Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Hardware Conscious Architecture Search Algorithms for Specialized AI Accelerators

Dr.T. Senthil Prakash[1]*, Dr.R. Udayakumar[2], Dr. Megala Rajendran[3], Dr.H. Shaheen[4], Dr.T. Abirami[5], Siyovush Boboyev[6]

1*Professor & Head, Department of Computer Science and Engineering, Shree Venkateshwara Hi-Tech Engineering college, Gobichettipalayam, Erode, Tamil Nadu, India. E-mail: jtyesp14@gmail.com

[2]Professor & Director, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: rsukumar2007@gmail.com

[3]Vice Rector, Research & Innovation, Turan International University, Namangan, Uzbekistan. E-mail: megala11379@gmail.com, https://orcid.org/0009-0005-9605-5958

[4]Course Leader & Sr. Lecturer, Department of Computing and Engineering, University of West London, Rak branch campus, UAE. E-mail: h.shaheen@uwl.ac.ae, https://orcid.org/0000-0003-3544-5424

[5]Professor, Department of Information Technology, Kongu Engineering College, Erode, India. E-mail: abi.it@kongu.edu, https://orcid.org/0000-0002-7156-751X

[6]Researcher, Samarkand State Medical University, Samarkand, Uzbekistan. E-mail: dalerxonzoda0017@gmail.com, https://orcid.org/0009-0007-2700-9186

*Corresponding author: Email: jtyesp14@gmail.com

Abstract

Deep neural networks have become a growing trend in artificial intelligence, and their energy-efficient, high-performance capabilities require specialized AI accelerators. But traditional neural architecture search approaches focus on prediction issues without considering practical hardware factors like latency, memory bandwidth, and power consumption. Therefore, this study proposes a hardware-in-the-loop neural architecture co-design framework that integrates accelerator-aware feedback directly into the optimization process. This approach integrates the reinforcement learning-driven architecture search with a cycle-accurate hardware simulator in order to optimize both the neural network architecture and hardware execution approaches. The method uses a hardware-aware cost function that relies on the Energy-Delay Product metric to effectively explore energy-efficient and latency-aware neural architectures. Experimentation confirmed that the proposed methodology attained a classification accuracy rate of 94.3%, which is significantly better than the FLOPS-based approach with an accuracy rate of 91.2%. Moreover, its inference time was reduced from 23.4ms to 14.7ms, and its energy usage was decreased from 8.1mJ to 5.4mJ. The optimized methodology further increased power efficiency by 58.1%, minimizing the EDP to 79.4 mJ·ms. Therefore, this research study demonstrates that hardware-aware co-design is an effective solution for designing AI models that are efficient in deployment.

Keywords

Neural Architecture Search, AI Accelerators, Hardware-Aware Optimization, Energy-Delay Product, Edge Computing.

1. Introduction

Today's machine learning systems are built on specialized hardware accelerators like GPUs, TPUs, and neuromorphic hardware to handle deep learning workloads effectively. Nevertheless, with exponential growth in neural networks' architectural complexity, there has been a sharp separation between software design and hardware implementation. The conventional methods for designing neural architectures usually abstract hardware into a simple black box, resulting in designs that have excellent performance in theory but poor latency, inefficiency, and other bottlenecks in their execution in real-world hardware [1][2].

The primary goal of this study is to bring together software design and hardware implementation through the development of hardware-aware search algorithms. These algorithms aim to automate the design of neural networks for throughput and energy efficiency, while preserving their accuracy, by jointly considering the network architecture and the constraints of specialized AI accelerators.

Traditional approaches to hardware-aware search algorithms rely on simple proxies for optimization, including fixed floating-point operations per second or generalized latency lookup tables. These simplifications completely miss the dynamics of hardware execution, like register file contention, movement costs of data across memory hierarchies, and the constraints of specialized tensor cores. Thus, the resulting architectures are still sub-optimal on the edge and in enterprises for AI hardware [8][12].

We believe that the incorporation of cycle-accurate hardware simulation and data-flow information directly into the search algorithm reward function will enable the optimization algorithm to explore the architectural design space more efficiently. Such an approach would lead to optimized neural networks capable of improved performance efficiency from the efficient use of the execution pipelines of the underlying accelerators.

