PhD position

Runtime verification of neural networks in embedded systems

Key words: Neural Network Robustness, Hardware Fault Detection, Runtime Verification, Functional Safety, Resource Optimization

1 Scientific Context

Classification algorithms based on machine learning are now widely used in embedded control systems. They allow either the inference of changes in the system's environment, in which case their output is used as input for control laws, or the direct inference of a control law. In both cases, if they produce an erroneous result, this can lead to a system failure, the consequences of which can be more or less catastrophic for its environment. It is therefore important to consider the robustness of these algorithms, or more precisely, the robustness of their implementation in an embedded system.

In this work, we will focus on their robustness in the presence of accidental hardware faults. Such faults can have various causes (radiation, electrical problems, etc.). They disrupt the operation of the system's electronics. From there, through propagation, these faults can impact the calculations necessary for performing an inference, and, for some, ultimately lead to a failure of the embedded system. To understand what these faults are, and how they can be detected, several elements must be taken into account.

2 Problems and Scientific Objectives

Neural networks exhibit intrinsic robustness linked, on the one hand, to a form of information redundancy in their structure and operation, and on the other hand, to the masking or amplitude attenuation performed by certain layers [1]. Thus, the systematic detection of faults at a very fine level, i.e., at the level of elementary operations of the architecture (arithmetic operators, memory transfers, etc.), in addition to a potentially very high cost, leads to a high false positive rate. It is therefore a solution that does not seem, a priori, well-suited. It should be noted, however, that the techniques used to reduce the size and resource consumption of neural networks to make them compatible with the constraints of embedded systems may lead to a partial revision of this assessment.

Furthermore, the output of a neural network is difficult to predict.

Thus, coarse-grained verification, i.e., at the input/output level, does not appear to be the ideal solution either, especially with the limited resources of an embedded system.

In this work, we propose to study the different possible granular levels for runtime verification of neural networks in the presence of hardware faults in embedded control systems. The work plan is as follows:

- Review of necessary concepts (neural networks, microarchitecture, dependability)
- Literature review on error detection and the robustness of neural networks
- Proposal of detection mechanisms, prototyping, and experimental characterization

3 Candidate Profile

The candidate must demonstrate an interest in the following topics and proven skills in at least some of them:

- Neural Networks and Embedded AI
  - Good understanding of neural network architectures (CNN, RNN, Transformers, etc.) and their implementation constraints in embedded systems.
  - Experience with model optimization techniques (quantification, pruning, knowledge distillation) for resource-constrained environments.
  - Knowledge of frameworks such as TensorFlow Lite, PyTorch Mobile, or ONNX Runtime for embedded deployment.
  - Familiarity with hardware/software co-design approaches for neural networks.
- Hardware Architecture and Fault Tolerance
  - Understanding of hardware faults (transient, permanent) and their impact on digital circuits, particularly in embedded systems.
  - Experience with error detection and correction techniques (error-correcting codes, redundancy, BIST).
  - Knowledge of FPGA prototyping and hardware-in-the-loop validation.
  - Experience with hardware description languages (VHDL, Verilog) and RTL design.
  - Knowledge of real-time operating systems (RTOS) and their integration with AI workloads.
  - Understanding of memory hierarchies (cache, scratchpad memory) and their impact on neural network inference.

4 Thesis Context

This thesis project is part of the Adaptive architectures for embedded artificial intelligence (AdaptING) action of the PEPR-IA program. The AdaptING project lies at the intersection of computer science (computer architectures) and machine learning. The PhD candidate will be fully integrated into the project (discussions, meetings, seminars) and will collaborate with other participating researchers and PhD/Master's students. They will have the opportunity to co-supervise interns. Teaching opportunities (paid in addition to their salary) may be offered if they wish.

Within the AdaptING framework, this project is a collaboration between the ASIC team at IETR and the STR team at LS2N. IETR contributes its expertise in circuit design, while LS2N contributes its expertise in embedded systems verification and programming. As both teams are based in Nantes, there will be frequent interaction. The successful candidate will be based at the IETR campus in Nantes.

- Research Laboratories: IETR (UMR CNRS 6164) and LS2N (UMR CNRS 6004)
- Research Teams: ASIC (Architecture, Systems, Infrastructure and Electronics) and STR (Real-Time Systems)
- University: Nantes University
- Location: IETR/Polytech, Nantes, France
- Expected Start Date: September or October 2026 (Duration: 3 years)

5 How to Apply

Applications must be submitted exclusively via the AMETHIS platform. Applications received outside the AMETHIS process will not be considered.

For further information, please send an email to:

- Sébastien Pillement, email: sebastien.pillement@univ-nantes.fr
- Bastien Deveautour, email: bastien.deveautour@univ-nantes.fr
- Sébastien Faucou, email: sebastien.faucou@univ-nantes.fr
- Mikaël Briday, email: mikael.briday@ec-nantes.fr

Applications will be reviewed as they are received. Interviews will be held after the application period. Applications that do not meet the required profile will not be considered for interviews.

Since integration into the IETR laboratory requires access to a restricted, secure area, we ask that you allow sufficient time for the review of your profile if you wish to apply. This process can take up to two months.

References

[1] Guanpeng Li et al. "Understanding Error Propagation in Deep-Learning Neural Networks' Accelerators and Applications". In: *IEEE Design Test* 42.3 (2025), pp. 7–13. DOI: 10.1109/MDAT.2025.3544537.

[2] Mohammad Amin Hasanpour et al. "EdgeMark: An automation and benchmarking system for embedded artificial intelligence tools". In: *Journal of Systems Architecture* 167 (2025), p. 103488. DOI: https://doi.org/10.1016/j.sysarc.2025.103488.

[3] Y. Liu et al. "Fault-Aware Training for Robust Neural Network Inference on Unreliable Hardware". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)* 43.5 (2024), pp. 1423–1436. DOI: 10.1109/TCAD.2024.3367890. URL: https://doi.org/10.1109/TCAD.2024.3367890.

[4] A. Shafiee et al. "Hardware-Efficient Neural Network Accelerators: Challenges and Opportunities in the Era of Approximate Computing". In: *Journal of Emerging Technologies in Computing Systems (JETC)* 19.3 (2023), pp. 1–20. DOI: 10.1145/3587136. URL: https://doi.org/10.1145/3587136.