DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Towards the Extension of FPG-AI Toolflow to RNN Deployment on FPGAs for On-board Satellite Applications

Tommaso Pacini, Emilio Rapuano*, Luciano Tuttobene*, Pietro Nannipieri*, Luca Fanucci*, Silvia Moranti†*

**Department of Information Engineering, University of Pisa, Pisa, Italy*

† European Space Research and Technology Centre, European Space Agency (ESA), Noordwijk, The Netherlands

IEEE EDHPC 2023 Conference – Antibes, France

Tommaso Pacini
tommaso.pacini@phd.unipi.it

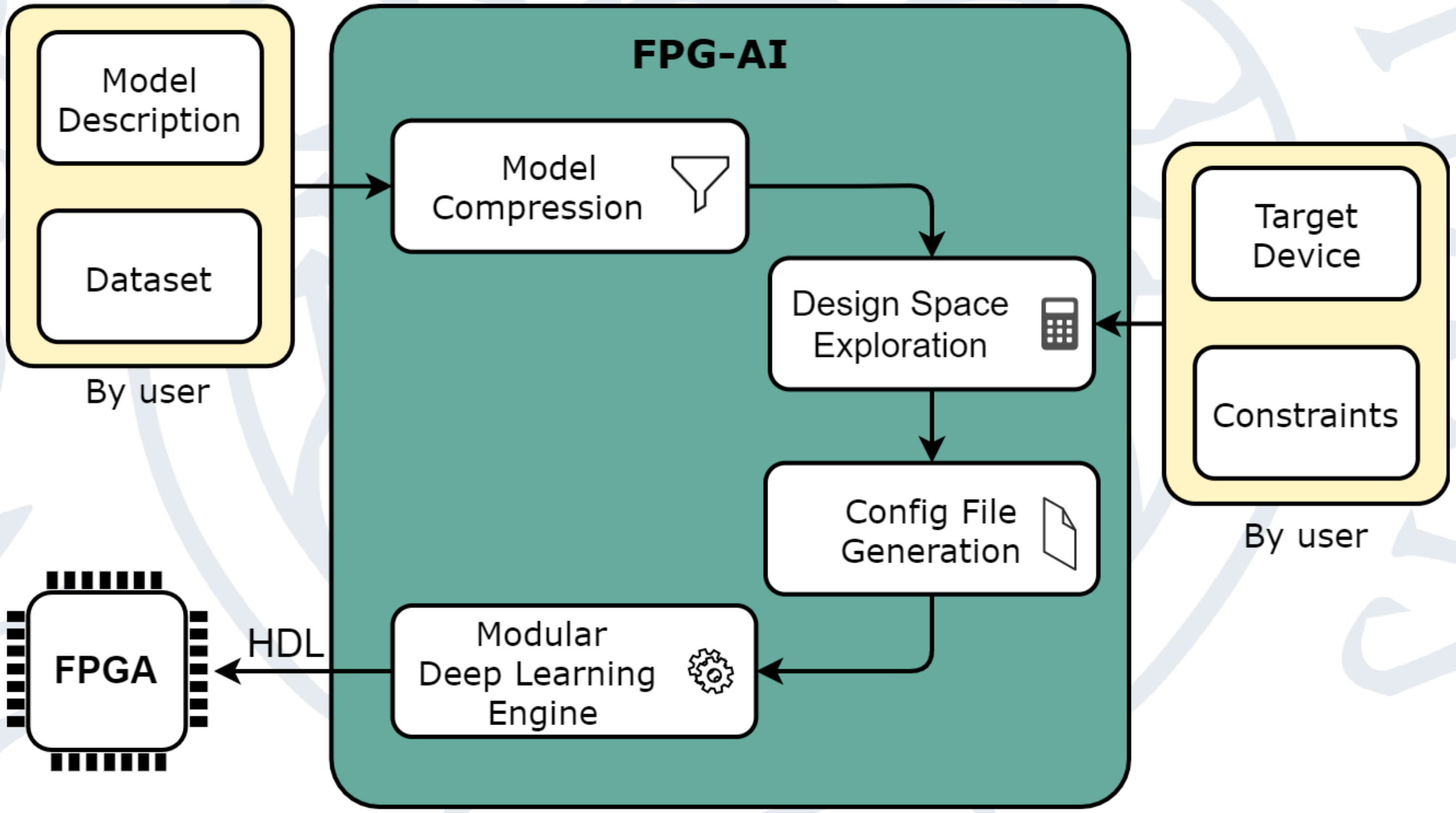
Outline

Outline

1. Background: FPG-AI Toolflow for CNNs
2. Post-training Quantization of RNN Layers
3. HW Acceleration for GRUs
4. Results for FDIR Case Study
5. Conclusions

Background: FPG-AI Toolflow for CNNs

- Automation toolflow for efficient deployment of pre-trained CNN models on FPGA technology [1], [2]



