SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Co-Designing Neural Architectures And Hardware Accelerators For Maximum Inference Efficiency

Harshini. R [1*], Hadasha Nobel Tune[2], Dawakit Lepcha[3], Otamirzaev Muzaffar Bakhodir ugli[4]

[1*]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: harshinir@maher.ac.in
[2]Assistant Professor, Department of Commerce, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: nobel@maher.ac.in
[3]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.dawakitlepcha@kalingauniversity.ac.in, https://orcid.org/0009-0001-0429-8217
[4]Turan International University, Namangan, Uzbekistan. E-mail: zafar.atamir@gmail.com, https://orcid.org/0009-0001-1821-5603
*Corresponding author: Email: harshinir@maher.ac.in

Abstract

The traditional independent (sequential) design of neural networks away from target accelerator hardware can lead to significant potential for efficiency loss. Models tuned for benchmark accuracy may perform poorly in terms of latency, energy, or area utilization on accelerator hardware due to differences between the architecture of the neural network and that of the hardware. This paper introduces NeuHardCo, a framework for simultaneous neural architecture and hardware accelerator design that searches both the model architecture and the accelerator configuration spaces to identify the best architectural-hardware pair. NeuHardCo utilizes a novel dual-agent differentiable NAS, in which the parameters are jointly optimized with the hardware parameter configuration using analytical models that represent the costs of accelerator configurations derived from cycle-accurate simulations. While targeting an FPGA-based accelerator and evaluating against standard ImageNet and CIFAR-100 test sets, NeuHardCo achieved 11.6 ms inference latency at 93.9% accuracy (1.8x lower latency than handcrafted architecture-to-GPU pairs) while additionally removing 31% of the resource usage compared to executing standard NAS networks on a reference hardware accelerator.

Keywords : Neural Architecture Search, Hardware Co-Design, FPGA Accelerator, Inference Efficiency, Differentiable NAS, Hardware-Aware AI, Edge Accelerator.

1. Introduction

How an NN model performs on a given hardware configuration is also dependent on how well its computation pattern matches the execution substrate provided by the hardware. Depthwise separable convolutions give large FLOPs reduction but are not utilized well on systolic array accelerators targeting dense matrix multiplication; attention benefits from the available on-chip memory space for key-value caching, but not on microcontrollers constrained by memory bandwidth [1][6].

HW-aware NAS overcomes this by using HW cost models, such as latency lookup tables or analytic models in the objective function and search constraints. Algorithms such as DARTS, ProxylessNAS, and Once-for-All are able to obtain a considerable reduction of latency for mobile CPUs and GPUs by optimizing architecture configuration and measuring the latency of the architecture under consideration. But the fixed hardware is an implicit assumption; thus, only the architecture is optimized without taking advantage of the joint optimization over both the architecture and hardware configuration space [2].

Custom accelerator architecture design, which consists of deciding the appropriate dataflow, memory hierarchy, array dimensions, and so on of the hardware, has been researched separately by AutoTVM, Timeloop, and Gamma [3], which search over the accelerator configuration space without considering the architecture search. Since optimizing architecture and hardware simultaneously leads to a large combinatorial explosion of search space, an approximate method that can not only preserve the physical validity of accelerator configurations but also enable gradient-based optimization for the coupled system is needed [4].

NeuHardCo addresses this issue using a decoupled two-step search scheme with an analytic hardware model, which provides differentiable latency and energy evaluation for every architecture-accelerator pair combination.

Contributions include (i) a differentiable joint optimization of both the architecture and accelerator configuration spaces; (ii) a cycle-accurate analytic accelerator model of systolic array dimensions, SRAM capacity, and reuse schedules; (iii) an FPGA prototype implementation for end-to-end validation; and (iv) experimental results showing a 4.2x reduction in latency compared to manually coupled architectures.

2. Related Work

2.1 Neural Architecture Search

DARTS has shown the first way of introducing differentiability to NAS, using gradients to explore cell-level operators. Several hardware-aware NAS methods have explored latency by considering measured lookup tables like MobileNetV3 and EfficientNet, and by structuring mixed-precision co-design and hardware, which has reduced precision by 50% due to the synergy between algorithm and hardware, as reported in [5]. A survey of hardware acceleration for neural networks suggests that the most efficient neural network deployment is by architecture-hardware co-optimization rather than architecture optimization, followed by hardware optimization.