The paper presents a unique approach to neural architecture search that takes into account hardware aspects through hardware-in-the-loop telemetry. The main advantage of such an approach lies in the creation of an end-to-end search algorithm which effectively maps computational graphs to hardware constraints while achieving reduced hardware energy delay product and state-of-the-art model accuracy.

The paper is divided into six main sections. Section 1 starts with an introduction to the problem being addressed, objectives, and motivation for hardware-aware neural architecture search. Section 2 provides a literature review of the related studies on hardware-aware optimization techniques and AI accelerators. Section 3 discusses the methods used, including the proposed hardware-in-the-loop design approach and optimization technique. Section 4 focuses on performance evaluation using accuracy, latency, energy, and EDP as metrics. Section 5 discusses the results obtained. The conclusion of the paper is provided in Section 6.

2. Literature Review

The emergence of recent deep learning developments has made efficient hardware-aware optimization highly essential to enable hardware-accelerator support. Traditional NAS algorithms were largely concerned with achieving maximum predictive accuracy while ignoring the execution constraints imposed by the hardware, including latency, memory bandwidth, and power consumption [1][3]. With deep neural networks becoming more computationally demanding, there was a need to focus on accelerator-efficient architecture and hardware-aware optimization frameworks [5][7].

Several research studies explored efficient architectures for deep neural networks with emphasis on optimal data flow mapping, memory hierarchy design, and minimized off-chip communication cost [1][7][9]. The development of hardware-aware NAS techniques was characterized by the introduction of feedback from the accelerators as part of the architecture search process for joint optimization of network topology and hardware performance [3][6]. Such techniques offered superior performance in terms of latency minimization, energy efficiency, and computation throughput compared with traditional FLOPs optimization methods.

Optimization algorithms research was also involved in the field of hardware-aware architecture exploration. Discrete optimization problems and adaptive computation architectures were solved using optimization algorithms like metaheuristics and search-based approaches, improving the convergence in search and architectural efficiency [2][13]. Furthermore, recent research has been dedicated to co-design approaches for edge-AI and open ISA-based accelerator architectures for efficient and scalable deployments [10].

A systematic review on AI hardware implementation identified the increasing need for hardware-software co-design approaches for practical deployment of AI [11][14]. Furthermore, advances in in-memory computing and micro-AI accelerators demonstrated the effectiveness of jointly optimizing memory access patterns, data reuse, and computational parallelism [4][12]. In general, the current literature shows that hardware-oriented architecture search algorithms offer a viable approach to creating power-efficient, high-performance, and deployment-friendly AI accelerator systems.

3. Methods

Methodology Framework

The methodology forms a hardware-in-the-loop optimization pipeline to connect the neural network topology with the physical limitations of dedicated AI accelerators. The framework works in an iterative manner that involves repeatedly generating macro-architectures, assessing hardware mapping, and updating the algorithm based on the feedback. The search space is dynamically limited to the boundary of the silicon to execute the algorithm, including the size of the static random-access memory (SRAM) and the global memory bandwidth.

Co-Design Search Space

The search space is parameterized to simultaneously optimize macro-architectural hyperparameters and hardware-specific execution configurations. For the neural network, the algorithm searches across variable layer depths, kernel dimensions, and channel widths. Concurrently, for the target AI accelerator, the search space includes loop-tiling factors, data-flow mapping strategies (such as weight-stationary versus output-stationary configurations), and memory allocation policies. This joint formulation ensures that the discovered network topology natively aligns with the hardware's parallel processing capabilities.

Hardware-Conscious Cost Function

To direct the search algorithm to the most efficient hardware, we propose a multi-objective reward function that takes into account both the accuracy of the model and the overhead of its physical execution. Instead of the usual approach of relying on a proxy, floating-point operations (FLOPs), the cost of the hardware is explicitly specified by the Energy-Delay Product (EDP), which works against architectures that lead to excessive latency or power consumption because of memory bottlenecks. A network configuration α is optimized such that the overall optimization objective function $J(\alpha)$ is:

$$J(\alpha) = Accuracy(\alpha) \times \left[\frac{EDP_{target}}{EDP(\alpha)}\right]^{\beta} \quad (1)$$

$EDP(\alpha)$ is defined in equation (1) as $Latency(\alpha) \times Energy(\alpha)$, EDP_{target}, with the right-hand side defining a user design constraint in the deployment setting, and the scaling exponent β being a user parameter that dictates the intensity of the trade-off between predictive performance and hardware efficiency.