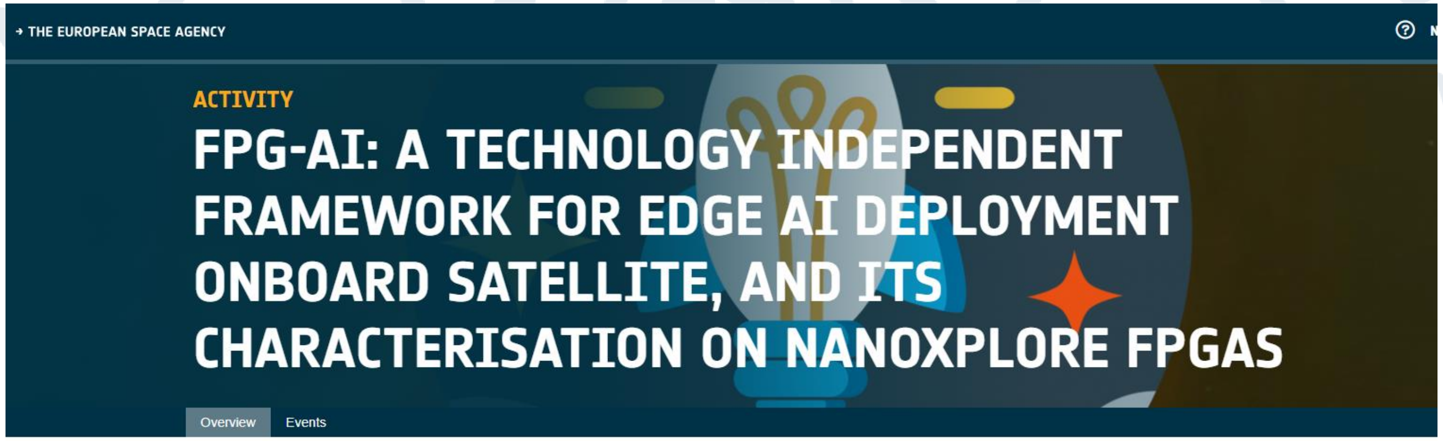
FPG-AI Key Features

- High degree of customization with respects to user's constraints on:
  - Resource consumption (DSP/On-chip memories)
  - Post-quantization application metric deviations
  - Inference time
- Unmatched device portability of the Modular Deep Learning Engine (MDE) thanks to:
  - Absence of third-party IPs
  - High scalability in terms of DSP/On-chip memory usage
  - Fine-grain configurable through a .vhd file

→ Enabling the implementation on FPGAs from different vendors and heterogeneous resource budgets!

FPG-AI: Current Objectives

- Extension of the FPG-AI to Recurrent Neural Networks (RNNs)
- Enabling the support for NanoXplore FPGA devices

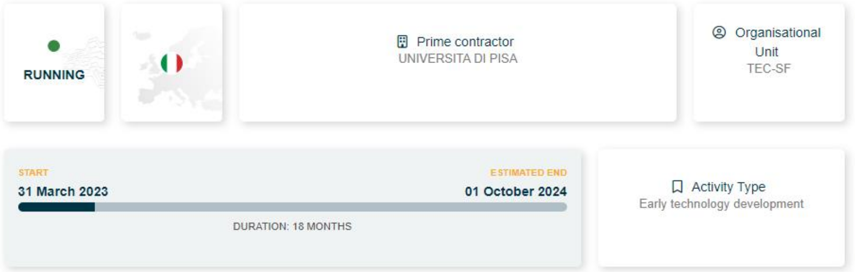


→ THE EUROPEAN SPACE AGENCY

ACTIVITY

FPG-AI: A TECHNOLOGY INDEPENDENT FRAMEWORK FOR EDGE AI DEPLOYMENT ONBOARD SATELLITE, AND ITS CHARACTERISATION ON NANOXPLORE FPGAS

Overview Events



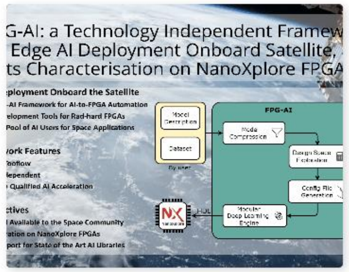
RUNNING

Prime contractor
UNIVERSITA DI PISA

Organisational Unit
TEC-SF

START 31 March 2023
ESTIMATED END 01 October 2024
DURATION: 18 MONTHS

Activity Type
Early technology development



G-AI: a Technology Independent Framew
Edge AI Deployment Onboard Satellite,
ts Characterisation on NanoXplore FPGA

ployment Onboard the Satellite
-AI Framework for AI-to-FPGA Automation
elopment Tools for Rad-hard FPGAs
Pool of AI Users for Space Applications

ork Features
oolflow
ependent
Qualified AI Acceleration

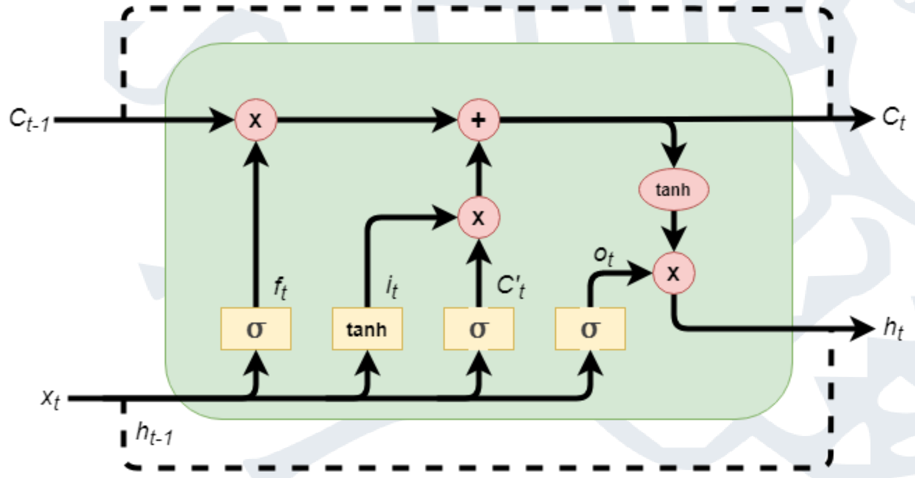
ctives
Available to the Space Community
ation on NanoXplore FPGAs
port for State of the Art AI Libraries

