Contribution ID: **322** Type: **Presentazione orale**

Custom FPGA Implementation of Neural Networks for Accelerated and Fully Controllable AI Processing in Medical Applications

Thursday, 29 May 2025 18:10 (20 minutes)

Neural Networks (NNs) are widely employed in tasks such as feature extraction, classification, segmentation, and reconstruction of quantitative MRI maps, particularly in scenarios lacking an analytical model, resulting in a reconstruction process that is computationally intensive and time-consuming. The versatility and ability of NNs to be trained on ground-truth datasets are particularly appealing in medical applications, where they can accelerate analysis and minimize human intervention.

While they can achieve high efficiency, a major obstacle to NN application is the computational resources required to run them. Fully Connected NNs (FCNs) and Convolutional NNs (CNNs) architectures need a great number of images for training, validation, and testing, with computational complexity scaling exponentially with the number of parameters due to the "curse of dimensionality". This often results in computational costs that can extend processing times to hours or days.

Field Programmable Gate Arrays (FPGAs) are programmable circuits, offering a valued alternative due to their high throughput, low latency, and inherent programmability. While providing high-level synthesis software tools for rapid software-to-firmware conversion, they also offer a great solution to overcome the computational weight of NN training and running. Implementing NNs on FPGAs, after a traditional software validation of an AI algorithm, involves the replication of NN core functionalities at the firmware level, manually mimicking the basic functions and operations performed within NN layers, aiming at accelerating processing by a factor of a few times to hundreds.

While high-level synthesis tools simplify FPGA implementation, they often limit customization and optimization. In contrast, FPGAs can be fully customized to exploit their complete acceleration potential, allowing for designs that are both highly efficient and targeted to specific application requirements.

This work takes a radically different approach, presenting a completely custom FPGA implementation of NN architectures that intentionally avoids reliance on high-level synthesis. Instead, every component is developed in VHDL, providing unparalleled control over the design, precision, and resource usage. This work focuses on the most used NN architectures, each implemented as a standalone module. Each module, implemented as reusable VHDL building blocks, supports full customization and manual optimization of key numerical operations, including quantization in fixed-point representation. Unlike existing FPGA NN tools, which prioritize ease of use at the cost of flexibility, our approach ensures complete transparency and adaptability in every design aspect.

We demonstrate the approach on the ALVEO U250 FPGA, which offers 1.7M Look-Up Tables (LUTs), 3.4M Flip-Flops (FFs), 12k Digital Signal Processors (DSPs), and 2.6k BRAMs. Key modules developed include Synaptic sum with ReLU activation, Convolution, MaxPooling, Padding, Dense layers, rescaling, Upsample, and Concatenate. Each module is validated against software counterparts to ensure precise numerical equivalence. Internal FPGA memory is developed to store hyperparameters, minimizing PCIe transfers and further improving execution times.

In the Fully Connected NN (FCN) example, the synaptic sum was implemented generically to serve all nodes, with layers linked through parameterizable VHDL instances. Hardware testing on a 6-layer FCN with 498 nodes demonstrated agreement with software results for 5000 inference samples, with computation times reduced from tens of μs in software to 280 ns on FPGA at 200 MHz.

For CNNs, a simplified 2-layer network (Convolution + ReLU, MaxPooling, Dense) achieved a total delay of 4.06 ns at 100 MHz, with the first result produced after 130 ns. Compared to Python implementations using TensorFlow/Keras or manually quantized software, our firmware achieved a speedup ranging from 3 to 6 orders of magnitude, with numerical discrepancies consistently below 0.1%.

On a parallel track, we are also developing the training process of the FCN directly on the FPGA. This involves developing custom VHDL modules for backpropagation, gradient computation, and weights updates, ensuring compatibility with the fixed-point quantization used during inference. The aim is to achieve a fully hardware-integrated workflow that minimizes data transfer between the FPGA and external systems, thus significantly reducing training time while maintaining precision. Preliminary tests suggest a total training time of 200 seconds, significantly faster than standard CPU-based training, which can be up to 250 times slower.

This custom FPGA implementation demonstrates the potential for NN acceleration using low-level, fully targeted firmware. Unlike high-level synthesis approaches, it provides precise optimization of hardware resources, dataflow, and numerical precision. By extending the framework to support both inference and training directly on hardware, this approach offers a comprehensive and highly efficient solution for applications that demand maximum performance and full control over design parameters.

Bibliography

- Gore, J.C.: Artificial intelligence in medical imaging. Magnetic Resonance Imaging 68, 1–4 (2020) https://doi.org/10.1016/j.mri.2019.12.006
- Barbieri, M., Brizi, L., Giampieri, E., Solera, F., Manners, D., Castellani, G., Testa, C., Remondini, D.: A deep learning approach for magnetic resonance fingerprinting: Scaling capabilities and good training practices investigated by simulations. Physica Medica 89, 80–92 (2021) https://doi.org/10.1016/j.ejmp.2021.07.013
- Barbieri, M., Brizi, L., Giampieri, E., Solera, F., Castellani, G., Testa, C., Remondini, D.: Circumventing the Curse of Dimensionality in Magnetic Resonance Fingerprinting Through a Deep Learning Approach. NMR Biomed (2022) https://doi.org/10.1002/nbm.4670
- Sanaullah, A., Yang, C., Alexeev, Y., Yoshii, K., Herbordt, M.: Real-time data analysis for medical diagnosis using fpga-accelerated neural networks. BMC Bioinformatics 19 (2018) https://doi.org/10.1186/s12859-018-2505-7
- Xiong, S., Wu, G., Fan, X., Feng, X., Huang, Z., Cao, W., Zhou, X., Ding, S., Yu, J., Wang, L., Shi, Z.: Mri-based brain tumor segmentation using fpga-accelerated neural network. BMC bioinformatics 22(1), 421 (2021) https://doi.org/10.1186/s12859-021-04347-6
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference (2017). https://arxiv.org/abs/1712.05877
- Grossi, M., Alfonsi, F., Prandini, M., Gabrielli, A.: A high throughput intrusion detection system (ids) to enhance the security of data transmission among research centers. Journal of Instrumentation 18(12), 12017 (2023) https://doi.org/10.1088/1748-0221/18/12/C12017

Primary authors: Mr RICCHI, Mattia (Istituto Nazionale di Fisica Nucleare); ALFONSI, Fabrizio (Istituto Nazionale di Fisica Nucleare); FENDILLO, Lucrezia (Dipartimento di Fisica e Astronomia, Università di Bologna); CASALI, Elena (Dipartimento di Fisica e Astronomia, Università di Bologna); RETICO, Alessandra (Istituto Nazionale di Fisica Nucleare); BRIZI, Leonardo (University of Bologna - Alma Mater Studiorum); GABRIELLI, Alessandro (Istituto Nazionale di Fisica Nucleare); TESTA, Claudia (Istituto Nazionale di Fisica Nucleare)

Presenter: Mr RICCHI, Mattia (Istituto Nazionale di Fisica Nucleare)

Session Classification: Technology Tracking

Track Classification: Technology tracking