Artificial Resilience in neuromorphic systems

Artificial Resilience in neuromorphic systems

ALESSIO CARPEGNA, Politecnico di Torino, Italy
STEFANO DI CARLO, Politecnico di Torino, Italy
ALESSANDRO SAVINO, Politecnico di Torino, Italy

Biological beings are intrinsically resilient. This means that they are able to continue to perform a task even if they are partially damaged or if some parts of them don't work as expected. This is true also for the human brain. The research in these last years, however, has been concentrated on Artificial Intelligence (AI), to try to emulate the capabilities of the brain to improve itself, learning from experience. Artificial Resilience (AR) is something not explored in detail yet. This four pages abstract present a Ph.D. path dedicated to the extensive study of Artificial Resilience in all its aspects. The study will target neuromorphic systems, in particular Spiking Neural Networks, an emerging type of neural network models that try to mimic the behavior of a biological brain in a faithful way. In addition to this they are in general more suitable for an hardware acceleration. The goal of the Ph.D. is to realize a complete neuromorphic accelerator, configurable and resilient, and to apply it to improve the resilience of other electronic systems. Such an accelerator will be able to target area- and power-constrained applications in mission-critical environments, providing a more efficient alternative to classical techniques like Error Correction Codes (ECC) or redundancy to improve the robustness of a complex electronic system.

CCS Concepts: • **Resilience**; • **Spiking Neural Networks**; • **Hardware accelerators**; • **Fault tolerance**; • **FPGA**;

Additional Key Words and Phrases: mission critical, transient errors, permanent errors, RISC V

1 INTRODUCTION

Artificial intelligence (AI) is one of the most active research fields nowadays. Its goal is to create systems that can learn from their experience and from the external world, to solve complex tasks without direct human supervision. The most obvious reference for such a system is the human brain itself. The most promising technology to achieve this goal at the moment seems to be the Artificial Neural Networks (ANN). They take inspiration from the organization and behavior of neurons observed in a biological neural network. To target real-world applications, however, the similarity between a real brain and an ANN is generally shallow, and the model of the neuron itself is only vaguely inspired by its biological counterpart. All the main models of ANN belong to this category: Convolutional Neural Networks (CNN), able to exceed human performance in image recognition, or Recurrent Neural Networks (RNN), suitable to analyze time series.

Spiking Neural Networks (SNNs) are an emerging type of artificial neural networks [6], born to mimic the behavior of a biological brain more authentically. First, the information between neurons is exchanged as binary spikes, thus minimizing resources to link neurons in the network. Second, the internal model of the neuron, which describes the evolution of its state concerning the incoming spikes, is inspired by what can be observed in biology. Finally, time is treated as an additional dimension in the input and the model itself. Thanks to all these characteristics, SNNs present many advantages compared to more classical models. In particular, the spikes' binary nature reduces the neuron's complexity. It makes SNNs in general more suitable to an implementation on a dedicated hardware accelerator [1]. Additionally, the spikes propagated from the input and the whole network are generally sparse. This allows performing

Authors' addresses: Alessio Carpegna, alessio.carpegna@polito.it, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, Italy, 10129; Stefano Di Carlo, stefano.dicarlo@polito.it, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, Italy, 10129; Alessandro Savino, alessandro.savino@polito.it, Politecnico di Torino, Corso Duca degli Abruzzi, 24, Turin, Italy, 10129.

the elaboration in an event-driven fashion, updating the neuron's state only when a spike is received in input, avoiding any computation otherwise, and reducing the power consumption to a minimum.

The second characteristic of biological systems is their resilience. They can continue the required operation when some parts are malfunctioning or missing. This is a characteristic that is also present in the brain itself. If one part is damaged, internal paths are adjusted to bypass it or correct the wrong information. This involves a learning phase, in which the brain gradually understands how it can work without one fundamental part and how to compensate for it. If the intelligence of biological creatures can be emulated, leading to AI, the same can be potentially done with its resilience, creating Artificial Resilience (AR).

2 ARTIFICIAL RESILIENCE

Nowadays, there are studies on the so-called fault tolerance of ANN models [7]. They are generally based on injecting faults inside the network and observing how it responds, i.e., how its accuracy, performance, and power consumption are affected. However, there are a few missing points in such an analysis. First, most of the studies target classical ANN, and only recently some attention has been dedicated to SNN [4, 9, 11]. Second, the analysis is often performed at a high level of abstraction, using software simulations of the ANN models. However, the current trend is to move computations to dedicated hardware accelerators to speed up the execution and make neural networks suitable for the area and power-constrained application, e.g., on IoT nodes or embedded systems. The current trend of moving from centralized Cloud Computing towards more efficient Edge Computing makes this even more apparent. So, an extended analysis of the real impact that faults can have on the hardware implementation of neural networks is required. Finally, there is a second perspective still not, or only sketchily, explored: as said before, resilience is learned by the brain, so it would be interesting to study a possible application of ANNs to make a general electronic system more resilient. In this sense, the network can be trained to recognize the presence of a fault and to correct it, for example, bypassing the part that is not working or compensating the errors to continue to provide a correct output.