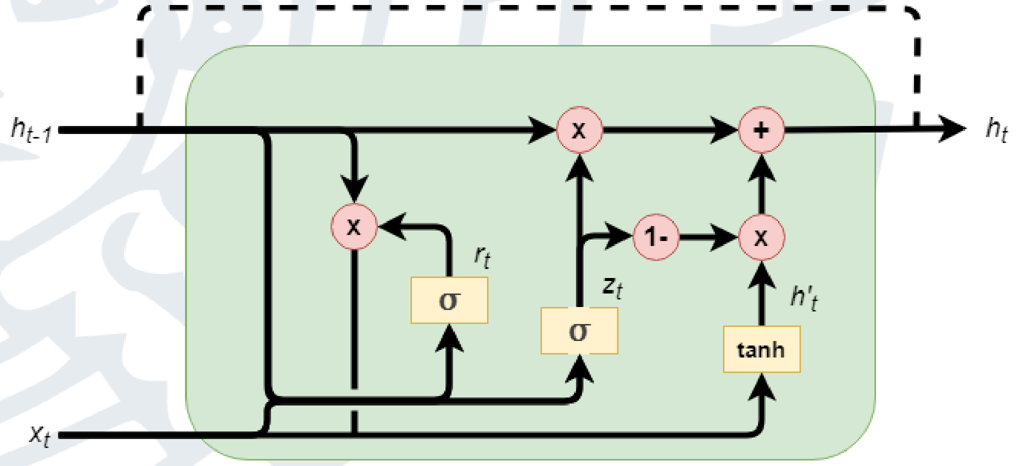
https://activities.esa.int/index.php/4000141108

Outline

Recurrent Neural Networks (RNNs)

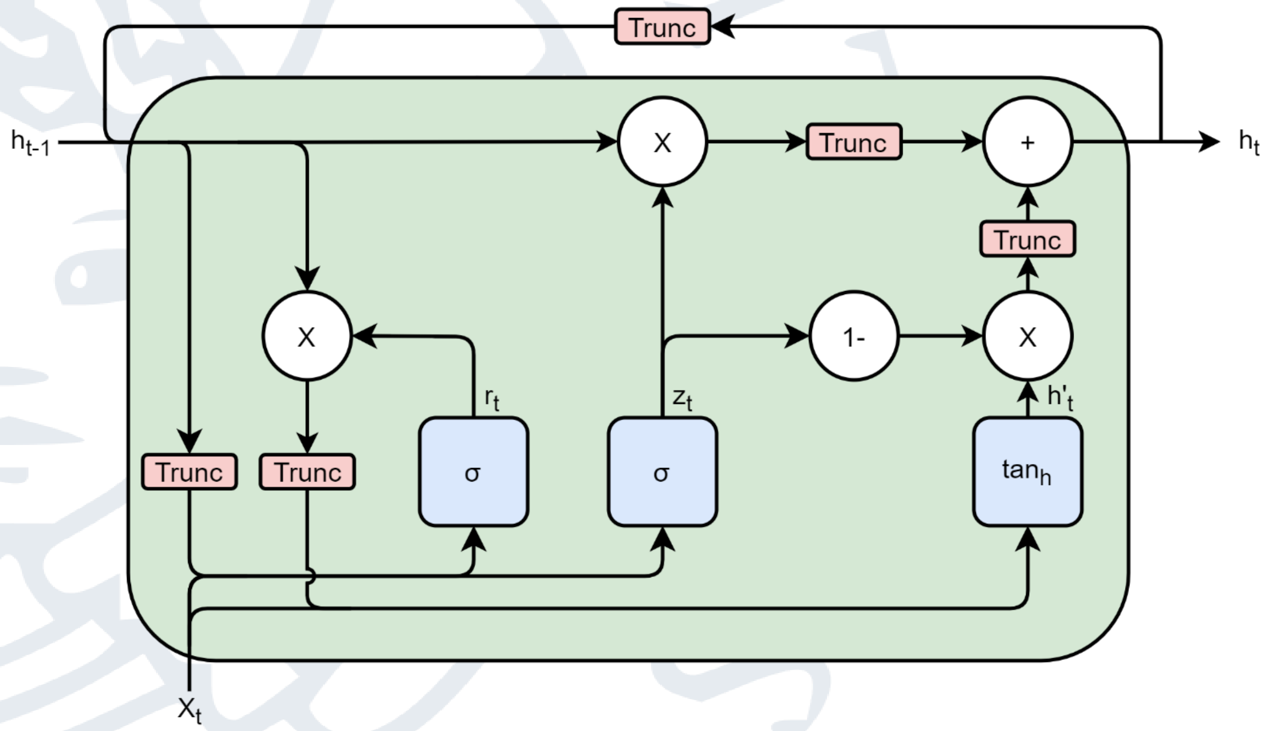
- Commonly used for sequence classification or time series forecasting tasks (e.g. FDIR onboard satellite)
- Exploiting feedback loops to deal with temporal sequences of data
- Most popular architectures:
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)





Model Compression of GRU layers

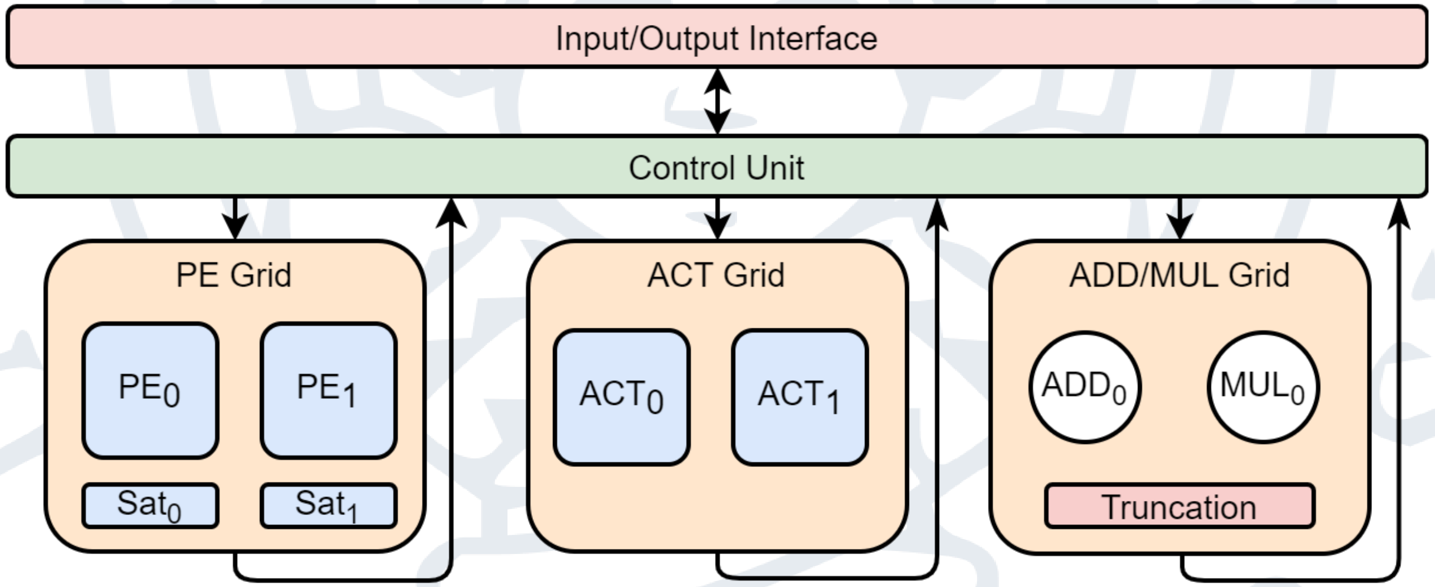
- Post-training quantization algorithm for the quantization of GRU layers [4]
- From **floating-point** to **fully fixed-point** arithmetic
- Degrees of freedom:
  - Features scale factor (LSB_X)
  - State scale factor (LSB_h)
  - Weights scale factor (LSB_w)
  - Truncated bits after state point-wise multiplier (b_{mul})
- Under the following hypothesis:
  - LSBs powers of 2
  - LSBs < 1
  - The output state can be returned as an input with a simple truncation operation!



