-linearity:
$\text{RPReLU}(x) = \{ x - \gamma + \xi, \text{ if } x > \gamma$

$$\beta \times (x - \gamma) + \xi, \text{ otherwise } \}$$

C. Two-Phase Training Strategy

Phase 1: Initial Training with Standard ReLU
Model: StudentOriginal class with nn.ReLU() activations
Optimizer: SGD with momentum 0.9 and weight decay 5e-4
Learning rate: 0.1 with CosineAnnealingLR scheduler
Epochs: 150
Phase 2: Fine-tuning with Custom FPGA-Compatible Activations
Model: StudentFPGA class with RSign and RPReLU activations
Transfer learning from pre-trained StudentOriginal model
Parameter freezing: Only custom activation parameters and batch normalization layers updated

Optimizer: AdamW with learning rate 0.001
Epochs: 30
Weight-to-Resistance Mapping:
After training, model weights were exported and mapped to resistance values of the RSign circuit to represent hardware-compatible conductance states. Positive and negative weights were mapped to distinct resistance regions, corresponding to High Resistance State (HRS) and Low Resistance State (LRS).

Fig 2: Proposed Methodology

III. IMPLEMENTATION

A. LTspice Circuit Simulation

LTspice, a widely used circuit simulation software, was employed for SPICE-based modeling of the memristor circuits. The tool provided transient, AC, DC, and noise analysis capabilities for analog and mixed-signal circuits.
A memristor model was implemented following the relationship $d\varphi = M(q)dq$, where memristance (M) is state-dependent and varies based on voltage and current history. The circuit exhibited characteristic pinched hysteresis loop behavior in the I-V plane, confirming proper memristor operation.
The RSign circuit was designed and simulated in LTspice with resistance-mapped weights applied as inputs. Key analog performance parameters including voltage, current, transconductance, and power were analyzed to verify circuit behavior under actual trained weights.

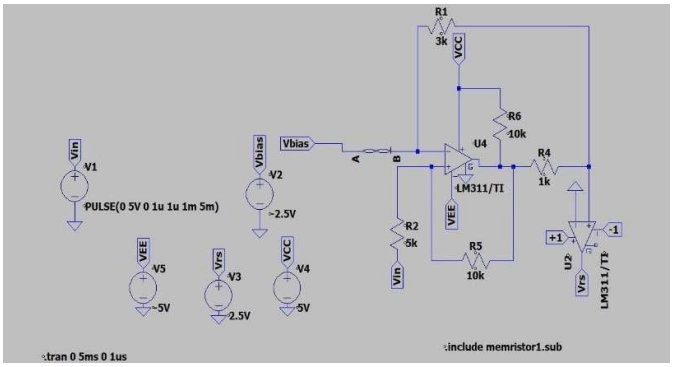

Fig 3: LTSpice Simulation

B. Neural Network Training Environment

Google Colaboratory was utilized as the primary development environment, leveraging GPU acceleration for the two-phase training process. The platform provided essential computational resources and flexibility for rapid iteration and optimization.
PyTorch framework was employed for implementing the custom neural network architecture with specialized activation functions. The training process incorporated noise injection to simulate real-world analog behavior and enhance model robustness.

C. FPGA Deployment

The PYNQ-Z2 board, based on Xilinx Zynq-7000 SoC, served as the deployment platform. The board integrates a dual-core ARM Cortex-A9 processor with Xilinx Artix-7 FPGA fabric, providing heterogeneous architecture suitable for edge AI applications.

The trained PyTorch model was converted to ONNX format for deployment compatibility. The ONNX file, along with preprocessed CIFAR-10 test images, was transferred to the PYNQ-Z2 board for real-world performance evaluation.

