Machine Learning Model Adaptations for Execution in Energy-Constrained Intermittent Systems

João Gabriel Peixoto
Department of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
joaogdrumond@aluno.puc-rio.br

Adriano Branco
Department of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
abranco@inf.puc-rio.br

Markus Endler
Department of Informatics
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
endler@inf.puc-rio.br

***Abstract*—With the global deployment of more Internet of Things (IoT) devices, the demand for running increasingly complex applications, such as predicting parameters in intricate systems or processing audio and images within the machine learning domain, has driven adaptations to existing models to lower memory and computational demands. Meanwhile, intermittent computing has brought forth a new paradigm for embedded, energy-constrained systems, requiring further modifications to the machine learning models and algorithms. Using approximate computing concepts to enable model execution in a transient energy environment has shown promising results. While advances in intermittent machine learning have enabled more neural network applications to be developed, many techniques and models haven't been widely tested and optimized yet. This work proposes conducting a comprehensive survey and outlines initial findings, positing further research needs to be done to understand the current scenario of intermittent machine learning and discover possible advances.**

***Index Terms*—intermittent computing, machine learning, model compression, microcontrollers, Internet of Things**

I. Introduction

A surge in Internet of Things (IoT) devices worldwide has been accompanied by the development of software frameworks and hardware accelerators to perform more complex operations and execute more demanding algorithms on less resourceful computing systems, such as machine learning models [1]. The advent of LiteRT [2] (formerly TensorFlow Lite), which is a runtime for artificial intelligence algorithm deployment in embedded systems, has enabled novel applications in the low-power embedded domain for micro-controllers previously unaccounted-for: those demanding machine learning usage. This was due to the challenges in developing meaningful applications, considering the huge resource scarcity. Moreover, hardware accelerators have been developed, as well as microcontrollers specifically for executing machine learning and, more generally, artificial intelligence applications, such as the LEA Accelerator in the MSP430FR board series[1], the MAX78000 by Maxim Integrated[2], and the ESP32-S series by Espressif Systems[3].

Machine learning models, ranging from linear regression through support vector classifiers to very complicated deep neural networks, can provide great insight into datasets, such as those collected by sensor nodes deployed in a wireless sensor network (WSN) [3]. This can be true for both predicting future results in a time series or parameters in a complex system being monitored [4] or for clustering the results obtained into more manageable subsets, reducing excess communication and improving energy efficiency [5]. Image and audio processing in the embedded devices is also a novel and promising trend, considering the increased accuracy obtained by employing machine learning models in those domains, compared to the classic algorithms [6] [7]. Allowing execution of those complex statistical analysis tools locally on the edge devices, ranging from very low-power microcontrollers to single-board computers, such as a Raspberry Pi, can reduce network traffic and also the wait time for a result to be provided to the end user while increasing data throughput [8]. Moreover, communication is quite expensive and can consume large amounts of power, which could be used in more immediately relevant computations [9].

In tandem with developing machine learning frameworks and accelerators for microcontrollers, energy harvesting technologies have been thoroughly explored over the past decade. By drawing power from environmental sources, battery-free operation becomes viable, reducing the environmental impact associated with the rapid increase in IoT devices. This allows for untethered use and enables novel applications previously limited by the constant maintenance requirements of batteries. However, executing meaningful programs in this domain significantly differs from conventional execution models, leading to a new computing paradigm: intermittent computing. By creating new frameworks and adapting existing ones to accommodate the variable energy supply from harvested sources,

This work was financed in part by the Brazilian National Council for Scientific and Technological Development (CNPq).

[1] Texas Instruments, Low-Energy Accelerator (LEA) Frequently Asked Questions (FAQ), Document SLAA720, Nov. 2016.

[2] Maxim Integrated, Artificial Intelligence Microcontroller with Ultra-Low-Power Convolutional Neural Network Accelerator, MAX78000 Datasheet, May 2021.

[3] Espressif Systems, ESP32-S3-WROOM-1 & WROOM-1U Datasheet v1.3, ESP32-S3 Datasheet, 2023.

intermittent operation has become less challenging and increasingly beneficial. Nonetheless, further research is essential to optimize performance in this intermittent environment [10].

By combining machine learning models and algorithms with intermittent computing, new advancements can be achieved to enhance the sustainability of the Internet of Things. As an example, reinforcement learning has been explored as a way to optimize energy usage [11]. However, machine learning models are typically sensitive to contexts where data is faulty, inconsistent, or otherwise discrepant in formats and values. Such scenarios often arise due to intermittency — where computing intermittency may lead to incomplete results, and communication intermittency can cause invalid data packets — necessitating additional mechanisms to mitigate these issues and ensure reliable execution [9].

Another promising approach involves improving quality-of-service metrics, such as minimizing response times and increasing the frequency of value updates. This can be achieved by enabling local model execution on energy-constrained and computationally limited devices, therefore reducing communication latency. However, achieving this requires significant modifications to neural network models, reducing memory and computational demands [12] and enabling energy- and intermittence-aware execution [9]. Various techniques, such as quantization [13], [14], pruning [15] [16], tensor decomposition [17] [18], knowledge distillation [19] and neural architecture search [20], have been developed to support these changes. Some techniques, notably quantization, and pruning, have already been incorporated into frameworks like LiteRT, which enable machine learning model execution on microcontroller devices, achieving substantial model size reduction while maintaining high accuracy [21]. Additionally, approximate computing methods — primarily related to anytime algorithms, which facilitate a trade-off between energy consumption and accuracy [22] - provide greater flexibility. By employing early or multi-exit algorithms and models [22] [23], it becomes possible to trade the reduced accuracy for an increase in data throughput, which can be acceptable or even advantageous depending on a system's operational requirements [24].