Outline

Hardware Architecture for GRU layers

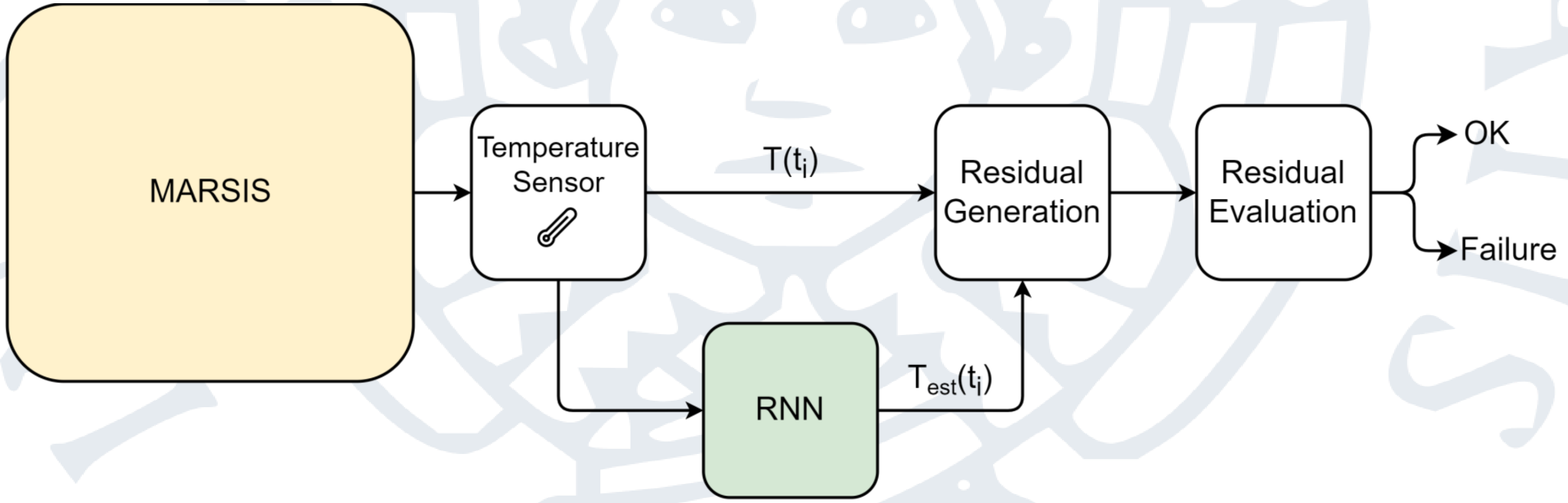
- Acceleration unit for GRU layers:
  - PE blocks for matrix multiplication
  - ACT blocks for activation functions implementation as piece-wise linear approximations
  - ADD, MUL blocks: point-wise adder and multiplier
  - Sat, Truncation blocks for pruning signal bitwidths



Outline

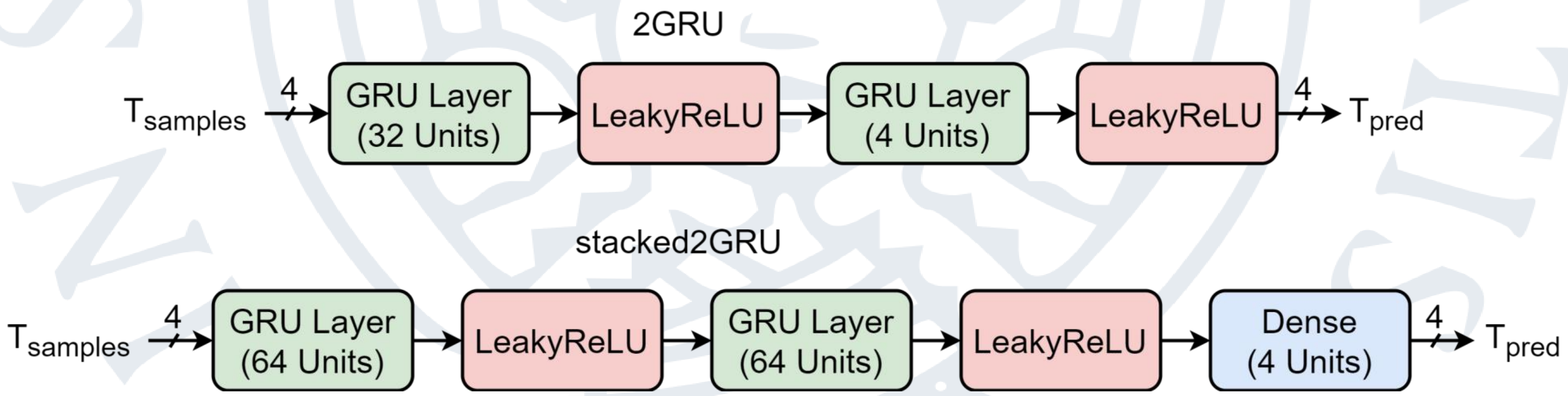
Case Study - Application

- Fault Detection Isolation and Recovery (**FDIR**) system based on **MARSIS** (Mars Advanced Radar for Subsurface and Ionosphere Sounding) dataset [3]
- MARSIS radar is one of the instruments installed inside the payload of the ESA's mission Mars Express, launched on 2nd June 2003
- **RNN-based models** used to forecast temperature values collected by the sensors



