Erudite Journal of Engineering Technology and Management Sciences
Vol. 3, No. 3, September 2023, pp. 01-04
Journal homepage: http://www.ejetms.com

A comprehensive review on Recent Advances in VLSI Architectures for Deep Learning in Embedded Systems

Shaik Nazmaali *

Assistant Professor, Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering Technology, Tirupati

Corresponding Author Email: nazmashaik119@gmail.com

Received: 12 March 2023
Accepted: 25 August 2023

Keywords:
Deep Learning,
VLSI Architectures,
Embedded Systems,
Hardware Accelerators,
Energy-Efficient Design

ABSTRACT

Deep learning has transformed several industries, including computer vision and natural language processing, as well as autonomous systems and robotics. However, due to restricted processing power and memory restrictions, deploying deep learning models on resource-constrained embedded devices presents various obstacles. This review paper delves into recent advancements in VLSI (Very Large Scale Integration) designs aimed at addressing these problems and enabling fast deep learning inference on embedded devices. The primary focus of this study is on new VLSI designs and approaches for optimizing deep learning execution on embedded systems that have developed in recent years. These include hardware-friendly quantization methods, model compression techniques, and custom hardware accelerators tailored for specific deep learning tasks. They also investigate the use of sparsity and efficient memory management to minimize the memory footprint, allowing deep learning to be performed in resource-constrained environments. The significance of energy-efficient design and low-power solutions for embedded systems is highlighted. Edge AI and IoT are developing phenomena, and VLSI designs are evolving to enable these applications. The goal of this review is to provide a helpful resource for implementing deep learning on embedded systems by demonstrating the most recent breakthroughs in VLSI designs to allow fast and scalable deep learning inference.

1. INTRODUCTION

Deep learning has experienced remarkable success in various domains, ranging from computer vision and natural language processing to autonomous systems and robotics [1]. However, deploying deep learning models on embedded systems, characterized by limited computational power and constrained memory resources, presents a set of formidable challenges. These challenges include the need for efficient hardware architectures capable of executing complex deep learning computations within the constraints of embedded environments [2]. In response to these challenges, recent years have witnessed a surge in research and development of VLSI (Very Large Scale Integration) architectures tailored for deep learning applications in embedded systems. This surge is driven by the increasing demand for intelligent, real-time processing in devices such as edge AI platforms and IoT (Internet of Things) endpoints. The most recent advancements in VLSI architectures designed specifically to address the unique requirements of deep learning on embedded systems delve into the core components of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and examine the hardware resources they demand [3]. Additionally, provide an overview of conventional hardware accelerators, including GPUs and TPUs, and the constraints they encounter when adapted to embedded applications. Primary focus the review is on the innovative solutions that have emerged to tackle these constraints. And it cover topics such as hardware-friendly quantization techniques, model compression methods, and the development of custom hardware accelerators finely tuned for particular deep learning tasks. Also explore strategies that reduce memory usage and leverage sparsity, enabling efficient execution in resource-constrained environments. Efficiency in power consumption is paramount in embedded systems, and the importance of energy-efficient design and low-power strategies in this context. Moreover, how these VLSI architectures align with emerging trends, like edge AI and IoT, and adapt to cater to their unique demands. analysis of the various VLSI-based solutions identify potential avenues for future research and development in this rapidly evolving field serve as a comprehensive resource for researchers and engineers interested in the deployment of deep learning on embedded systems, presenting the latest developments in VLSI architectures to facilitate efficient and scalable deep learning inference [5].

2. Deep Learning Fundamentals

2.1 Overview of Deep Learning Models:

Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have gained widespread popularity due to their ability to

automatically learn and extract features from data. CNNs are particularly effective for image recognition and computer vision tasks, while RNNs excel in sequence data analysis, making them ideal for natural language processing and speech recognition [6].

2.2 Hardware Requirements for Deep Learning:

Deep learning models are highly compute-intensive, requiring massive parallel processing capabilities. GPUs (Graphics Processing Units) have played a pivotal role in accelerating deep learning tasks, thanks to their ability to handle parallel computations efficiently. In recent years, TPUs (Tensor Processing Units) have emerged as specialized hardware designed explicitly for deep learning, offering even greater performance [7].

2.3 Challenges in Deploying Deep Learning on Embedded Systems:

Deploying deep learning on embedded systems presents several challenges. Embedded systems often have limited computational power, memory constraints, and energy consumption concerns. This creates a need for specialized VLSI architectures to optimize deep learning inference in these resource-constrained environments [8].