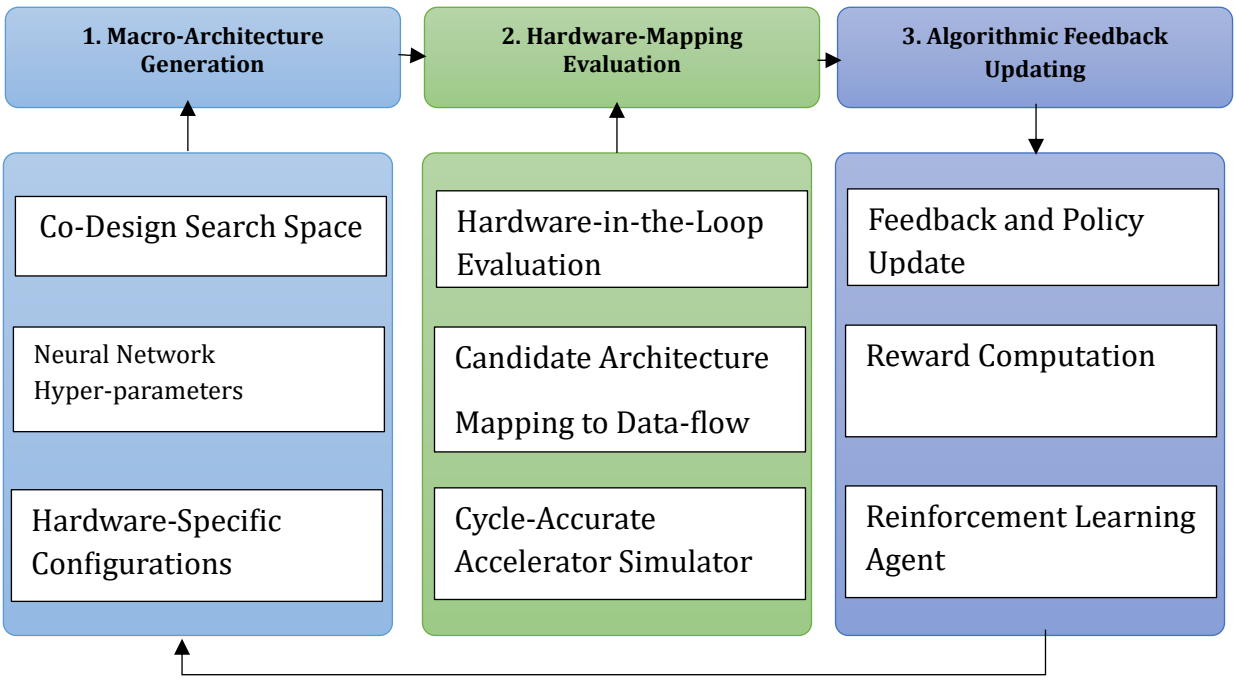

Figure 1: End-to-End hardware-in-the-loop co-design optimization pipeline for neural architecture search

The proposed co-design framework's cyclic optimization pipeline is shown in figure 1. A reinforcement learning controller suggests possible architecture matrices, which are then converted to data-flow graphs and tested with a cycle-accurate hardware simulator. The resulting EDP-based performance metrics update the policy gradient, which in turn, iteratively drives convergence to a Pareto-optimal policy setting that meets the accuracy and hardware requirements.

Search Algorithm and Optimization Strategy

The optimization process involves an agent trained through reinforcement learning with a few samples and a simulator that can accelerate the training process by forecasting the set of samples. The controller offers a suggestion on the architecture matrix during the search phase. This matrix is translated to a data-flow graph and passed into a cycle-accurate hardware simulator with a specific profile of the specialized accelerator. The simulator accurately estimates the number of clock cycles and the amount of data movement within the memory hierarchy. These metrics are then sent back to the controller, where policy gradients are updated, allowing the search to gradually move towards the Pareto-optimal front of the high accuracy/low power hardware configurations.