Case Study - Models

- Two GRU-based models analyzed:
  - 2GRU: few GRU units w/o Dense layer
  - stacked2GRU: more GRU units w/ Dense layer
- Input: temperature measurements
- Output: predictions on future temperature values
- Maximum Absolute Error (MAE) used as quality metric



Evaluation and Characterization

- Model Compression Results:

Model Architecture	2GRU	stacked2GRU
Floating-point MAE	0.064793	0.046151
Fixed-point MAE	0.076331	0.054003
Quantization Error	0.011538	0.007852
FP Memory Footprint [Mb]	0.117	1.16
FXP Memory Footprint [Mb]	0.018	0.145
Compression Rate	84.5%	87.5%

Evaluation and Characterization

- Hardware implementation on a Xilinx Kintex Ultrascale KU060:

Model Architecture	LUT	FF	LUTRAM	BRAM	DSP	Frequency [MHz]
2GRU	28390 (8.56%)	10088 (1.52%)	-	-	354 (12.83%)	86.90
stacked2GRU	92482 (27.88%)	42567 (6.42%)	-	-	1170 (42.39%)	72.46

Benchmarking with Nvidia Jetson Nano

		Kintex US XQRKU060	Jetson Nano (5W)	Jetson Nano (MAXN)
Mean Absolute Error	2GRU	0.076331	0.064793	0.064793
	stacked2GRU	0.054003	0.046151	0.046151
Power [W]	2GRU	0.728	2.522	2.918
	stacked2GRU	0.874	2.549	3.121
Inference Time [µs]	2GRU	4.56	833.72	510.74
	stacked2GRU	15.87	1409.25	863.26
Energy per Inference [µJ]	2GRU	3.32	2102.6	1490.3
	stacked2GRU	13.87	3592.2	2694.2
TID [Krad]		100	20	20
SEL [MeV·cm²/mg]		80	N/A	N/A

On-going Development

- Complete the design of an end-to-end flow for RNNs
- Enable the support for NanoXplore technology
  - Currently implementing a small-size CNN model (LeNet-5) on NanoXplore NG-Ultra FPGA
  - Presentation of preliminary results at «NanoXplore's 5th BRAVE days», 28-29 November, European Space Research and Technology Centre (ESTEC)

NanoXplore

Thank you for your attention!

Tommaso Pacini

Tommaso.pacini@phd.unipi.it

References

[1] T. Pacini, E. Rapuano, L. Fanucci: “FPG-AI: A Technology-Independent Framework for the Automation of CNN Deployment on FPGAs”, IEEE ACCESS Journal, March2023

[2] T. Pacini, E. Rapuano, P. Nannipieri, L. Fanucci: “A Technology-Independent Toolflow for Automating AI Deployment on FPGAs for On-board Satellite Applications”, 5th SpacE FPGA Users Workshop, March 2023

[3] N. Ferrante, G. Giuffrida, P. Nannipieri, A. Bechini, L. Fanucci: “Fault detection exploiting artificial intelligence in satellite systems”, 2nd International Conference on Applied Intelligence and Informatics, Sept 2022

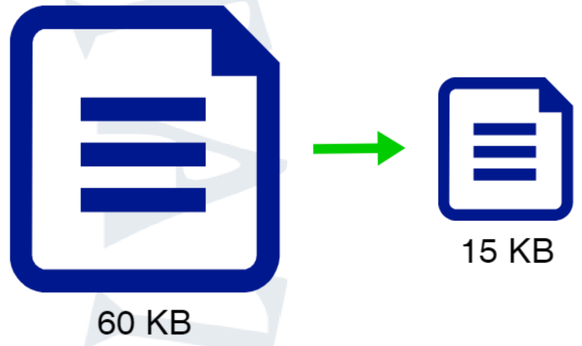
[4] E. Rapuano, T. Pacini, L. Fanucci: “A Post-training Quantization Method for the Design of Fixed-Point-Based FPGA/ASIC Hardware Accelerators for LSTM/GRU Algorithms”, Hindawi Computational Intelligence and Neuroscience Journal, May 2022

