Flexible ultra low power architecture supporting different artificial intelligence algorithms in the Internet of Things context

Project dates

From October 1, 2023 to September 30, 2026

Funding

Scholarship from the Ministry of Higher Education, Research and Innovation (MESRI) (approximately €2,045 gross/month) for three years.

Supervisors

Sébastien Pillement (IETR), Andrea Pinna (LIP6) and Pierre Langlois (Polytechnique Montréal)

Project description

This project aims to design computing architectures suitable for the implementation of a wide range of artificial intelligence (AI) and machine learning (ML) algorithms in the Internet of Things (Internet of Things - IoT).

AI and ML algorithms include data analysis operations for different purposes including prediction, classification, segmentation, preparation of recommendations, and decision making. These algorithms usually require the manipulation of a large amount of information and involve a large number of calculations. It is expected that the results produced by these algorithms will be correct and precise, that they will be available when needed, and that the processing will use as little energy as possible. These expectations often imply that AI and ML algorithms must be implemented in sophisticated computational architectures with particular memory hierarchies and interfaces, characteristics that often go beyond those of traditional microprocessors.

Implementing AI and ML algorithms in the Internet of Things (IoT) presents significant additional challenges. The objects vary greatly in nature, size, function or capabilities, and include smartphones, self-driving cars, smart cameras, household appliances and miniaturized drones. They integrate sensors, processing units, software, communication systems and sometimes actuators. The very nature of the AI algorithms implemented in each object varies depending on its task and its specific functionalities. In the most extreme cases, we can imagine objects that must operate completely independently for days, months or years, and whose nature of the task changes depending on time, place and circumstances. It is therefore reasonable to assume that these objects will have to be capable of implementing a wide variety of ML algorithms during their useful life.

There are several classes of ML algorithms: decision trees, random forests, k-nearest neighbors, and different variants of neural networks (basic, convolutional). , feedback, etc.). These algorithms involve different calculation operators, from the nature of the operations carried out to the precision required. Each calculation can relate to a small or large number of data which may or may not be collocated in memory. Memory access patterns can thus be very varied. Depending on the application considered, the implementation of one or more of these classes of algorithms may be necessary. Furthermore, the nature of the calculations and the data used can be significantly different in the inference and training phases of the system. It is difficult for a single processor to optimally implement all types of operations and memory access corresponding to the different classes of ML algorithms.

This project focuses on the design of a processor that can effectively implement different AI and ML algorithms for IoT objects. The processor must be flexible, that is to say, it must be able to adapt easily and quickly to different classes of algorithms. The hierarchy and memory interfaces of the system must also easily adapted to each class of algorithms and their representations and storage of data. In order to support the operation of independent autonomous objects, processors must be adapted to both the inference and learning phases of the different algorithms. Finally the processor must also meet the strict throughput and latency specifications of the targeted

application areas. It will have to achieve very high energy efficiency so that they can be used in objects powered by batteries.

Research objectives

The project will pursue the following research objectives.

- Prepare a review of the state of the art relevant to the project.
- Propose flexible architectures allowing the implementation of more than one class or subclass of AA algorithms, emphasizing throughput or energy consumption.
- Propose hierarchies and memory interfaces suitable for different classes or subclasses of AA algorithms, emphasizing throughput or energy consumption.
- Demonstrate the validity of the proposed architectures by implementing them in relevant technologies (ASIC, FPGA, CGRA, ASIP, etc.).
- Measure the performance and energy consumption of architectures developed in real use contexts. Compare the results to the state of the art.

Methodology

- Literature review and analysis of existing architectures.
    - Draw up a list of classes of IoT applications likely to benefit from the design of a flexible computing architecture.
    - Make a list of the types of AI and ML algorithms that can be implemented in interconnected objects. For each type of algorithm, determine the calculation operators and memory access patterns. Determine the invariants for their implementation.
    - Draw up a list of the types of interconnected objects according to their location in the cloud (center, periphery, embedded object, etc.) and describe their characteristics.
    - Draw up an inventory of the state of the art of the architectures presented to implement different classes of ML algorithms (see among others Mourshed 2022, the references cited by Ahmadi 2021, Jouppi 2018, Luo 2017, Chen 2016, Du 2015 , etc.).
    - Create an inventory of flexible processor architectures that can effectively implement algorithms from several different classes, in machine learning or other categories.
    - Writing a literature review.
- Modeling and implementation of selected existing architectures and reproduction of results.
- Proposal of new architectures that can implement the calculations of two, three, four or more different classes of algorithms. Exploitation, among other things, of the concepts of parallelism, pipeline and systolic networks, and emphasis on innovative solutions for interconnections between data paths and memories. Modeling of architectures and estimation of performance and costs.
- Description of new architectures using data flow diagrams and hardware description languages. Verification of systems by simulation.
- Synthesis and implementation in different technologies (ASIC, FPGA, CGRA, ASIP, etc.) and extraction of the performances achieved.
- Comparison of flexible architectures with traditional architectures.
- Writing articles.

