Research Statement

I am Hajar Falahati, a senior postdoctoral researcher at the School of Computer Science, Institute for Research in Fundamental Sciences (IPM). Throughout my postdoctoral program, my research has primarily focused on two key areas:

1. General-purpose graphic processing units (GPGPUs), simply called GPUs, encompassing power-efficiency, memory management, and heterogeneous CPU-GPU systems.

2. Machine learning (ML), involving the design of hardware architectures for ML algorithms, the implementation of energy-efficient strategies using learning-based approaches, and the development of ML-based applications across various domains.

The following sections provide a brief overview of my prior research (Postdoctoral Research) and outline my future research initiatives (Faculty Research Proposal).

Postdoctoral Research

As a postdoctoral researcher, my research was focused on two main areas:

I. GPUs:

GPUs, known for their highly programmable and multithreaded nature, serve as a primary accelerator across various application domains. These GPUs consist of stream multiprocessors (SMs) connected to memory controllers via a network-on-chip, leveraging extensive thread parallelism to mask the extended latency of memory accesses. However, they encounter challenges related to low energy efficiency, underutilization of resources, and costly memory accesses in terms of both power consumption and performance.

To address these issues, our research has delved into several key areas:

1. Memory wall issue mitigation: We proposed a chain-based prefetching [1] and explored prefetching data within a heterogeneous CPU-GPU architecture to tackle this challenge.
2. High power consumption: Our work involves estimating idle periods and harnessing various power management techniques, such as power-gating and voltage scaling, and reducing bank conflicts in a multi-level low-power register file [2, 3].
3. Memory system inefficiency: We explored strategies to exploit data reuse among different thread blocks assigned to diverse SMs. This included facilitating neighbor data exchange, enhancing the scheduler to maximize data reuse, proposing a unified on-chip cache to manage both L1D and shared memory accesses, and exploring the potential of a scalable 3D-memory [4, 5, 6, 7, 8, 9].

Our efforts have resulted in the production of nine papers: seven have been accepted, and two are currently in preparation, addressing and advancing these crucial aspects within GPU architecture and performance optimization.

II. *Machine Learning:*

Machine learning (ML) algorithms, particularly neural networks (NNs) - among the most promising, find extensive use in a multitude of applications, progressing towards larger and deeper structures (DNNs). The execution of DNNs involves managing a vast number of parameters, leading to intensive computations, increased memory accesses, on-chip storage requirements, demands on off-chip and on-chip memory bandwidth, and overall high energy consumption.

While ML algorithms traditionally run on general-purpose platforms (e.g., CPUs and GPUs) these platforms incur significant energy expenses. In response, both academia and industry have witnessed numerous endeavors to design specialized hardware accelerators. However, these tailored hardware accelerators encounter challenges regarding flexibility and contend with frequent data movements between the memory system and the hardware accelerator.

We have conducted extensive research focusing on several key areas:

1. Development of a specialized accelerator for generative adversarial networks [11].
2. Exploration of a compression dataflow approach leveraging input and weight similarities, processing only unique inputs and weights to harness potential sparsity [12].
3. Introduction of a compression framework ensuring an equal distribution of unique inputs to unique weights and devising dedicated hardware support to maximize the proposed compression dataflow's potential [13].
4. Creation of a hardware accelerator ensuring both flexibility and energy efficiency in training various ML algorithms. This involves representing ML algorithms as compute patterns executed on customized compute engines, and the proposal of a fast, scalable network-on-chip for DNN accelerators [14, 15].
5. Introduction of a compression weight framework suggesting a novel weight encoding for large-value weights with higher precisions. This includes hardware support for accelerating DNNs using the proposed weight quantization approach [17].
6. Development of a reinforcement learning approach aimed at managing energy consumption in smartphones [18].

Our efforts have resulted in the production of nine papers: two accepted papers, one in arXiv, and six currently under preparation. These contributions aim to address various challenges and advance the field of machine learning, particularly in optimizing hardware acceleration, compression techniques, and energy management strategies.

Faculty Research Proposals

As an applicant for the faculty program as an assistant professor at IPM, my focus for future research aims to center around several key areas within the realm of hardware accelerators and machine learning paradigms.

I. *Hardware accelerators for different machine learning paradigms:*

   While prior research has focused on developing compression techniques to alleviate memory and computation demands, these techniques often render DNNs irregular in their structure. Additionally, other avenues of research have delved into alternative execution paradigms, such as non-von Neumann architectures. This includes exploring analog accelerators, in-memory processing using conventional DRAMs or non-volatile memories (e.g., Resistive Random Access Memory - ReRAM), spiking neural networks, and investigating hyper-dimensional computing (HDC). These alternative paradigms aim to offer innovative solutions for the efficient processing and execution of complex neural networks.