3. Hardware Accelerators for Deep Learning

3.1 GPUs and TPUs: Traditional Hardware Accelerators:

GPUs (Graphics Processing Units) have been a cornerstone in accelerating deep learning tasks due to their highly parallel architecture, making them well-suited for training and inference tasks. TPUs (Tensor Processing Units) are custom-designed hardware accelerators created by Google, optimized specifically for machine learning workloads, and offer substantial performance gains in deep learning tasks [9].

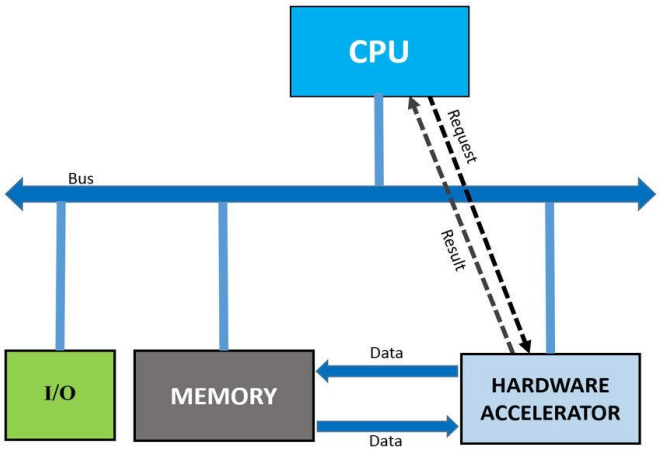

Fig. 1: Hardware Accelerator architecture The CPU 'offloads' its task to the FPGA, which performs the action and returns the result to the CPU in a expedited manner

3.2 Limitations of Traditional Accelerators in Embedded Systems:

While GPUs and TPUs excel in terms of performance, they may not be well-suited for embedded systems. These traditional hardware accelerators tend to be power-hungry, bulky, and often overprovisioned for the requirements of embedded applications. Additionally, they may not efficiently exploit the unique characteristics of embedded environments [10].

4. Hardware-Friendly Quantization Methods

4.1 Fixed-Point and Dynamic Fixed-Point Quantization:

Fixed-point quantization is a technique that reduces the precision of deep learning models by representing weights and activations as fixed-point numbers [11]. Dynamic fixed-point quantization allows for the adaptation of the quantization scheme based on data statistics, offering a balance between precision and efficiency.

4.2 Binary and Ternary Quantization:

Binary quantization represents weights and activations as binary values (1-bit precision), while ternary quantization extends this concept to use three values (-1, 0, 1) to further reduce precision [12]. These methods significantly reduce the memory and computational demands of deep learning models.

4.3 Mixed-Precision Quantization:

Mixed-precision quantization combines different precision levels for different parts of the deep learning model, allowing for flexibility in resource allocation. For example, it may use higher precision for critical layers and lower precision for less critical ones, optimizing the trade-off between accuracy and resource usage [13].

5. Model Compression Techniques

5.1 Pruning and Weight Sharing:

Pruning involves identifying and removing redundant or less important weights and neurons in a deep learning model, reducing its size and computational requirements [14]. Weight sharing goes a step further by reusing shared weights for multiple connections, which results in further compression and efficiency gains.

5.2 Knowledge Distillation:

Knowledge distillation is a process where a smaller model, known as the student, is trained to mimic the behavior of a larger, pre-trained model, known as the teacher. This transfer of knowledge results in a compact model with reduced computational requirements while maintaining performance.

5.3 Compact Architectures (e.g., MobileNet, SqueezeNet):

Compact architectures, exemplified by models like MobileNet and SqueezeNet, are designed from the ground up to be efficient in terms of computational and memory requirements. These models often employ techniques such as depth-wise separable convolutions and aggressive feature map reduction to achieve high performance with reduced parameters [15].

6. Custom Hardware Accelerators

6.1 Design Considerations:

When designing custom hardware accelerators for deep learning, several key considerations come into play. These include the choice of hardware platform, architectural decisions, power efficiency, memory optimization, and the need to tailor the hardware to specific deep learning tasks [16].

6.2 FPGA-based Accelerators:

Field-Programmable Gate Arrays (FPGAs) are versatile hardware platforms that can be reconfigured to accelerate specific deep learning workloads [17]. They offer flexibility

and programmability, making them suitable for a wide range of applications.

6.3 ASIC-based Accelerators:

Application-Specific Integrated Circuits (ASICs) are custom-designed hardware accelerators optimized for specific deep learning tasks. They offer high performance and energy efficiency, but they lack the flexibility of FPGAs [18].