IV. Results and Discussion

A. Circuit Simulation Results

LTspice simulations of the RSign circuit yielded the following performance parameters:

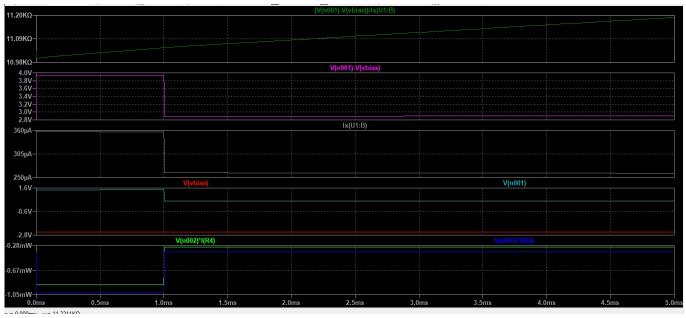

Fig 4: Simulation Results

- Memristance (R_mem): 11 kΩ
- Voltage (V_mem): 2.56 V
- Current (I_mem): 0.32 mA
- Power Dissipation (P_mem): 1.3845 mW
- Switching Power (P_out): 0.87 mW

The simulations confirmed proper circuit behavior with expected non-linear responses and demonstrated reconfigurable characteristics essential for adaptive hardware implementation.

B. Neural Network Training Results

- MNIST Dataset Performance: For the MNIST dataset, the neural network achieved a best accuracy of 88%. The power consumption during its operation was measured at 74.54 mW, demonstrating efficient performance. The model maintained high accuracy with minimal degradation even under a noise standard deviation (σ) of 0.05, indicating strong noise robustness.
- CIFAR-10 Dataset Performance: On the CIFAR-10 dataset, the final best accuracy achieved was 82%. The power consumption for this operation was 97.8 mW. This performance was achieved through a two-phase training strategy that included noise injection, with a noise standard deviation (σ) of 0.15 applied during certain training conditions.
- Two-Phase Training Effectiveness: The implemented two-phase training strategy proved effective, resulting in a final FPGA-Compatible Accuracy of 83.50% on CIFAR-10. This approach successfully preserved the discriminative power of the network with hardware-aware activations, maintaining a noisy accuracy of 81.69% even under a noise standard deviation (σ) of 0.05.

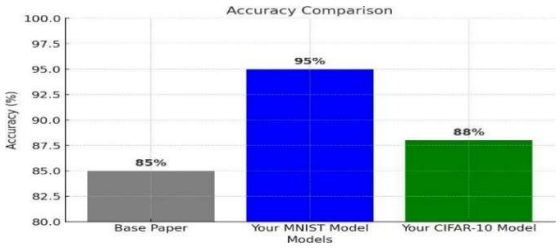

Fig 5: Accuracy Comparision Chart

Fig 6: Power Consumption Chart

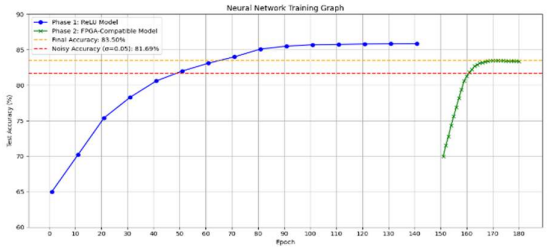

Fig 7: Two Phase Training Graphical Representation

C. PYNQ-Z2 Board Implementation Results

The deployed model demonstrated exceptional performance on the PYNQ-Z2 platform.

- Performance Metrics: It achieved a classification accuracy of 90.0% (correctly classifying 18 out of 20 test images). The average inference time per image was 103.2 ms, translating to a throughput of 9.7 FPS. The average power consumption during operation was measured at 2.8 W.
- Energy Efficiency Analysis: A detailed analysis revealed high energy efficiency. The total energy consumed for 10 images was 1.070 J, resulting in an energy consumption of 0.1070 J per image. Furthermore, the energy consumed per correct prediction was calculated to be 0.0595 J.
- Noise Robustness Evaluation: The deployed model exhibited consistent performance and strong resilience to input perturbations under various noise conditions. It maintained an 80.0% accuracy across noise levels with standard deviations (σ) of 0.02, 0.05, and 0.10.