Towards the Extension of FPG-AI Toolflow to RNN Deployment

Backup Slides

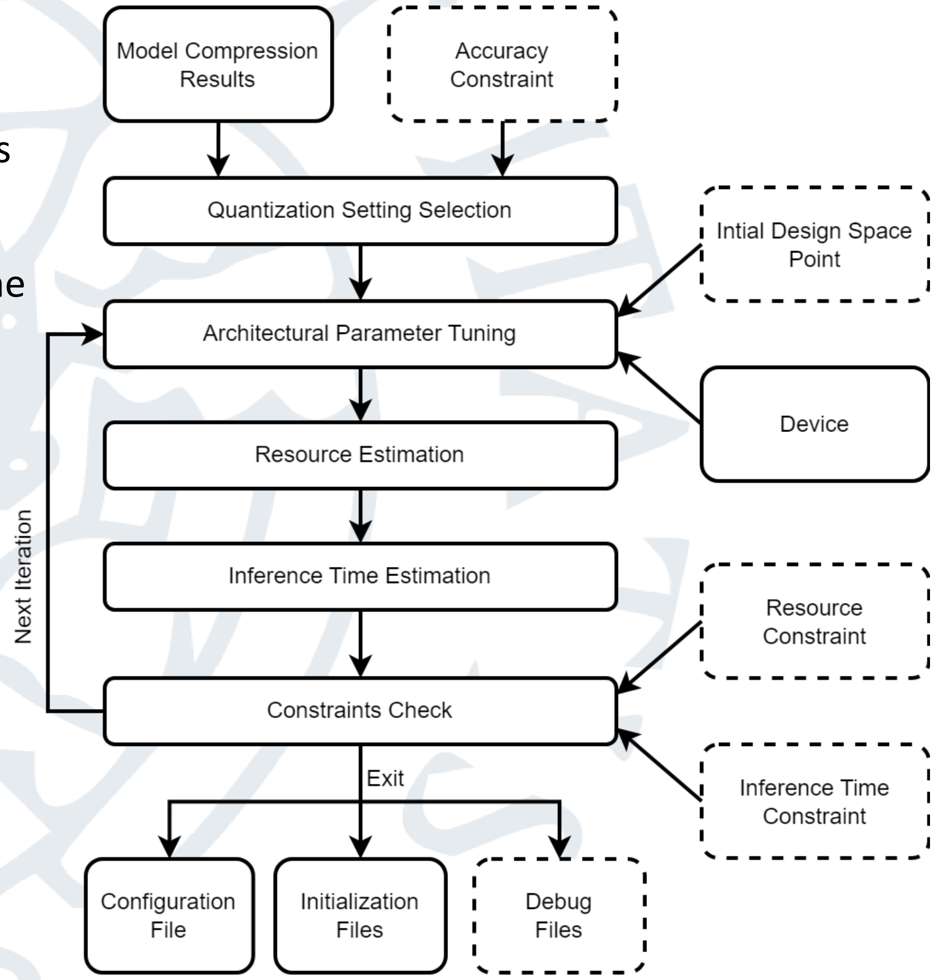
Model Compression

- **Post-training** quantization:
  - **Dynamic** quantization applied on layers’ inputs and weights
  - **Truncation** at layers’ output for further memory footrpint reduction
  - From **floating-point** to **fixed-point** arithmetic for boosting hardware efficiency:
    - Timing performance
    - Area consumption
    - Power consumption
- **Layer folding**: Batch Normalization, Average Pooling



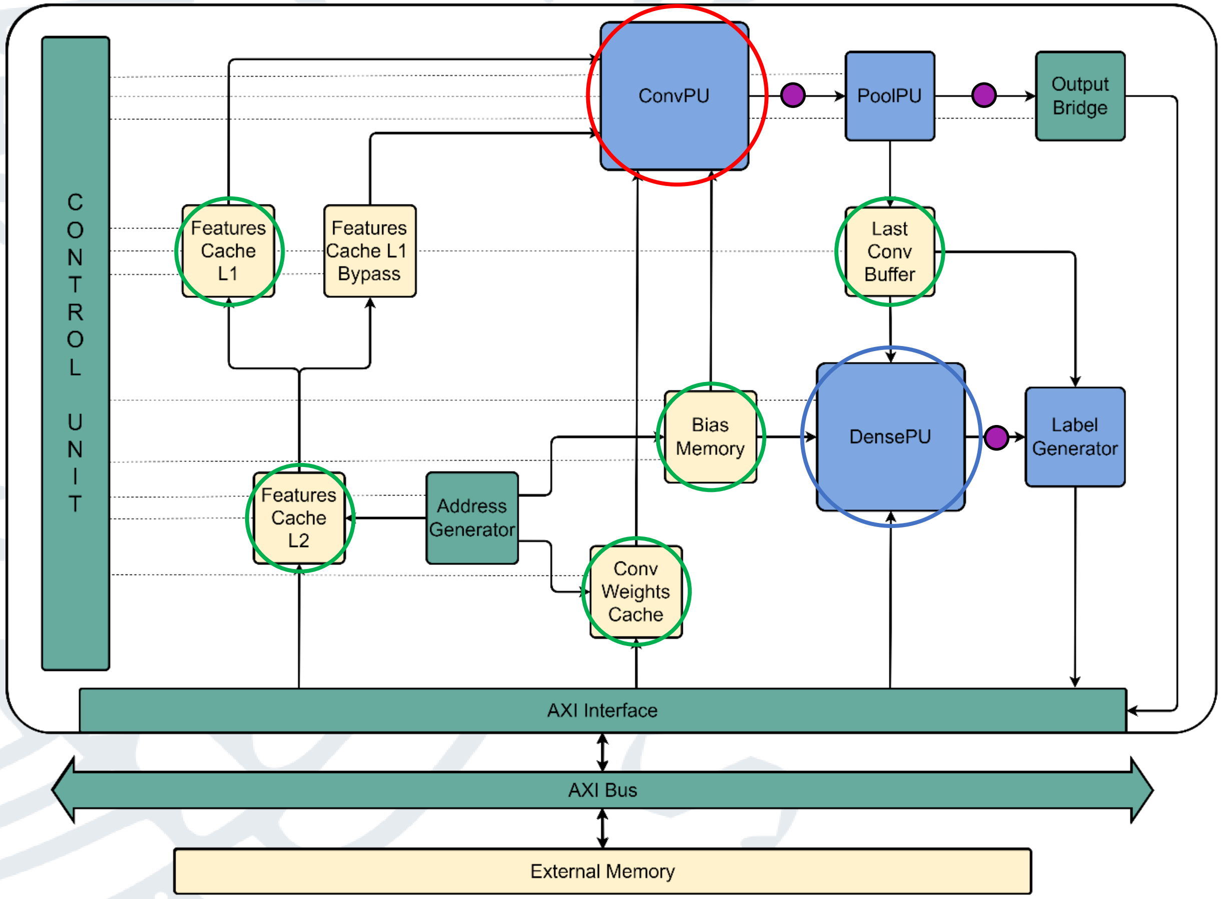
Design Space Exploration (DSE)