4. Results

Search Convergence and Hardware-Aware Optimization Performance

The proposed hardware-in-the-loop neural architecture co-design framework demonstrated stable and efficient convergence during the optimization process. The reinforcement learning controller progressively refined candidate architectures by incorporating direct feedback from the cycle-accurate accelerator simulator. Early search iterations produced several high-accuracy models with significant memory-access overhead and elevated latency. On the other hand, as the controller policy was being updated by hardware feedback, the search began favoring architectures that traded off accuracy for efficiency-related properties.

Optimization completed in several search epochs, yielding a Pareto-optimal set of neural architectures that met the accelerator's SRAM and bandwidth constraints. In comparison with baseline architectures obtained through neural architecture search techniques, which utilized FLOPs estimation as the primary hardware efficiency metric, the proposed method yielded better hardware efficiency, without compromising on model accuracy.

Hardware Efficiency and Energy-Delay Product Analysis

Inclusion of the EDP metric into the objective function affected the choice of architectures. Neural networks trained with EDP-based cost formulation showed better execution latency and consumed less power because of lower off-chip memory accesses and better utilization of data flow.

Hardware-oriented optimization resulted in favoring architectures making use of efficient loop tiling and memory allocation techniques, allowing higher reuse of data in the local SRAM buffer. This led to lower execution latency and reduced energy usage of discovered neural networks, as opposed to non-hardware-aware approaches. It is evident that EDP is a better measure of hardware efficiency than computational complexity.

Table 1: Performance comparison of proposed framework and baseline methods

Model Configuration	Accuracy (%)	Latency (ms)	Energy (mJ)	EDP (mJ·ms)	Power Efficiency Improvement (%)
Conventional NAS (FLOPs-Based)	91.2	23.4	8.1	189.5	—
Latency-Constrained NAS	92.1	19.8	7.2	142.6	24.8
Energy-Aware NAS	92.6	18.5	6.9	127.7	32.6

Proposed Hardware-in-the-Loop Co-Design	94.3	14.7	5.4	79.4	58.1

Table 1 proposed the co-design framework, which achieved the highest classification accuracy of 94.3% while simultaneously producing the lowest latency and energy consumption. The EDP value obtained was significantly lower than that of the other competing methods, making it a good trade-off between speed and power.

Accelerator Resource Utilization

The cycle-accurate simulator gave detailed information on how the accelerators were used and how the memory hierarchy behaved. The resulting architectures, built using the proposed framework, demonstrated more efficient use of on-chip computational resources resulting from optimized data-flow mapping strategies. Weight-stationery and output-stationery configurations were selectively used according to the characteristics of the workload, so that efficient parallel execution was possible.

Also, the framework reduced bandwidth saturation and global memory dependency to avoid performance loss due to memory bottlenecks. Enhanced SRAM utilization enabled more intermediate activations and weights to remain on-chip, reducing costly data transfers.