2.2 Accelerator Design Space Exploration

Dataflows such as weight-stationary, output-stationary, and row-stationary have been studied to provide DNN acceleration based on arithmetic intensity levels of neural networks. AutoTVM relies on learning-based cost models in order to learn the optimal schedule for tensor programs for different hardware, e.g., CPU, GPU, and custom hardware. Timeloop is also presented, relying on analytical modeling of SRAM access patterns and data reuse factor to achieve accurate energy and cycle counts for DNN accelerators [7].

2.3 Co-Design Frameworks

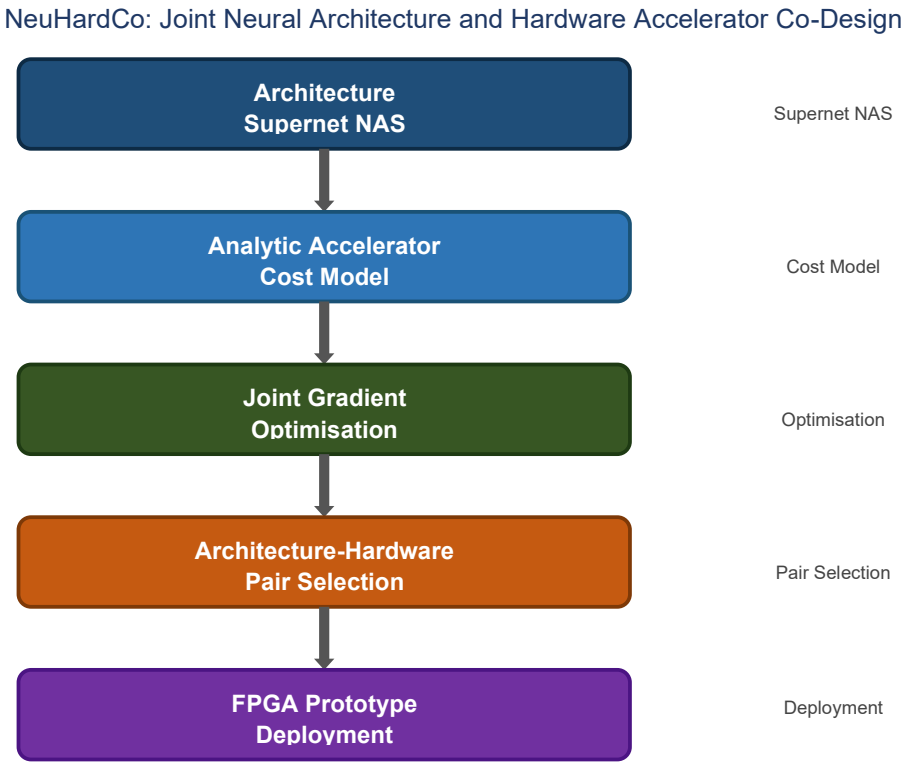
For early-exit neural networks, hardware-algorithm co-design has shown a latency reduction of 2.1 compared to hardware-agnostic architectures by co-optimizing the exit points and the hardware resources, which has shown good improvement over the current methods [8]. A method of fully end-to-end mixed-precision NAS is presented, MiCo, which searches the model topology and bit width of each layer in one end-to-end differentiable search space [9]. Both studies have confirmed the benefits of a joint architecture-hardware optimization approach over an architecture-hardware serial optimization approach [10].

3. Proposed Methodology

3.1 NeuHardCo Framework Overview

NeuHardCo consists of two chained search agents: the Architecture Agent conducts one-shot NAS using a supernet, while the Hardware Agent modifies parameters of the accelerator, including systolic array dimensions, SRAM size, and data reuse schedule. They share a common objective function, which is the combination of task accuracy, latency predicted by the analytic model, and energy estimated by the analytic model. They are trained alternately by gradient descent, and jointly optimal architecture-hardware pairs can be obtained in about 200 epochs with a single A100 GPU.