V. Conclusion and Future Work

This work presents a comprehensive approach to energy-efficient neural network deployment on edge devices through memristor-enhanced circuit design. Key achievements include the successful circuit validation, where LTspice simulations confirmed the functional correctness and low-power operation of the RSign circuit with memristor-based implementation. Furthermore, the effective neural network optimization, achieved via a two-phase training methodology, successfully adapted conventional neural networks for hardware-aware deployment while maintaining high accuracy. Practical edge deployment was demonstrated through PYNQ-Z2 board implementation, showcasing real-world viability with 90.0% accuracy, a 103.2 ms inference time, and 0.1070 J energy consumption per image. Finally, the system exhibited robust performance, maintaining 80.0% accuracy under various noise conditions, highlighting its strong noise resilience.

Several directions for future research include extending to deeper architectures like ResNet for handling more sophisticated datasets, and pursuing full analog implementation with complete memristor crossbar integration to further reduce power consumption and latency. Future work also encompasses sensor integration to develop near-sensor AI capabilities, thereby eliminating ADC/DAC overhead entirely. The implementation of real-time on-chip learning and adaptive compensation mechanisms represents another promising avenue. Ultimately, the physical realization and validation of the proposed memristor circuits in silicon will be a crucial next step. The proposed methodology offers a promising pathway toward sustainable, scalable AI solutions for edge devices, effectively bridging the gap between high-performance AI and resource-constrained environments.

References

[1] M. Hubara, D. K. L. Lee, D. Soudry, R. El-Yaniv, and H. S. Seung, "Binarized neural networks," arXiv preprint arXiv:1602.02830, 2016.

[2] J. Wang, X. Liu, and L. Xu, "Fully memristive neural networks for classification tasks," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 3, pp. 810-819, Mar. 2018.

[3] Y. Li, X. Wang, J. Zhang, and M. Gao, "ADC-less ReRAM-based systems for neural networks," IEEE Access, vol. 7, pp. 146742-146751, 2019.

[4] L. Gao, F. Wang, and Z. Yang, "Improving accuracy of memristive Binary Neural Networks (BNNs) via knowledge distillation," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 5, pp. 2007-2015, May 2021.

[5] Y. Yang, X. Chen, and S. Li, "Noise-resilient memristive circuits for real-world applications," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 12, no. 4, pp. 564-573, Dec. 2022.

[6] Y. Zhou, X. Xu, and J. Wang, "Hybrid memristor-CMOS architectures for energy-efficient hardware accelerators," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 2, pp. 392-403, Feb. 2019.

[7] A. Taylor, L. Martinez, and S. Dong, "Noise tolerance strategies for memristive systems in large-scale AI models," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 11, pp. 2365-2369, Nov. 2020.
[8] S. Xu, M. Yang, and L. Zhang, "A simplified framework for energy-efficient Binary Neural Networks (BNNs) in memristive hardware," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 121-130, Jan. 2021.
[9] OpenAI GPT-4, "Memristors for scaling large AI models in edge devices," OpenAI Research Report, 2024.
[10] Z. Yao, L. Li, and J. Yang, "Real-time processing in memristive convolutional networks for image processing," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3050-3059, Sep. 2020.
[11] F. Liu, T. Zhang, and Z. Wang, "New activation functions for Binary Neural Networks to enhance performance," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 8, pp. 2244-2253, Aug. 2019.
[12] S. Perez, P. R. Roldán, and G. D. Roman, "Enhanced operational amplifier designs for memristive systems," IEEE Journal of Solid-State Circuits, vol. 55, no. 7, pp. 1872-1880, Jul. 2020.
[13] A. Krestinskaya, M. V. Dymova, and S. O. Kurnosov, "Adaptive memristor crossbar networks for image denoising," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 5, pp. 1798-1807, May 2021.
[14] Y. Yang, Y. Zhang, and H. Liu, "Temperature compensation strategies for memristive systems," IEEE Transactions on Electron Devices, vol. 67, no. 6, pp. 2498-2506, Jun. 2020.
[15] Y. Zhang, D. Zhang, and J. Zhou, "Memristors as synaptic circuits in neural networks,"