Hardware acceleration for fast MRF map reconstruction: FPGA porting of a deep learning algorithm

Mattia Ricchi[1,2], Camilla Marella[3], Fabrizio Alfonsi[2], Marco Barbieri[4], Alessandra Retico[1,5], Alessandro Gabrielli[2,3], Leonardo Brizi[3], Claudia Testa[2,3].

[1] *Department of Computer Sciences, University of Pisa, IT*
[2] *INFN, Bologna, IT*
[3] *Department of Physics and Astronomy, University of Bologna, IT*
[4] *Department of Radiology, Stanford University, Stanford, CA, United States*
[5] *INFN, Pisa, IT*

Introduction

A hardware acceleration for Magnetic Resonance Fingerprinting (MRF)[1] maps reconstruction from clinical MRI is presented.
MRF revolutionizes MRI by quickly generating multi-parametric maps from a single scan. Traditionally, signal matching with simulated dictionaries posed challenges due to computational limitations. Barbieri et al.[2,3] proposed a Neural Network (NN) approach to address this, yet its training demands significant resources and time. Recently, using Field Programmable Gate Array (FPGA) acceleration for NN algorithms emerged as a solution, promising faster processing and reduced power consumption. This study aims to optimize NN for FPGA acceleration, specifically targeting MR parameter reconstruction (T_1 and T_2) from clinical images. By adapting the NN[2,3] for FPGA compatibility, significant enhancements in processing speed and efficiency are anticipated.

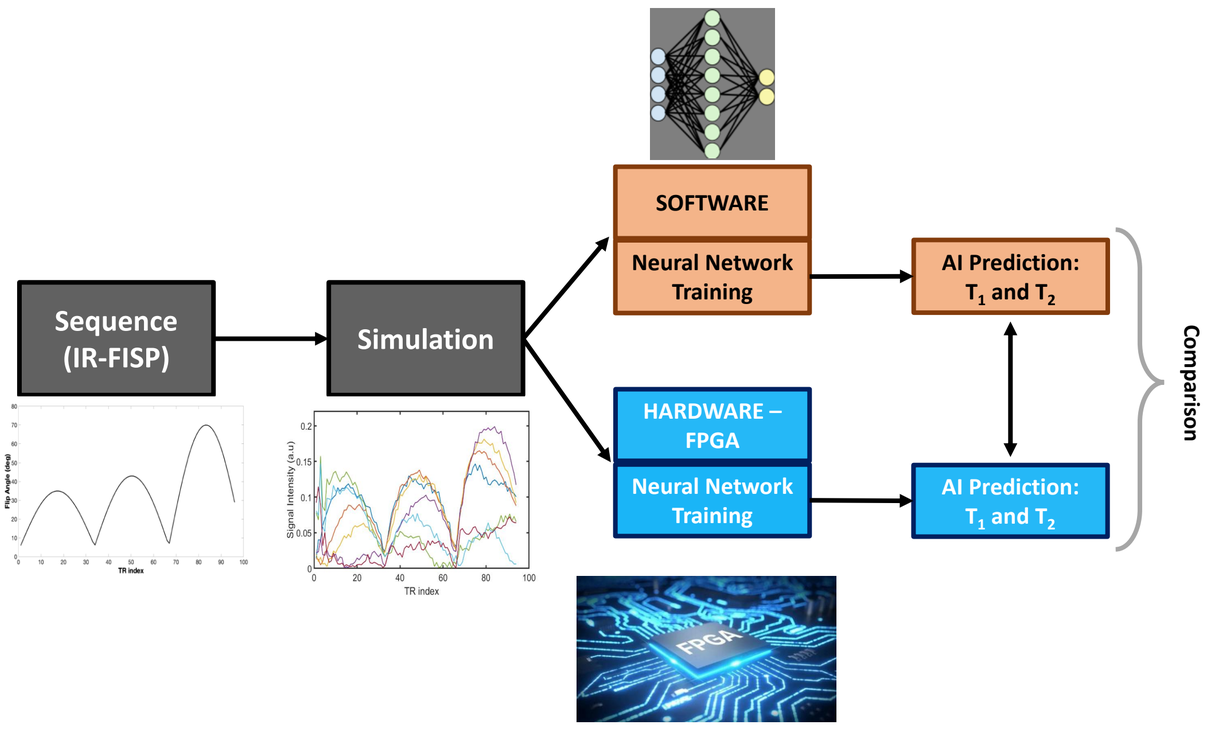

Figure 1: Framework of the proposed method

Materials and Methods

FPGA

An FPGA is a reconfigurable integrated circuit that can be programmed to perform various tasks, offering flexibility, low and fixed latency, power efficiency, and high performance. These characteristics position the FPGA as a promising tool for accelerating NN training. The available FPGA boards are the VCU 1525 and the ALVEO U250.

Original NN

The NN model[2,3] consists of seven fully connected layers and 2032 nodes. The training was supervised, running for 500 epochs with 1000 gradient steps each, maintaining a fixed batch size of 500 and a learning rate of 10^{-4}.