   1. Training: While the training of DNN models traditionally occurs on robust cloud-based systems, there's a growing interest in conducting this training on edge devices, known as edge computing. This shift is driven by considerations such as latency, privacy concerns, and limitations in communication bandwidth [18].

      We aim to facilitate training on edge devices involving the application of data compression techniques. These techniques aim to reduce both the computational load and the volume of data processed, allowing for more efficient training directly on these edge devices.

   2. Sparsity in ReRAM-based DNN accelerators: The crossbar structure of ReRAMs makes them a promising alternative for accelerating DNNs where computation is handled via analog signals. However, ReRAM-based architectures encounter two primary challenges, limiting their effectiveness in accelerating DNNs:

      i. Inefficiency in sparse model acceleration: ReRAM architectures are proficient in processing dense DNN models but struggle to effectively accelerate sparsity. This limitation arises from the necessity for complete rows or columns to be entirely zero, restricting the potential for sparsity to specific sparse patterns.

      ii. Uncertainty robustness issues: These accelerators are susceptible to uncertainty and robustness issues due to imperfections associated with the computational substrates. As computation occurs using the physical attributes of memristive devices, uncertainties arise, affecting the accuracy of computations [19-24].

We conduct a review of previous proposals, highlighting their inherent limitations. Taking into account these challenges, we are considering two principal directions for addressing these issues:

i. Exploration of customized convolution techniques: Investigating alternative convolution techniques specifically tailored for the ReRAM structure to enhance the processing of sparse DNNs within this architecture.

ii. Pattern detection and arrangement for sparse DNNs: Detecting patterns within inputs and weights to organize sparse DNNs more uniformly. This involves arranging the sparsity in a more regular manner by considering either exact or approximate values, aiming to improve the efficacy of ReRAM-based DNN accelerators.

3. Sparsity in HDC-based learning methods: Hyper-dimensional Computing (HDC) represents an emerging computational framework inspired by essential elements of human memory. HDC leverages high-dimensional binary vectors, known as hypervectors, and the mathematical principles of hyperdimensional spaces to simulate human-like perception and cognition. The application of HDC in performing ML algorithms involves the manipulation and comparison of large patterns within memory, allowing for training in one or a few shots. Moreover, HDC inherently exhibits robustness to uncertainties, making it suitable for emerging non-von Neumann approaches like in-memory computing.

However, working with high-dimensional vectors entails significant computational and memory requirements. While certain research focuses on binary vectors and their sparsity, they often face challenges related to reduced accuracy. Other studies concentrate on in-memory processing, but these systems are vulnerable to frequent write operations and have limitations in terms of endurance [25-29].

II. *Developing learning-based medical devices:* Numerous initiatives in both industry and academia focus on enhancing the well-being of both healthy and disabled individuals by utilizing smart devices, such as smart glasses, wearable devices, and biochips.
Our aim is exploring the creation and implementation of medical devices that integrate learning-based approaches to enhance diagnosis, treatment, and healthcare outcomes. E.g., we aim to introduce portable medical devices designed to identify objects and humans, provide explanations, and extract human emotions [30]. These devices are intended to assist in various aspects of healthcare and wellness, catering to both healthy individuals and those with specific medical needs.

III. *Supporting learning models on GPUs:* GPUs stand as promising options for accelerating ML algorithms. Recent GPU advancements leverage tensor cores specifically designed for performing multiply-accumulator (MAC) operations, supporting diverse data representations, and managing memory accesses through customized units. However, executing DNNs on GPUs presents notable challenges [31-34].

Our objective is to support and optimize learning models on GPUs, ensuring efficient utilization of these resources by:

1. Optimizing memory storage: Storing data in both on-chip storage and global memory to minimize frequent data transfers among different memory models and between the CPU and GPU, respectively.
2. Addressing sparsity and data reuse: Tackling sparsity and enhancing data reuse within tensor cores to minimize frequent data movements while executing DNNs on GPUs.
3. Sparsity-aware execution strategies: Implementing reformations such as sparsity-aware warps and conducting a portion of execution within memory, utilizing on-chip cache or the register file, to further optimize the processing of DNNs on GPUs.