- DSE inputs:
  - CNN model
  - Target FPGA
  - User’s constraints (optional)
    - Parameters initial values
    - Minimum accuracy
    - Maximum inference time
    - Resources limitation
- Exploration of the architectural parameters space through an iterative algorithm
  - Detailed analytical MDE model for performance and resource estimation
- DSE outputs:
  - MDE configuration file
  - Initialization files for memories
  - Textual debug files (optional)


Model Compression Results
Accuracy Constraint
Quantization Setting Selection
Intial Design Space Point
Architectural Parameter Tuning
Device
Resource Estimation
Inference Time Estimation
Resource Constraint
Next Iteration
Constraints Check
Inference Time Constraint
Exit
Configuration File
Initialization Files
Debug Files

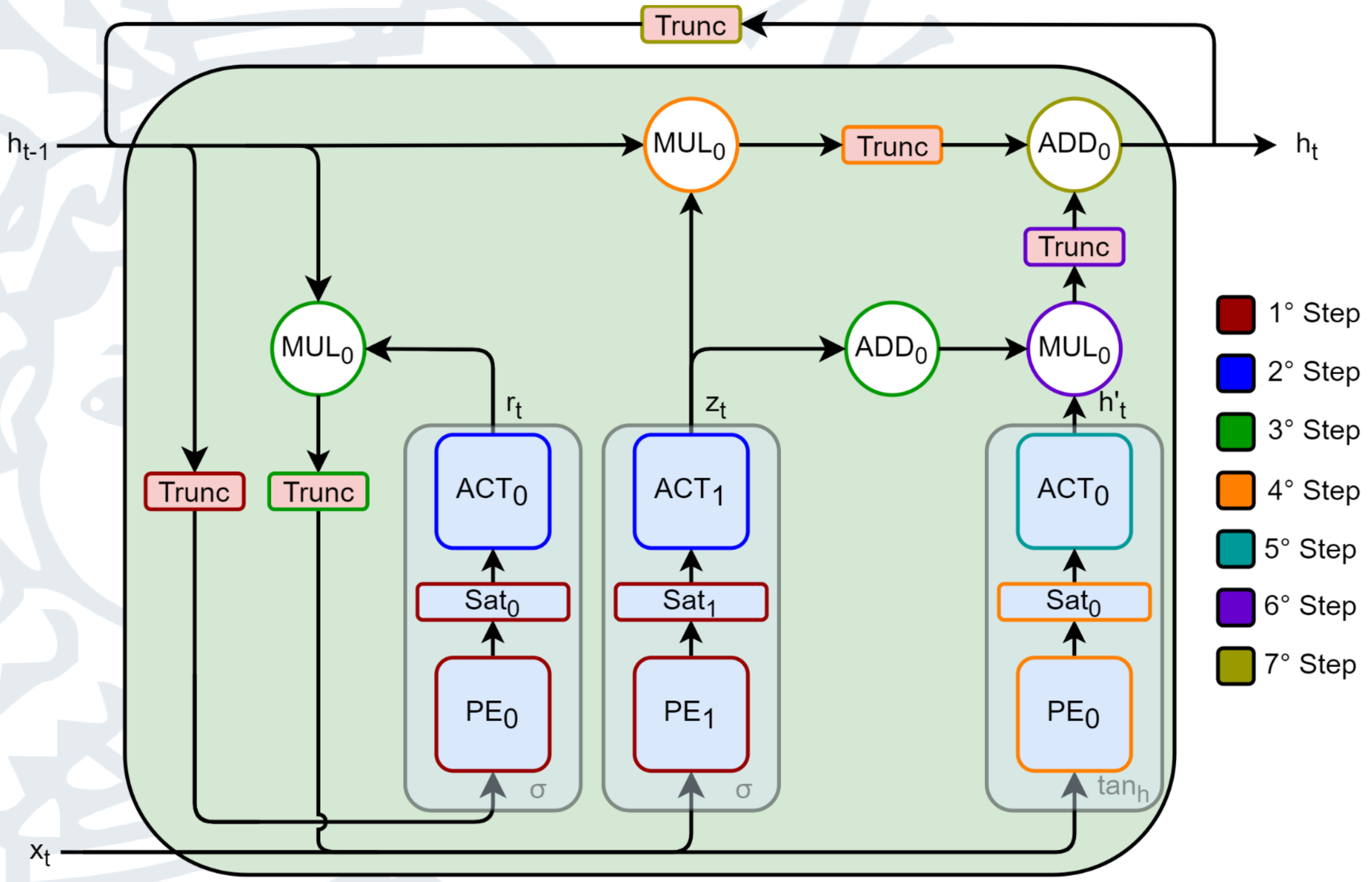
Modular Deep Learning Engine (MDE)

- **High portability**: no third-party IPs used
- **High scalability** in terms of DSP/On-chip Memory utilization
- **Easily configurable** through a file (.vhd)
  - **Model parameters:**
    - Input shape, # Layers, Layers type, # Classes, etc.
  - **Quantization parameters:**
    - Inputs/weights bitwidths, Truncation and Saturation bits
  - **Architectural parameters:**
    - # MAC units (N_{PE}),# Neurons, Memory primitives for each IP, etc.