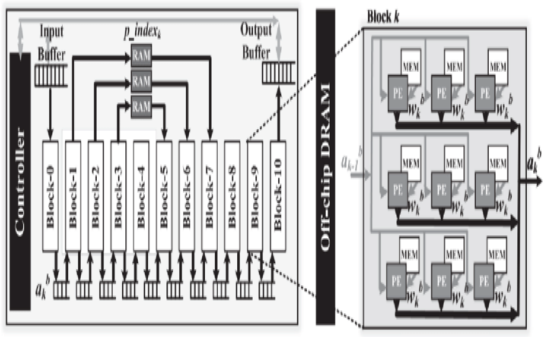

Fig. 2: The architecture of the ASIC-based accelerator.

6.4 Domain-Specific Accelerators (e.g., Edge TPUs):

Domain-specific accelerators, like Google's Edge TPUs, are designed for particular tasks within deep learning and AI, such as image processing and neural network inference. These accelerators are tailored for edge and IoT applications [19].

7. Memory Management and Sparsity

7.1 Efficient Memory Usage:

Efficient memory usage in deep learning involves optimizing how data is stored and accessed during training and inference. Techniques such as data batching, memory pooling, and data reuse can significantly reduce memory consumption [20].

7.2 Leveraging Sparsity in Deep Learning:

Sparsity in deep learning refers to scenarios where a significant portion of weights or activations in a model are zero. Leveraging sparsity can lead to substantial memory and computational savings. Techniques like structured sparsity and pruning are discussed [21].

7.3 Techniques for Reducing Memory Footprint:

Various techniques are available for reducing the memory footprint of deep learning models. These include weight quantization, weight sharing, compression algorithms, and efficient model serialization formats. Each technique has its strengths and limitations [22].

8. Energy-Efficient Design

8.1 Low-Power Strategies for Embedded Systems:

Low-power strategies for embedded systems involve a range of techniques, including voltage scaling, clock gating, and dynamic voltage and frequency scaling (DVFS). These strategies are essential for reducing power consumption while maintaining adequate performance [23].

8.2 Trade-offs Between Performance and Power Consumption:

Designing energy-efficient embedded systems often requires striking a balance between performance and power consumption. Increasing performance can lead to higher power usage, and vice versa. These trade-offs are crucial considerations in the design of embedded systems.

9. VLSI Architectures for Edge AI and IoT

9.1 Tailoring VLSI Solutions to Edge AI:

Edge AI involves deploying artificial intelligence and deep learning models on local devices or "at the edge" rather than relying on cloud-based processing. Customizing VLSI solutions for edge AI means designing hardware to meet the specific requirements of low-latency, real-time, and on-device AI applications. This may involve optimizing for power efficiency, real-time processing, and the unique demands of edge AI use cases such as autonomous vehicles, robotics, and smart surveillance [24].

9.2 IoT Devices and Their Deep Learning Needs:

Internet of Things (IoT) devices encompasses a wide range of applications and are often characterized by their small form factor, limited computational resources, and connectivity to the internet. Many IoT devices benefit from efficient deep learning solutions for tasks like sensor data analysis, anomaly detection, and local decision-making. Custom VLSI architectures play a crucial role in enabling deep learning on IoT devices by addressing the unique challenges of connectivity, power efficiency, and real-time processing in IoT applications.

10. Future Research Directions

Future research directions in VLSI architectures for deep learning in embedded systems encompass a multifaceted approach, including addressing unresolved challenges like balancing performance and power efficiency, exploring opportunities for innovation through novel quantization and sparsity techniques, and staying abreast of emerging trends, such as neuromorphic computing and ethical considerations, to foster continued advancement in the field and enhance the efficiency of embedded systems for deep learning applications.

11. Conclusion

The field of VLSI architectures for deep learning in embedded systems has made significant strides, offering promising solutions to address the challenges of deploying efficient artificial intelligence at the edge, and future endeavors should focus on resolving remaining issues, fostering innovation, and adapting to emerging trends to unlock the full potential of embedded deep learning.