Adapted NN

The first two layers of the Original NN were removed, moving from 2032 to 496 nodes and from 719392 to 40480 parameters. The network is then quantized through Quantization-Aware training to convert all parameters to integers to meet the available resources of the FPGA hardware.

Following conventional software validation, the NN algorithm can be converted into FPGA-compatible hardware, enhancing processing speeds by several to hundreds of times.

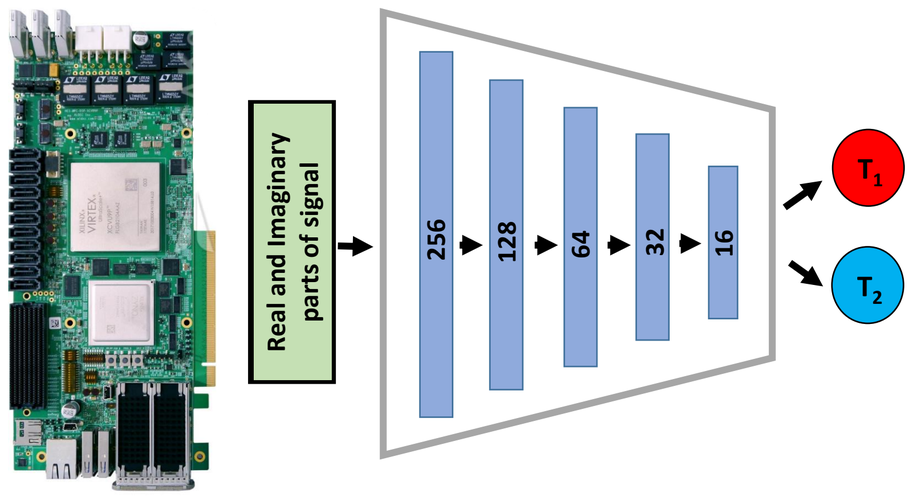

Figure 2: Fully Connected Neural Network design. The Network is a Feed Forward Net of 5 fully connected layers, uses a Rectified Linear Unit (ReLU) as the activation function for the neurons.

FPGA NN implementation

The porting operations from software to firmware for FPGA are at the first stages, developing a generalized single node module and backpropagation algorithm. To balance FPGA memory usage with 0.1 millisecond precision in T_1 and T_2 predictions, weights are represented with 8 bits and other parameters with 18 bits.

A low-level HDL design approach, without high-level synthesis support, was chosen to allow complete control over the structure and ensure data protection by utilizing an on-FPGA firewall security algorithm[4].

Results

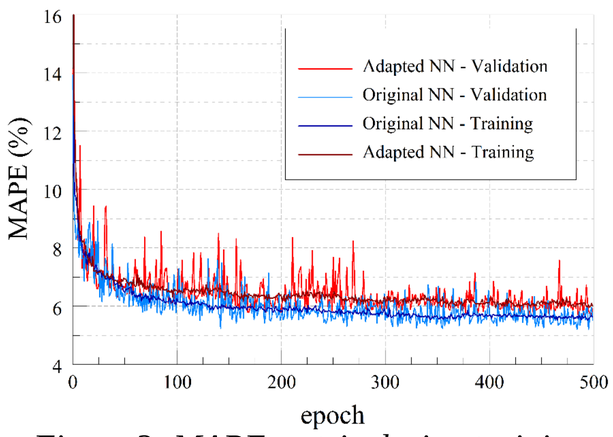

Figure 3: MAPE metric during training for the Original and Adapted NNs

	T_1		T_2	
	Original	Adapted	Original	Adapted
MAPE (%)	2.15	2.36	8.89	11.1
MPE (%)	-0.66	0.12	0.02	-3.12
RMSE (ms)	75	78	145	148

The results show that the Adapted NN performs consistently even after significant resizing to meet FPGA hardware specifications.

An initial resource estimation suggests implementing a portion of a layer on the FPGA and executing it multiple times to cover NN operations. Backpropagation will likewise occur with partial parallelization.

Accounting for input data transmission and processing time, we anticipate a total training time of under 15 minutes (~16 h on CPU).

Discussion and Conclusions

The estimated training time is significantly lower than the time required by standard CPU of a factor up to 60. The ability to train the NN in a few minutes rather than hours could enable the creation of a network trained with user-selected parameters. For instance, if a network is designed for a MRF sequence with specific parameters, but the scanner settings (TR or FA) change, re-training offline and subsequent offline data analysis would be necessary. Training the network with FPGA on the scanner could facilitate real-time reconstruction and visualization of maps online.

e-mail: mattia.ricchi@phd.unipi.it

References

[1] Ma D et al, Nature 2013.
[2] Barbieri M et al, Physica Medica 2021.
[3] Barbieri M et al, NMR in Biomed 2022.
[4] Grossi M et al, Journal of Instrumentation, 2023.