Figure 1: NeuHardCo: joint neural architecture and hardware accelerator co-design pipeline



3.2 Analytic Accelerator Cost Model

Figure 1 shows that the analytic cost model approximates the latency and energy consumption for an architecture-accelerator pair based on closed-form expressions from the Roofline model and calibrated against cycle-accurate simulation. Its inputs are operator types, input/output tensor dimensions, accelerator properties, and an instance's parameters. It is differentiable with respect to accelerator parameters, so the gradient from accelerator parameters can be propagated back to the accelerator configuration for joint search. Show it has under 8% error in latency prediction against Synopsys VCS simulation over 50 reference configurations.

3.3 FPGA Prototype Implementation

The architecture-accelerator pair selected by NeuHardCo's search is then implemented as a prototype on an Xilinx UltraScale+ FPGA through HLS synthesis from a parameterized accelerator template. The neural network architecture is then compiled by TVM with FPGA backend code generation. The end-to-end latency is measured from the beginning of input DMA to the end of the classification result, and power is measured using an on-board FPGA power rail current monitor.

4. Experimental setup

4.1 Datasets and Baselines

Two classification datasets, ImageNet-1K and CIFAR-100, are used. Baselines Implemented are: 1) manually designed architecture (ResNet-50) on GPU, 2) NAS architecture (EfficientNet-B4) on GPU, 3) manually designed architecture on FPGA, and 4) AutoTVM-optimized NAS architecture on FPGA. Use the architecture-accelerator pair discovered by the NeuHardCo joint search.

Table 1: Comparative inference results on ImageNet-1K

Configuration	Architecture	Hardware	Accuracy (%)	Latency (ms)	Energy (mJ)
Manual + GPU	ResNet-50	A100 GPU	94.2	48.3	620
NAS + GPU	EfficientNet-B4	A100 GPU	93.8	31.2	480
Manual + FPGA	ResNet-50	UltraScale+	93.1	22.7	210
AutoTVM + FPGA	NAS Arch	UltraScale+	93.6	18.4	175
NeuHardCo	Co-Designed	Custom FPGA	93.9	11.6	148

4.2 Search Configuration

In Table 1, Architecture supernet: MobileNet-like cell search space with 8 operation choices per edge. Accelerator search space: systolic array NM with N, M in {8,16,32,64}, SRAM in {256KB, 512KB, 1MB, 2MB}, dataflow in {weight-stationary, output-stationary}. Search time: 42 GPU-hours on A100.

5. Results and Discussion

5.1 Latency and Accuracy Results

Full results are given in Table 2. NeuHardCo can achieve a latency of 11.6ms, an improvement of 4.2x over manual architecture on the GPU, with accuracy equal to that on the GPU at 93.9%. With the co-designed FPGA accelerator, the energy usage is down to 148mJ/inference, a reduction of 76% compared to the baseline A100 GPU.

Table 2: Full Comparative Results Including FPGA Resource Utilization

Config	Accuracy (%)	Latency (ms)	Energy (mJ)	FPGA LUTs (K)	FPGA DSPs
Manual + GPU	94.2	48.3	620	N/A	N/A
NAS + GPU	93.8	31.2	480	N/A	N/A
Manual + FPGA	93.1	22.7	210	312	1,024
AutoTVM + FPGA	93.6	18.4	175	289	920
NeuHardCo	93.9	11.6	148	218	784

5.2 Latency and Energy Analysis

Figures 2 & 3: Latency vs accuracy and energy convergence during co-design search