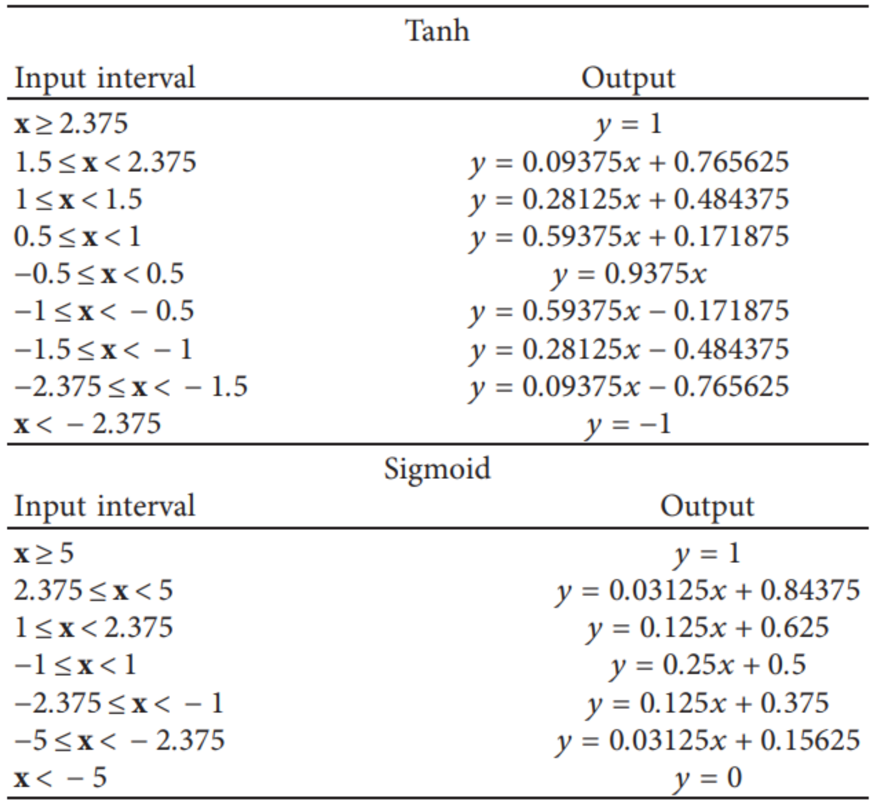
Enhanced Scheduling for Resource Sharing

- Implemented resource-sharing strategy within each GRU layer
- Exploiting layer specific dataflow to optimize resource consumption
- Same sequence of operations (e.g. gate units, point-wise operators) repeated in separate time slots
- Design of a control logic that enables the reuse of resources over time
- Achieving resource savings w/o affecting the throughput of the layer

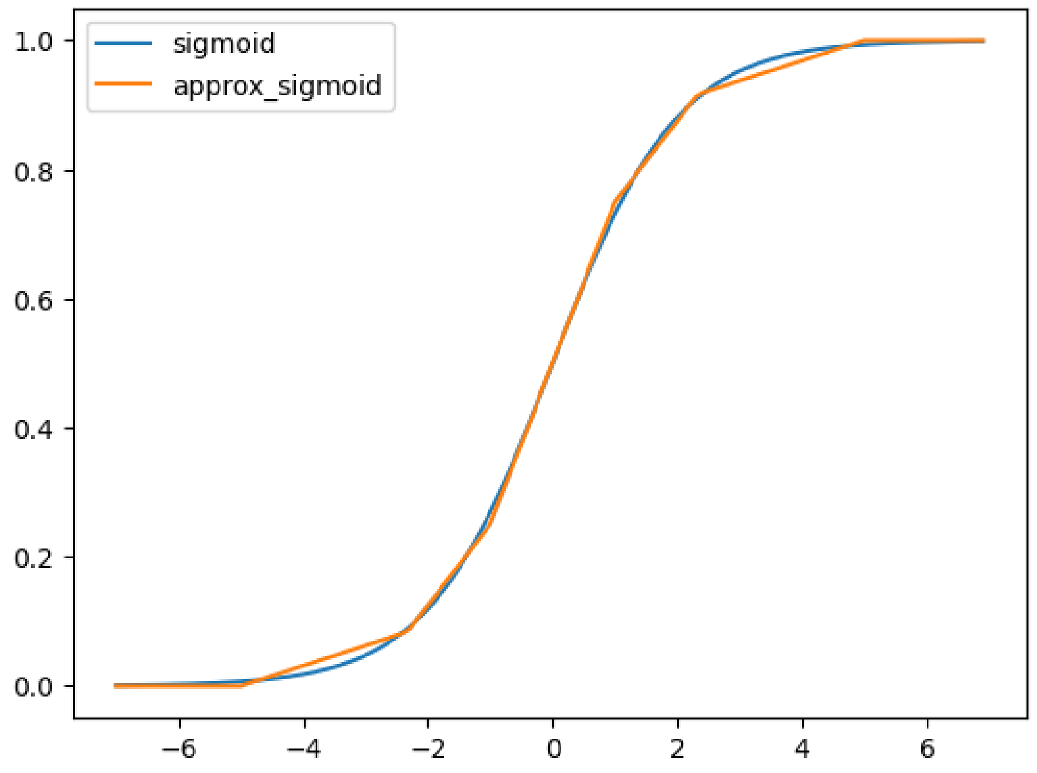


Quantization of Activation Functions

- Non-linear activation functions (sigmoid, tanh) computed exploiting a piece-wise linear approximation:



Tanh	
Input interval	Output
$\mathbf{x} \geq 2.375$	$y = 1$
$1.5 \leq \mathbf{x} < 2.375$	$y = 0.09375x + 0.765625$
$1 \leq \mathbf{x} < 1.5$	$y = 0.28125x + 0.484375$
$0.5 \leq \mathbf{x} < 1$	$y = 0.59375x + 0.171875$
$-0.5 \leq \mathbf{x} < 0.5$	$y = 0.9375x$
$-1 \leq \mathbf{x} < -0.5$	$y = 0.59375x - 0.171875$
$-1.5 \leq \mathbf{x} < -1$	$y = 0.28125x - 0.484375$
$-2.375 \leq \mathbf{x} < -1.5$	$y = 0.09375x - 0.765625$
$\mathbf{x} < -2.375$	$y = -1$
Sigmoid	
Input interval	Output
$\mathbf{x} \geq 5$	$y = 1$
$2.375 \leq \mathbf{x} < 5$	$y = 0.03125x + 0.84375$
$1 \leq \mathbf{x} < 2.375$	$y = 0.125x + 0.625$
$-1 \leq \mathbf{x} < 1$	$y = 0.25x + 0.5$
$-2.375 \leq \mathbf{x} < -1$	$y = 0.125x + 0.375$
$-5 \leq \mathbf{x} < -2.375$	$y = 0.03125x + 0.15625$
$\mathbf{x} < -5$	$y = 0$


sigmoid
approx_sigmoid