ENAC LAB
Interactive Informatics

ANITI

Extending Constraint-Based Neural Network Quantization for FPGA Deployment

dorra.ben-khalifa@enac.fr
ENAC LAB – LII Team

M2 Internship, starting February 2027

1 Context

Deploying deep neural networks (DNNs) on embedded systems requires balancing numerical accuracy, memory usage, computational performance, and energy consumption. Fixed-point arithmetic reduces implementation costs, particularly on hardware platforms with limited floating-point support, but it introduces rounding errors that propagate through the network. Choosing the number of integer and fractional bits of each value, so that these errors remain controlled without wasting bits, is therefore a significant challenge. Previous work introduced the tool POPINNS [2], which takes a TensorFlow model and determines fixed-point formats by constraint solving. The operations of each layer are analyzed to derive how they propagate numerical errors, and these analyses are expressed as constraints on the formats, whose solutions guarantee that the errors stay under control. The tool then generates the corresponding fixed-point C code for embedded systems. This approach currently covers the usual layers of DNNs, such as fully connected and convolutional layers.

Recent neural network architectures introduce operations that are not yet supported by POPINNS, including GELU activations, element-wise additions in residual connections, tensor concatenations, and normalization layers, as well as hardware-oriented replacements for these operations [3]. Extending the tool to these operations is essential to support modern architectures, including residual convolutional networks and Vision Transformers. In addition, extending the constraint-based approach to larger models raises challenges in terms of scalability and computational efficiency. Finally, adapting the code generation workflow to FPGA-based platforms requires translating the optimized fixed-point formats into hardware implementations, while accounting for numerical accuracy, resource utilization, and latency. Addressing these challenges will broaden the applicability of POPINNS to modern neural networks deployed on resource-constrained embedded systems.

2 Objectives

The first objective of the internship is to extend the existing tool to support neural network operations that are currently missing. This includes activation functions such as LeakyReLU, Tanh, HardTanh, Sigmoid, and GELU, as well as additional operations such as element-wise addition, matrix multiplication, and tensor concatenation [3]. Particular attention will be paid to the theoretical analysis of these operations and their sensitivity to fixed-point quantization errors, in order to integrate them into the existing constraint-based approach.

The second objective is to investigate the applicability of the extended prototype to more complex neural network architectures, including convolutional networks with residual connections and Vision Transformers. This will involve assessing the scalability of the constraint-solving approach and identifying the limitations encountered when handling larger models.

Finally, the internship will explore the extension of the code generation workflow to FPGA-based embedded systems. The generated fixed-point implementations will be evaluated in terms of numerical accuracy, hardware resource utilization, and latency. The deployment workflow may rely on High-Level Synthesis (HLS) tools, such as Vitis HLS, or direct hardware code generation using VHDL or Verilog.

The approach will be evaluated on selected case studies, potentially including ACAS Xu for aircraft collision avoidance and a vision-based neural network trained on the LARD runway detection dataset[1]. The final choice of architectures and experiments will depend on the progress of the prototype extensions and the available hardware resources. Formal verification of relevant functional properties may also be investigated where feasible [1, 4].

3 Required Skills

The candidate should have a good knowledge of embedded AI and neural network architectures, and programming skills in C/C++ and Python. Knowledge of embedded systems and FPGA design, including hardware description languages such as VHDL or Verilog, is expected. Familiarity with formal methods is also required. Interest in numerical precision, floating-point and fixed-point arithmetic, and quantization would be valuable.

4 Application

To apply, please send an email to dorra.ben-khalifa@enac.fr including the following documents (in a single PDF or a zip archive):

- Your CV;
- A cover letter describing your motivation and your interest in the topic;
- Your transcripts for the M1 and, if available, the first M2 results;
- One or more samples of your technical or scientific work, such as a project report, an internship report, or a publication;
- The names and contact details of one or two academic or professional references (recommendation letters are welcome but optional);
- Any additional material you consider relevant to your application, such as a GitHub repository or a personal project.

Deadline: applications must be received by **December 05, 2026**.

References

[1] Wahiba Bachiri, Yassamine Seladji, and Pierre-Loïc Garoche. Formal specification and SMT verification of quantized neural network for autonomous vehicles. *Sci. Comput. Program.*, 245:103316, 2025.

[2] Dorra Ben Khalifa and Matthieu Martel. Efficient implementation of neural networks usual layers on fixed-point architectures. In *Proceedings of the 25th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems, LCTES*, pages 12–22. ACM, 2024.

[1] `https://github.com/deel-ai/LARD`

[3] Lubin Gauthier, Francis Bony, and Julien Perchoux. hardyt: A post-training hardware-aware layernorm replacement. In *IEEE International Symposium on Circuits and Systems, ISCAS*, pages 3831–3835. IEEE, 2026.

[4] Haoze Wu, Omri Isac, Aleksandar Zeljic, Teruhiro Tagomori, Matthew L. Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark W. Barrett. Marabou 2.0: A versatile formal analyzer of neural networks. In *Computer Aided Verification - 36th International Conference, CAV, Proceedings, Part II*, volume 14682 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 2024.