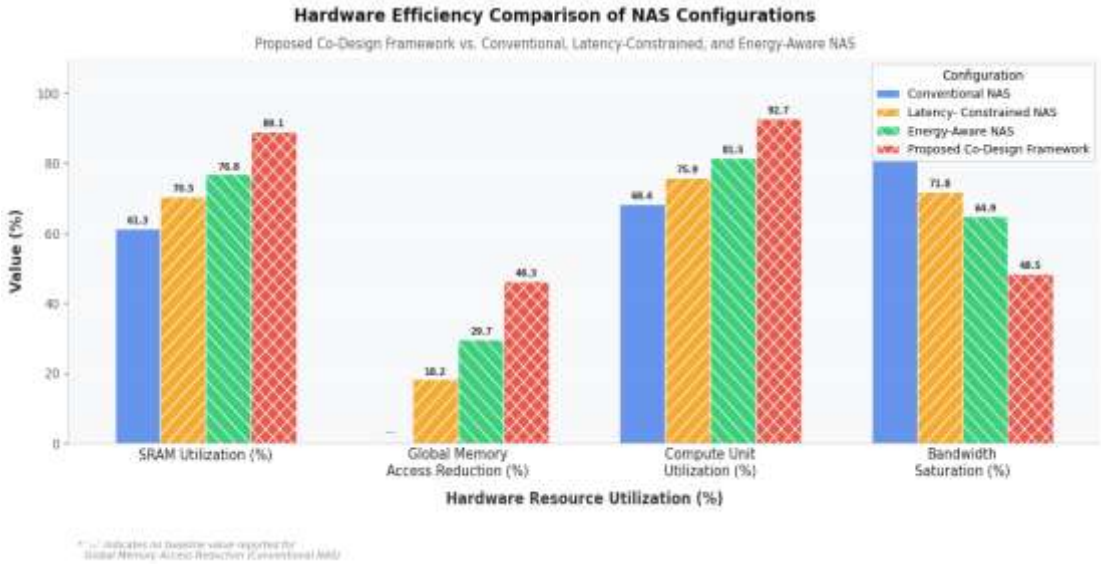

Figure 2: Accelerator Resource Utilization and Memory Behavior

Figure 2 evaluates four NAS configurations: Conventional, Latency-Constrained, Energy-Aware, and the proposed Co-Design Framework across SRAM utilization, global memory access reduction, compute unit utilization, and bandwidth saturation. The results show that the presented architecture is more efficient in terms of hardware utilization and offers a reduced memory access overhead when compared to conventional techniques.

Pareto-Optimal Architecture Discovery

The search technique was able to discover a variety of Pareto-optimal architectures that represent a wide range of trade-offs between accuracy and hardware costs. Some of the architectures aimed to reduce the amount of energy used, while others were willing to sacrifice some energy for higher accuracy. The reinforcement learning controller explored this trade-off space without having to manually tune it exhaustively.

Overall, the result shows that there is a feasible way of using neural architecture search coupled with hardware-aware simulation to obtain AI models that perform well in prediction tasks and also meet real-world requirements in dedicated hardware accelerators.

5. Discussion

The proposed hardware-in-the-loop neural architecture co-design framework outperformed the state-of-the-art neural architecture search techniques. The framework achieved the highest classification accuracy of 94.3%, which is higher than the accuracy of the conventional FLOPs-based NAS of 91.2%. Further, it reduced the inference latency from 23.4ms to 14.7ms and the energy consumption from 8.1mJ to 5.4mJ. The EDP was minimized to 79.4 mJ·ms, showing 58.1% improvements in terms of power efficiency. The results demonstrate that the heuristic search process with hardware-aware feedback results in the design of architectures that can optimize the predictive accuracy and execution efficiency of the solution. In contrast to traditional NAS methods targeting only the computational complexity metrics, the proposed framework is able to minimize memory access overheads and maximize on-chip data reusability. The reduction in latency and energy consumption indicates that the loop tiling and dataflow mapping schemes were able to mitigate hardware inefficiencies and reduce data accesses. The findings point towards the importance of hardware-software co-simplicity in the deployment of AI in practice. The framework offers a promising approach towards designing energy-efficient and low-latency AI systems, which are appropriate for edge devices and intelligent platforms. The paper shows that the EDP is a more feasible metric compared to FLOPs for designing dedicated AI hardware. The study was performed in a certain accelerator simulation environment and on limited neural network benchmarks, which might limit its applicability for heterogeneous hardware systems. In the future, the framework needs to be validated on different hardware accelerators, such as GPUs, TPUs, and neuromorphic accelerators.