IV. *Employing learning-based approaches to manage GPU resources:* Despite benefiting from memory hierarchy and high thread level parallelism (TLP), GPUs still face challenges related to memory inefficiency and resource underutilization. In response to these issues, there is a growing focus on learning-based approaches.

Our goal is utilizing learning-based methodologies to effectively manage and optimize GPU resources, aiming to enhance their efficiency and performance across various applications, specifically aimed at:

1. Cache management: Implementing strategies to effectively manage caches, ensuring optimized data storage and retrieval, enhancing overall memory efficiency.
2. Throttling engine in prefetching techniques: Applying learning-based techniques to regulate the throttling engine within prefetching methods. This will facilitate more efficient and timely data retrieval.
3. Prefetching data in irregular applications: Developing learning-based approaches to prefetch data in irregular applications, catering to the specific challenges presented by these diverse and non-uniform workloads.

References

[1] S. Mostofi, H. Falahati, N. Mahani, P. Lotfi-Kamran, H. Sarbazi-Azad, “Snake: A Variable-length Chai-based Prefetching for GPUs,” MICRO, 2023.

[2] M. Sadrosadati, B. ehsani, H. Falahati, R. Ausavarungnirun, A. Tavakol, M. Abaei, L. Orosa, Y. Wang, H. Sarbazi-Azad, O. Mutlu, “ITAP: Idle-Time-Aware Power Management Technique for GPU Execution Units,” ACM TACO, 2019.

[3] M. Sadrosadati, A. Hajiabadi, A. Mirhosseini, S. B. Eslami, H. Falahati, H. Sarbazi-Azad, M. Drumond, B. Falsafi, R. Ausavarungnirun, O.Mutlu, “High Concurrency Latency Tolerant Register Files for GPUs,” ACM TOCS, 2021.

[4] N. Nematollahi, M. Sadrosadati, H. Falahati, M. Barkhordar, H. Sarbazi-Azad, “Neda: Supporting Direct Inter-Core Neighbor Data Exchange in GPUs,” IEEE CAL, 2018.

[5] N. Nematollahi, M. Sadrosadati, H. Falahati, M. Barkhordar, M. P. Drumond, H. Sarbazi-Azad, B. Falsafi, “Efficient Near-Neighbor Data Exchange in GPUs,” ACM TACO, 2020.

[6] H. Falahati, M. Sadrosadati, Q. Xu, J. G'omez-Luna, B. Saber, H. Jeon, S. Hessabi, H. Sarbazi-Azad, O. Mutlu, M. Annavaram, M. Pedram, “Cross-core Data Sharing for Energy-Efficient GPUs,” To be appread to ACM TACO.

[7] E. Yousefzadeh-Asl-Miandoab, M. Sadrosadati, H. Falahati, S. Darabi, P.Khorsand, N. Akbarzadeh, P. Lotfi-Kamran, H. Sarbazi-Azad, “OSM: Off-Chip Shared Memory for GPUs,” IEEE TPDS.

[8] B. Saber, H. Falahati, M. Sadrosadati, S. Hessabi, “Data Sharing Aware Scheduling for Reducing Memory Accesses in GPUs,” Preparing to be Submitted to IEEE TC.

[9] N. Akbarzadeh, M. Sadrosadati, H. Falahati, J. Gómez Luna, H. Sarbazi Azad, O. Mutlu, “Panacea: A High-bandwidth, High-capacity Memory for GPUs,” Preparing to be submitted to Sigmetrics.

[10] A. Yazdanbakhsh, H. Falahati, P. J. Wolfe, K. Samadi, N. S. Kim, H. Esmaeilzadeh, “GANAX: AUnified MIMD-SIMD Acceleration for Generative Adversarial Networks,” ISCA, 2018.

[11] H. Falahati, M. Peyro, H. Amini, M. Taghian, M. Sadrosadati, P. Lotfi-Kamran, H. Sarbazi Azad, “Data-Aware Compression of Neural Networks,” IEEE CAL, 2021.

[12] H. Falahati, N.Mahani, H. Amini, F. Khashei, P. Lotfi-Kamran, H. Sarbazi Azad, “A comprehensive Data-Aware Compression Framework for Accelerating Deep Neural Networks,” Preparing to be submitted to ISCA 2024.

[13] H. Falahati, P. Lotfi-Kamran, M.Sadrosadati, H.Sarbazi-Azad, “ORIGAMI: A Heterogeneous Split Architecture for In-Memory Acceleration of Learning,” arXiv 2018.