References

1. Wiriyathammabhum, P., Summers-Stay, D., Fermüller, C., & Aloimonos, Y. (2016). Computer vision and natural language processing: recent approaches in multimedia and robotics. ACM Computing Surveys (CSUR), 49(4), 1-44.
2. Ajani, T. S., Imoize, A. L., & Atayero, A. A. (2021). An overview of machine learning within embedded and mobile devices–optimizations and applications. Sensors, 21(13), 4412.
3. Moons, B., Bankman, D., & Verhelst, M. (2019). Embedded deep learning. Embedded Deep Learning.
4. Poletti, F., Poggiali, A., Bertozzi, D., Benini, L., Marchal, P., Loghi, M., & Poncino, M. (2007). Energy-efficient multiprocessor systems-on-chip for embedded computing:

Exploring programming models and their architectural support. IEEE Transactions on Computers, 56(5), 606-621.
5. An, H., Ha, D. S., & Yi, Y. (2020, September). Powering next-generation industry 4.0 by a self-learning and low-power neuromorphic system. In Proceedings of the 7th ACM International Conference on Nanoscale Computing and Communication (pp. 1-6).
6. Ciaburro, G., & Venkateswaran, B. (2017). Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles. Packt Publishing Ltd.
7. Shams, S., Platania, R., Lee, K., & Park, S. J. (2017, June). Evaluation of deep learning frameworks over different HPC architectures. In 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS) (pp. 1389-1396). IEEE.
8. Shuvo, M. M. H., Islam, S. K., Cheng, J., & Morshed, B. I. (2022). Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. Proceedings of the IEEE.
9. Vasilache, N., Zinenko, O., Theodoridis, T., Goyal, P., Devito, Z., Moses, W. S., ... & Cohen, A. (2019). The next 700 accelerated layers: From mathematical expressions of network computation graphs to accelerated gpu kernels, automatically. ACM Transactions on Architecture and Code Optimization (TACO), 16(4), 1-26.
10. Roth, W., Schindler, G., Zöhrer, M., Pfeifenberger, L., Peharz, R., Tschiatschek, S., ... & Ghahramani, Z. (2020). Resource-efficient neural networks for embedded systems. arXiv preprint arXiv:2001.03048.
11. Kim, S., & Kim, H. (2021). Zero-centered fixed-point quantization with iterative retraining for deep convolutional neural network-based object detectors. IEEE Access, 9, 20828-20839.
12. Xu, C., Yao, J., Lin, Z., Ou, W., Cao, Y., Wang, Z., & Zha, H. (2018). Alternating multi-bit quantization for recurrent neural networks. arXiv preprint arXiv:1802.00150.
13. Chang, S. E., Li, Y., Sun, M., Jiang, W., Shi, R., Lin, X., & Wang, Y. (2020). MSP: an FPGA-specific mixed-scheme, multi-precision deep neural network quantization framework. arXiv preprint arXiv:2009.07460.
14. Yeom, S. K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K. R., & Samek, W. (2021). Pruning by explaining: A novel criterion for deep neural network pruning. Pattern Recognition, 115, 107899.
15. Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. Journal of Advanced Research, 35, 33-48.
16. Zaman, K. S., Reaz, M. B. I., Ali, S. H. M., Bakar, A. A. A., & Chowdhury, M. E. H. (2021). Custom hardware architectures for deep learning on portable devices: a review. IEEE Transactions on Neural Networks and Learning Systems.
17. Wang, C., Lou, W., Gong, L., Jin, L., Tan, L., Hu, Y., & Zhou, X. (2017). Reconfigurable hardware accelerators: Opportunities, trends, and challenges. arXiv preprint arXiv:1712.04771.
18. Song, W. J. (2021). Hardware accelerator systems for embedded systems. In Advances in Computers (Vol. 122, pp. 23-49). Elsevier.
19. Véstias, M. (2020). Processing systems for deep learning inference on edge devices. Convergence of Artificial Intelligence and the Internet of Things, 213-240.
20. Mittal, S., & Vaishay, S. (2019). A survey of techniques for optimizing deep learning on GPUs. Journal of Systems Architecture, 99, 101635.
21. Kurtz, M., Kopinsky, J., Gelashvili, R., Matveev, A., Carr, J., Goin, M., ... & Alistarh, D. (2020, November). Inducing and exploiting activation sparsity for fast inference on deep neural networks. In International Conference on Machine Learning (pp. 5533-5543). PMLR.
22. Sudharsan, B., Breslin, J. G., & Ali, M. I. (2020, October). RCE-NN: a five-stage pipeline to execute neural networks (cnns) on resource-constrained iot edge devices. In Proceedings of the 10th International Conference on the Internet of Things (pp. 1-8).
23. Akgün, G., Ali, M., & Göhringer, D. (2021, August). Power-aware computing systems on FPGAs: a survey. In 2021 31st International Conference on Field-Programmable Logic and Applications (FPL) (pp. 45-51). IEEE.
24. Merenda, M., Porcaro, C., & Iero, D. (2020). Edge machine learning for ai-enabled iot devices: A review. Sensors, 20(9), 2533.
25. Asghari, P., Rahmani, A. M., & Javadi, H. H. S. (2019). Internet of Things applications: A systematic review. Computer Networks, 148, 241-261.