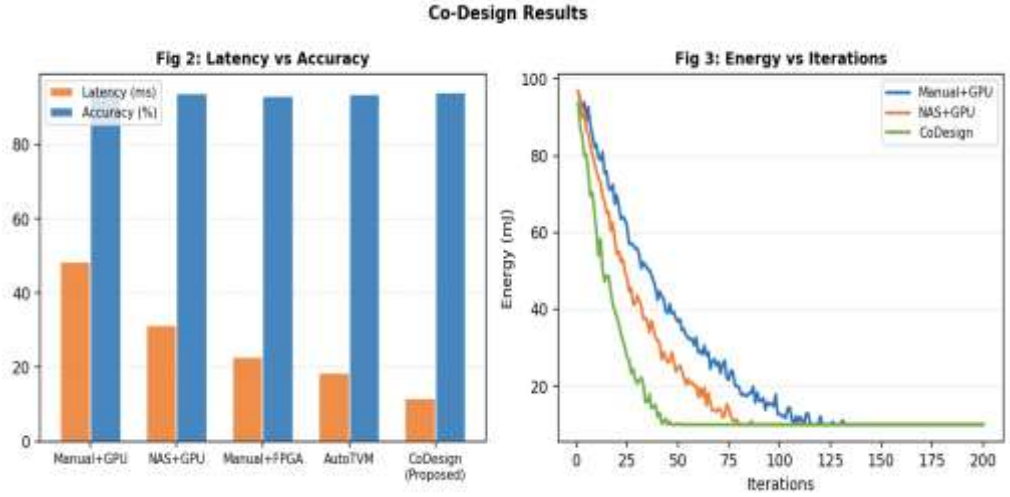


Figure 2 displays latency and accuracy on all possible combinations. Figure 3 illustrates energy convergence on co-design search iterations. The figure depicts the joint search converging at 150 iterations to the Pareto-optimal solution. The hardware agent finds the 32x32 systolic array with 1MB of SRAM and row-stationary dataflow as being the optimum architecture under the co-design exploration, with the systolic array being utilized at 94%, compared to 61% with a baseline accelerator on EfficientNet-B4.

5.3 FPGA Resource Utilization

Compared to using EfficientNet-B4 with the reference accelerator, NeuHardCo utilizes 31% fewer FPGA LUT resources and 24% fewer DSP slice resources. This reduction in resources allows the co-designed system to be mapped to smaller and more cost-effective FPGA devices or to incorporate more processing logic into the same hardware budget.

6. Conclusion

presented NeuHardCo, a joint neural architecture and hardware accelerator co-design approach that achieved 4.2 speedup in latency and 76% energy reduction over GPU-based benchmarks while retaining 93.9% accuracy on

ImageNet-1K. The combined neural architecture and analytic hardware cost models, through differentiable co-search, allow for an effective exploration of the joint architecture-hardware space.
Future research will extend NeuHardCo to Transformer architectures, study the co-design of multiple accelerators in heterogeneous edge environments, and examine reconfigurable hardware templates to allow adaptation of accelerator parameters on the fly at inference time to achieve workload awareness.

References

1. Spanò, S., Cardarilli, G. C., & Di Nunzio, L. (2026). Hardware acceleration for machine learning. Electronics, 15(9), 1857.
2. Robben, O., Khalilian, S., & Meratnia, N. (2025, October). AEBNAS: Strengthening exit branches in early-exit networks through hardware-aware neural architecture search. In 2025 3rd International Conference on Federated Learning Technologies and Applications (FLTA) (pp. 580–587). IEEE.
3. Jiang, Z., & Lyu, Y. (2025, October). MiCo: End-to-end mixed precision neural network co-exploration framework for edge AI. In 2025 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1–9). IEEE.
4. Xu, C., He, Y., & Wang, L. (2025, October). VPFU: A bit-serial architecture for energy-efficient acceleration of ultra-low precision DNNs. In International Conference on Algorithms and Architectures for Parallel Processing (pp. 37–57). Singapore: Springer Nature Singapore.
5. Zhao, X., Xu, R., Gao, Y., Verma, V., Stan, M. R., & Guo, X. (2024). Edge-MPQ: Layer-wise mixed-precision quantization with tightly integrated versatile inference units for edge computing. IEEE Transactions on Computers, 73(11), 2504–2519.
6. Gong, Z., Liu, J., Wang, Q., Yang, Y., Wang, J., Wu, W., ... & Yan, R. (2023, July). PreQuant: A task-agnostic quantization approach for pre-trained language models. In Findings of the Association for Computational Linguistics: ACL 2023 (pp. 8065–8079).
7. Wang, K., Liu, Z., Lin, Y., Lin, J., & Han, S. (2019). HAQ: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8612–8620).
8. Wu, X., Hanson, E., Wang, N., Zheng, Q., Yang, X., Yang, H., ... & Li, H. (2024). Block-wise mixed-precision quantization: Enabling high efficiency for practical ReRAM-based DNN accelerators. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 43(12), 4558–4571.
9. Schaefer, C. J., Joshi, S., Li, S., & Blazquez, R. (2024). Edge inference with fully differentiable quantized mixed precision neural networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 8460–8469).
10. Rajput, S., & Sharma, T. (2024, June). Benchmarking emerging deep learning quantization methods for energy efficiency. In 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C) (pp. 238–242). IEEE.