The presented Ph.D. path aims to explore Artificial Resilience completely and apply it in particular to neuromorphic systems. The goal is to create a resilient accelerator for an SNN, fully configurable, to target a generic task and then exploit it in different electronic systems to make them resilient. This can facilitate the application of SNNs, particularly suitable to be accelerated through dedicated hardware, to mission-critical applications, to make them able to work in such harsh conditions and to use them to improve the robustness and fault tolerance of other systems. Such a solution can represent a valuable alternative to current fault tolerance approaches which rely on redundancy and Error Correction Coding (ECC) to guarantee resilience [2]. Figure 1 graphically depicts the main steps required in the design of such a structure.

3 DESIGN STEPS

The first part of the Ph.D. will be dedicated to creating a complete and flexible hardware accelerator for SNNs (Figure 1, STEP 1). The goal is to obtain a fully configurable structure to target many different applications. In this way, the resilience problem can be studied in various scenarios, and the architecture can be modified to improve its robustness.

Since the accelerator's design has many degrees of freedom, this first part aims to develop a framework to create it following the user's requests automatically. The framework will be open source and developed using python language. The accelerator will be designed using VHDL. Different neuron models, network structures, learning methods, input encoding, and output decoding techniques can be easily explored. The framework will be able to address both offline (on a software version of the model) and online (directly on the accelerator) training. In this way, it will be possible to

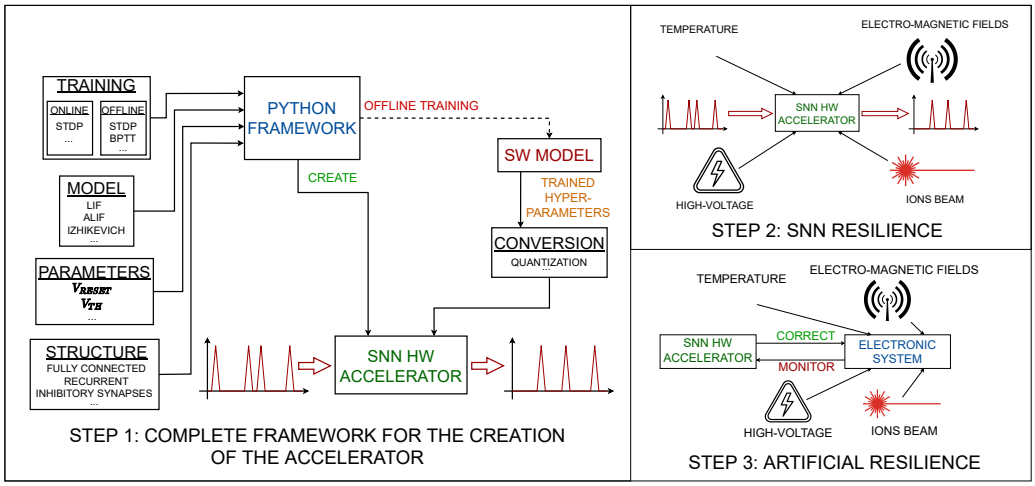

Fig. 1. Block diagram of the Ph.D. work

evaluate if the network can learn resilience while running. In literature, there are many different accelerators dedicated to a huge variety of tasks. However, there is still the need for a generalized approach that gives the possibility to configure the accelerator depending on the target application, guiding in the choice of the internal parameters and hyper-parameters, to select the design that best fits the specific task.

Once a complete architecture is available, a comprehensive study on its resilience can be performed (Figure 1, STEP 2). As Figure 1 (STEP 1) shows, a software simulation can be created in parallel to the accelerator to train the model offline or even to compare the results obtained by the two different implementations. This allows the possibility of performing fault injection and analyzing the effects at different levels of abstraction, evaluating the accuracy differences in the two cases. A first analysis step will be performed at a software level for an easier and faster evaluation. Secondly, a more detailed study will be conducted on the hardware accelerator itself, analyzing the impact of faults with different network characteristics. All the different kinds of internal faults will be evaluated including permanent (e.g., stack-at faults), intermittent, and transient (e.g., bit flips) faults. Finally, when all the main criticalities have been studied, countermeasures to improve the robustness of the accelerator will be proposed to make the architecture more resilient.