Timeline

- October 2024 to March 2025: analysis of existing architectures and writing of a literature review
- March 2025 to February 2026: proposal for new architectures
- March 2026 to February 2027: evaluation of architectures
- March to September 2027: writing of the thesis and articles

Bibliographie

[1] M. Traore, J. M. Pierre Langlois, et J. Pierre David, « ASIP Accelerator for LUT-based Neural Networks Inference », in 2022 20th IEEE Interregional NEWCAS Conference (NEWCAS), juin 2022, p. 524-528. doi: 10.1109/NEWCAS52662.2022.9842211.

[2] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, et F. Hussain, « Machine Learning at the Network Edge: A Survey », ACM Comput. Surv., vol. 54, nº 8, p. 1-37, nov. 2022, doi: 10.1145/3469029.

[3] M. Ahmadi, S. Vakili, et J. M. P. Langlois, « CARLA: A Convolution Accelerator With a Reconfigurable and Low-Energy Architecture », IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, nº 8, p. 3184-3196, aug. 2021, doi: 10.1109/TCSI.2021.3066967.

[4] M. Ahmadi, S. Vakili, et J. M. P. Langlois, « Heterogeneous Distributed SRAM Configuration for Energy-Efficient Deep CNN Accelerators », in 2020 18th IEEE International New Circuits and Systems Conference (NEWCAS), juin 2020, p. 287-290. doi: 10.1109/NEWCAS49341.2020.9159814.

[5] M. Ahmadi, S. Vakili, et J. M. P. Langlois, « An Energy-Efficient Accelerator Architecture with Serial Accumulation Dataflow for Deep CNNs », in 2020 18th IEEE International New Circuits and Systems Conference (NEWCAS), juin 2020, p. 214-217. doi: 10.1109/NEWCAS49341.2020.9159818.

[6] Y. S. Shao et al., « Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture », in Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, New York, NY, USA, oct. 2019, p. 14-27. doi: 10.1145/3352460.3358302.

[7] I. Palit, Q. Lou, R. Perricone, M. Niemier, et X. S. Hu, « A Uniform Modeling Methodology for Benchmarking DNN Accelerators », in 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), nov. 2019, p. 1-7. doi: 10.1109/ICCAD45719.2019.8942095.

[8] Y.-H. Chen, T.-J. Yang, J. Emer, et V. Sze, « Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices ». arXiv, 20 mai 2019. doi: 10.48550/arXiv.1807.07928.

[9] A. M. Abdelsalam, A. Elsheikh, J.-P. David, et J. M. P. Langlois, « POLYCiNN: Multiclass Binary Inference Engine using Convolutional Decision Forests », in 2019 Conference on Design and Architectures for Signal and Image Processing (DASIP), oct. 2019, p. 13-18. doi: 10.1109/DASIP48288.2019.9049176.

[10] X. Liu, W. Wen, X. Qian, H. Li, et Y. Chen, « Neu-NoC: A high-efficient interconnection network for accelerated neuromorphic systems », in 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), janv. 2018, p. 141-146. doi: 10.1109/ASPDAC.2018.8297296.

[11] N. P. Jouppi, C. Young, N. Patil, et D. Patterson, « A domain-specific architecture for deep neural networks », Commun. ACM, vol. 61, nº 9, p. 50-59, août 2018, doi: 10.1145/3154484.

[12] A. M. Abdelsalam, A. Elsheikh, J.-P. David, et J. M. Pierre Langlois, « POLYBiNN: A Scalable and Efficient Combinatorial Inference Engine for Neural Networks on FPGA », in 2018 Conference on Design and Architectures for Signal and Image Processing (DASIP), oct. 2018, p. 19-24. doi: 10.1109/DASIP.2018.8596871.

[13] A. M. Abdelsalam, F. Boulet, G. Demers, J. M. Pierre Langlois, et F. Cheriet, « An Efficient FPGA-based Overlay Inference Architecture for Fully Connected DNNs », in 2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig), déc. 2018, p. 1-6. doi: 10.1109/RECONFIG.2018.8641735.

[14] T. Luo et al., « DaDianNao: A Neural Network Supercomputer », IEEE Trans. Comput., vol. 66, nº 1, p. 73-88, janv. 2017, doi: 10.1109/TC.2016.2574353.

[15] Y.-H. Chen, T. Krishna, J. S. Emer, et V. Sze, « Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks », IEEE J. Solid-State Circuits, vol. 52, nº 1, p. 127-138, janv. 2017, doi: 10.1109/JSSC.2016.2616357.

[16] Y.-H. Chen, T. Krishna, J. Emer, et V. Sze, « 14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks », in 2016 IEEE International Solid-State Circuits Conference (ISSCC), janv. 2016, p. 262-263. doi: 10.1109/ISSCC.2016.7418007.

[17] Z. Du et al., « ShiDianNao: Shifting vision processing closer to the sensor », in 2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), juin 2015, p. 92-104. doi: 10.1145/2749469.2750389.

[18] J. Zheng, Y. Liu, X. Liu, L. Liang, D. Chen, et K.-T. Cheng, « ReAAP: A Reconfigurable and Algorithm-Oriented Array Processor With Compiler-Architecture Co-Design », IEEE Transactions on Computers, vol. 71, nº 12, p. 3088-3100, déc. 2022, doi: 10.1109/TC.2022.3213177.