[14] H. Falahati, P. Lotfi-Kamran, M.Sadrosadati, J. G'omez-Luna, M. Barkhordar, H.Sarbazi-Azad, O. Mutlu, “Fractal: Compute-Pattern-Aware Acceleration of Machine Learning Algorithms,” Submitted to HPCA, 2022.

[15] F. Tahmasebi, H. Falahati, M. Sadrosadati, H. Sarbazi Azad, “A Fast and Scalable Network-on-Chip for DNN Accelerators,” Preparing to be submitted to IEEE TECS.

[16] A. Khayati, H. Amini, H. Falahati, B. Fazeli, “Data-aware quantization,” Preparing to be Submitted to IEEE TC.

[17] E. Aghapour, M. Sadrosadati, H. Falahati, A. Pathania, T. Mitra, H. Sarbazi Azad, “Deep Reinforcement Learning Based Power Manger to MinimizeEnergy Consumption while Satisfying Quality of Service,” Preparing to be submitted to sigmetrics.

[18] V. Sze, Y.U. Chen, T.J, Yang, J.O. Emer, “Efficient Processing of Deep Neural Networks,” Springer, 2020.

[19] T. H. Yang, H. Y. Cheng, C. L. Yang, I. C. Tseng, H. W. Hu, H. S. Chang, H. P. Li “Sparse ReRAM Engine: Joint Exploration of Activation and Weight Sparsity in Compressed Neural Networks,” ISCA 2019.

[20] W. Zhang, G. Bin, T. Jianshi, Y. Peng, Y. Shimeng, C, Meng-Fan, Y. Hoi-Jun, Q. He, W. Huaqiang, “Neuro-Inspired Computing Chips,” Nature Electronics, 2020.

[21] L. Deng, L. Guoqi, H. Song, S. Luping, X. Yuan, “Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey,” Proceedings of the IEEE, 2020.

[22] N. Challapalle, S. Rampalli, L. Song, N. Chandramoorthy, K.. Swaminathan, J. Sampson, Y. Chen, V. Narayanan, "GaaS-X: Graph Analytics Accelerator Supporting Sparse Data Representation Using Crossbar Architectures," ISCA 2020.

[23] S. Mittal, "A survey of ReRAM-based Architectures for Processing-In-Memory and Neural Networks," MDPI, 2019.

[24] G. Karunaratne, M. Gallo, G. Cherubini, L. Benini, A. Rahimi, A. Sebastian, "In-memory Hyperdimensional Computing," Nature Electronics, 2020.

[25] E. Hassan, Y. HALAWANI, B. Mohammad, H. Saleh, "Hyper-Dimensional Computing Challenges and Opportunities for AI Applications," IEEE Access, 2021.

[26] M. Imani, A. Rahimi, D. Kong, T. Rosing, J. M. Rabaey, "Exploring Hyperdimensional Associative Memory," HPCA, 2017.

[27] M. Imani, Z. Zou, S. Bosch, S. A. Rao, S. Salamat, V. Kumar, Y. Kim, T. Rosing, " Revisiting HyperDimensional Learning for FPGA and Low-Power Architectures," HPCA 2021.

[28] G. Karunaratne, M. Schmuck, M. L. Gallo, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, "Robust High-Dimensional Memory-Augmented Neural Networks," Nature Communications, 2021.

[29] W. Xu, Y. Zhang, X. Tang, "Parallelizing DNN Training on GPUs: Challenges and Opportunities, " WWW 2021.

[30] Li, Shan, and Weihong Deng. "Deep facial expression recognition: A survey." IEEE Transactions on Affective Computing (2020).

[31] B. Feng, Y. Wang, T. Geng, A. Li, Y. Ding, "APNN-TC: Accelerating Arbitrary Precision Neural Networks on Ampere GPU Tensor Cores," SC 2021.

[32] S. Dong, Y. Sun, N. B. Agostini, E. Karimi, D. Lowell, J. Zhou, J. Cano, J. L. Abell, D. Kaeli, "Spartan: A Sparsity-Adaptive Framework to Accelerate Deep Neural Network Training on GPUs, " IEEE TPDS, 2021.

[33] T. Gale, M. Zaharia, C. Young, E. Elsen, "Sparse GPU Kernels For Deep Learning," SC, 2020.

[34] Z. Gong, H. Ji, C. W. Fletcher, C. J. Hughes, S. Baghsorkhi, J. Torrellas, "SAVE: Sparsity Aware Vector Engine for Accelerating DNN Training and Inference on CPUs," MICRO 2020.