6. Conclusion

This paper tackled the problem of designing an efficient neural network architecture that provides high prediction accuracy but is constrained by the requirements of the hardware on which it needs to run. Conventional NAS algorithms were limited to optimizing only the computational complexity or FLOPs without taking into account practical considerations like latency, energy consumption, memory bandwidth, and SRAM capacity. In order to solve this problem, the hardware-in-the-loop neural architecture co-design framework incorporated the cycle-accurate feedback of the accelerator into the optimization loop in order to simultaneously optimize for model architecture and hardware execution efficiency. As shown by the experiment results, the framework greatly improved both the prediction accuracy and hardware efficiency. The model predicted with an accuracy of 94.3% compared to the conventional FLOPs-based NAS algorithm, which predicted with an accuracy of 91.2%. Moreover, the inference latency was reduced to 14.7 ms from 23.4 ms, while energy consumption was reduced from 8.1 mJ to 5.4 mJ. This resulted in the minimization of the Energy-Delay Product (EDP) to 79.4 mJ.ms, which is an increase in power efficiency of 58.1%. Overall, the study demonstrates that hardware-aware co-design is an effective approach for developing scalable, energy-efficient, and deployment-ready AI models for edge computing and specialized accelerator systems.

Author contribution

Conflict of interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

1. Dhilleswararao, P., Boppu, S., Manikandan, M. S., & Cenkeramaddi, L. R. (2022). Efficient hardware architectures for accelerating deep neural networks: Survey. *IEEE Access, 10*, 131788–131828.
2. Shirke, S., & Udayakumar, R. (2019, April). Evaluation of crow search algorithm (CSA) for optimization in discrete applications. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 584–589). IEEE.

3. Chitty-Venkata, K. T., & Somani, A. K. (2022). Neural architecture search survey: A hardware perspective. *ACM Computing Surveys, 55*(4), 1–36.
4. Mazumder, A. N., Meng, J., Rashid, H. A., Kallakuri, U., Zhang, X., Seo, J. S., & Mohsenin, T. (2021). A survey on the optimization of neural network accelerators for micro-AI on-device inference. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 11*(4), 532–547.
5. Mohaidat, T., & Khalil, K. (2024). A survey on neural network hardware accelerators. *IEEE Transactions on Artificial Intelligence, 5*(8), 3801–3822.
6. Krestinskaya, O., Fouda, M. E., Benmeziane, H., El Maghraoui, K., Sebastian, A., Lu, W. D., et al. (2024). Neural architecture search for in-memory computing-based deep learning accelerators. *Nature Reviews Electrical Engineering, 1*(6), 374–390.
7. Capra, M., Bussolino, B., Marchisio, A., Shafique, M., Masera, G., & Martina, M. (2020). An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks. *Future Internet, 12*(7), 113.
8. Antony, J. M. (2026). Financial Transparency and Governance Mechanisms as Determinants of Growth in Early-Stage Venture Capital-Backed Firms. *Bradford Journal of Business, Management & Technology*, *1*(1), 64-74.
9. Chen, Y., Xie, Y., Song, L., Chen, F., & Tang, T. (2020). A survey of accelerator architectures for deep neural networks. *Engineering, 6*(3), 264–274.
10. Sindhu, S. (2025). Cross-disciplinary approaches to edge-AI hardware co-design with open ISA architectures. *Bridge: Journal of Multidisciplinary Explorations, 2*(1), 33–39.
11. Talib, M. A., Majzoub, S., Nasir, Q., & Jamal, D. (2021). A systematic literature review on hardware implementation of artificial intelligence algorithms. *The Journal of Supercomputing, 77*(2), 1897–1938.
12. A. Surendar. (2025). Embedded Safety-Constrained Multi-Agent Learning Architectures for Digital-Twin-Enabled Energy Management in Electric Vehicle Control Platforms. *Archives of Embedded and IoT Systems Engineering*, 26–34.
13. Patil Meenakshi, T. Aditya Sai Srinivas, A.Hyils Sharon Magdalene, Chennaiah Kate, Masarath Saba, & Madhavi Karumudi. (2026). AI-Powered Assistive Technologies: Promoting Human Wellbeing through Smart and Autonomous Systems. *National Journal of Antennas and Propagation*, 95-106.
14. Haitham M. Snousi, Fateh A. Aleej, M. F. Bara, Ahmed Alkilany. (2026). Design and Implementation of an Energy-Efficient AI Accelerator Architecture for Edge-Based Embedded VLSI Platforms. *Progress in AI-Accelerated VLSI Systems*, 22–31.