II. Related Works

Several studies have developed deep neural network (DNN) models with adaptations to enable execution in intermittent environments. For instance, [22] introduces a multi-exit architecture and leverages quantization and pruning among the previously mentioned compression techniques. These rates are optimized by a multi-agent reinforcement learning algorithm, with rewards based on the average accuracy across all exits given an available energy trace. An online algorithm determines exit selection by assessing whether to proceed with additional inference or to exit early, balancing energy availability against the reduced accuracy resulting from not executing the entire network. As each exit requires specific training, the training process prioritizes the accuracy of the most likely exits. Another work, *ePerceptive* [23], proposes modifications to enhance the perceptive abilities of low-power devices using DNNs. Its neural network architecture accommodates various input resolutions, recognizing that lower-resolution data usually requires lower energy consumption. Furthermore, similarly to the previous approach, multiple exits enable the network to produce valid outputs even under conditions of energy scarcity. Model compression techniques like quantization and pruning have been effectively applied in intermittent scenarios. Some studies [22] have combined these techniques with architectural modifications, while others have concentrated on optimizing pruning specifically for intermittent environments [16].

III. Future Works

Despite significant architectural modifications proposed and executed in recent years, incorporating compression techniques into model design and further adjustments are still required for optimal execution. Multi- or early exiting, which has already been implemented, is just one of the approximate computing paradigm approaches that can be implemented for machine learning models. Meanwhile, among model compression techniques, the application of knowledge distillation and tensor decomposition in an intermittent environment has been lacking. Classically, quantization and pruning have been more explored, though there has been an interest in intermittent neural architecture search recently [20]. Moreover, most works have restricted themselves to neural network models and mostly convolutional [22], [23]. Support vector machines have been adapted to implement an anytime algorithm approach [24], but other machine learning architectures have been largely unexplored. Given the emerging nature of the field, conducting a survey could provide valuable insights into its current state and highlight potential areas for future advancements.

References

[1] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine learning for microcontroller-class hardware: A review," *IEEE Sensors Journal*, vol. 22, no. 22, pp. 21362–21390, 2022.

[2] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang, and P. Warden, "Tensorflow lite micro: Embedded machine learning on tinyml systems," 2021.

[3] L. F. Kim, K. Choi, S. Lee, and D. Har, "Machine learning for advanced wireless sensor networks: A review," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12379–12397, 2021.

[4] J. Pardo, F. Zamora-Martínez, and P. Botella-Rocamora, "Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes," *Sensors*, vol. 15, no. 4, pp. 9277–9304, 2015.

[5] S. K. and V. Vaidehi, "Clustering and data aggregation in wireless sensor networks using machine learning algorithms," in *2018 International Conference on Recent Trends in Advance Computing (ICRTAC)*, pp. 109–115, 2018.

[6] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. A. Velasco-Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," *CoRR*, vol. abs/1910.13796, 2019.

[7] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[8] D. Liu, H. Kong, X. Luo, W. Liu, and R. Subramaniam, "Bringing ai to edge: From deep learning's perspective," *Neurocomputing*, vol. 485, pp. 297–320, 2022.

[9] B. Gobieski, B. Lucia, and N. Beckmann, “Intelligence beyond the edge: Inference on intermittent embedded systems,” in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS ’19)*, (New York, NY, USA), pp. 199–213, Association for Computing Machinery, 2019.

[10] S. Ahmed, B. Islam, K. S. Yildirim, M. Zimmerling, P. Pawelczak, M. H. Alizai, B. Lucia, L. Mottola, J. Sorber, and J. Hester, “The internet of batteryless things,” *Communications of the ACM*, vol. 67, pp. 64–73, Feb. 2024.

[11] F. Fraternali, B. Balaji, Y. Agarwal, and R. K. Gupta, “Aces: Automatic configuration of energy harvesting sensors with reinforcement learning,” *ACM Transactions on Sensor Networks*, vol. 16, July 2020.

[12] D. Ghimire, D. Kil, and S.-h. Kim, “A survey on efficient convolutional neural networks and hardware acceleration,” 2022.

[13] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “Xnor-net: Imagenet classification using binary convolutional neural networks,” 2016.

[14] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, “Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients,” 2018.

[15] T.-J. Yang, Y.-H. Chen, and V. Sze, “Designing energy-efficient convolutional neural networks using energy-aware pruning,” 2017.

[16] C.-C. Lin, C.-Y. Liu, C.-H. Yen, T.-W. Kuo, and P.-C. Hsiu, “Intermittent-aware neural network pruning,” in *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023.

[17] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, “Predicting parameters in deep learning.”

[18] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” 2017.

[19] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015.

[20] H. R. Mendis, C.-K. Kang, and P.-c. Hsiu, “Intermittent-aware neural architecture search,” 2021.

[21] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” 2016.

[22] Y. Wu, Z. Wang, Z. Jia, Y. Shi, and J. Hu, “Intermittent inference with nonuniformly compressed multi-exit neural network for energy harvesting powered devices,” 2020.

[23] A. Montanari, M. Sharma, D. Jenkus, M. Alloulah, L. Qendro, and F. Kawsar, “Eperceptive: Energy reactive embedded intelligence for batteryless sensors,” in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 382–394, 2020.

[24] F. Bambusi, F. Cerizzi, Y. Lee, and L. Mottola, “The case for approximate intermittent computing,” in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 463–476, 2022.