At this point, the obtained resilient accelerator can be used to make other electronic systems resilient to errors as well (Figure 1, STEP 3). The same sources of errors will be evaluated, but the accelerator will be used to monitor the target system, detect errors and correct them. For example, if the network is trained using the correct results provided by a specific component, it can be used to compensate its output in case it is wrong due to an internal fault. Or, as an alternative, a malfunctioning part can be turned off, bypassed, or substituted by other similar components, taking into account a possible reduction in the performance or accuracy. There are many techniques to explore, while the purpose is the same for all of them: to make the system able to conclude its task in the presence of internal errors, being them transient or permanent. The study will be conducted on state-of-the-art electronic systems targeting RISC-V architectures. Again a first step will consist in applying all the different methods at a simulation level, using modern computer architecture simulators. When the problem is thoroughly addressed and efficient methods found, a more detailed analysis will be conducted directly on the hardware, for example, using a RISC-V RTL description and synthesizing it, together with the designed accelerator, onto an evaluation platform, like an FPGA.

4 CURRENT WORK

Currently, the proposed Ph.D. project is undergoing the first phases of the first step with the design of Spiker, an FPGA-optimized hardware accelerator for SNN. The goal in this phase was to explore a simple architecture, trying to minimize the required resources and to set a reference upper bound to performance, exploiting the largest available parallelism. The accelerator has been tested over the MNIST dataset and compared with other reference designs in literature [5, 8, 10]. It out-performs all of them, with a classification time around $200\mu s/image$ with a clock frequency of 100MHz, around one order of magnitude smaller concerning the fastest design, keeping the energy consumption fully comparable to the most optimized solution, with an average of $13mJ/image$. Currently Spiker has been tested with a single layer network architecture that is not optimized for accuracy and can reach around 74% of correct classifications. The next step will consist in evaluating more optimized network configurations. In literature, many of them report an accuracy comparable to state of the art CNN [3], so those will be the starting point for the analysis.

The obtained results are auspicious and set a strong base for a future generalization of the structure and for the creation of the complete framework.

REFERENCES

[1] Maurizio Capra, Beatrice Bussolino, Alberto Marchisio, Guido Masera, Maurizio Martina, and Muhammad Shafique. 2020. Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead. *IEEE Access* 8 (2020), 225134–225180.

[2] Simone Dutto, Alessandro Savino, and Stefano Di Carlo. 2021. Exploring Deep Learning for In-Field Fault Detection in Microprocessors. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*. 1456–1459.

[3] Pierre Falez, Pierre Tirilly, Ioan Marius Bilasco, Philippe Devienne, and Pierre Boulet. 2019. Multi-layered Spiking Neural Network with Target Timestamp Threshold Adaptation and STDP. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[4] Anju P. Johnson, Junxiu Liu, Alan G. Millard, Shvan Karim, Andy M. Tyrrell, Jim Harkin, Jon Timmis, Liam J. McDaid, and David M. Halliday. 2018. Homeostatic Fault Tolerance in Spiking Neural Networks: A Dynamic Hardware Perspective. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 2 (2018), 687–699.

[5] De Ma, Juncheng Shen, Zonghua Gu, Ming Zhang, Xiaolei Zhu, Xiaoqiang Xu, Qi Xu, Yangjing Shen, and Gang Pan. 2017. Darwin: A neuromorphic hardware co-processor based on spiking neural networks. *Journal of systems architecture* 77 (2017), 43–51.

[6] Wolfgang Maass. 1997. Networks of spiking neurons: The third generation of neural network models. *Neural networks* 10, 9 (1997), 1659–1671.

[7] Sparsh Mittal. 2020. A survey on modeling and improving reliability of DNN algorithms and accelerators. *Journal of systems architecture* 104 (2020), 101689.

[8] Daniel Neil and Shih-Chii Liu. 2014. Minitaur, an Event-Driven FPGA-Based Spiking Network Accelerator. *IEEE transactions on very large scale integration (VLSI) systems* 22, 12 (2014), 2621–2628.

[9] Theofilos Spyrou, Sarah A. El-Sayed, Engin Afacan, Luis A. Camuñas-Mesa, Bernabé Linares-Barranco, and Haralampos-G. Stratigopoulos. 2021. Neuron Fault Tolerance in Spiking Neural Networks. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*. 743–748.

[10] Qian Wang, Youjie Li, Botang Shao, Siddhartha Dey, and Peng Li. 2017. Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA. *Neurocomputing (Amsterdam)* 221 (2017), 146–158.

[11] Rachmad Vidya Wicaksana Putra, Muhammad Abdullah Hanif, and Muhammad Shafique. 2021. ReSpawn: Energy-Efficient Fault-Tolerance for Spiking Neural Networks considering Unreliable Memories. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. 1–9.