Edge AI and Machine Learning

Romina Soledad Molina, Ph.D.

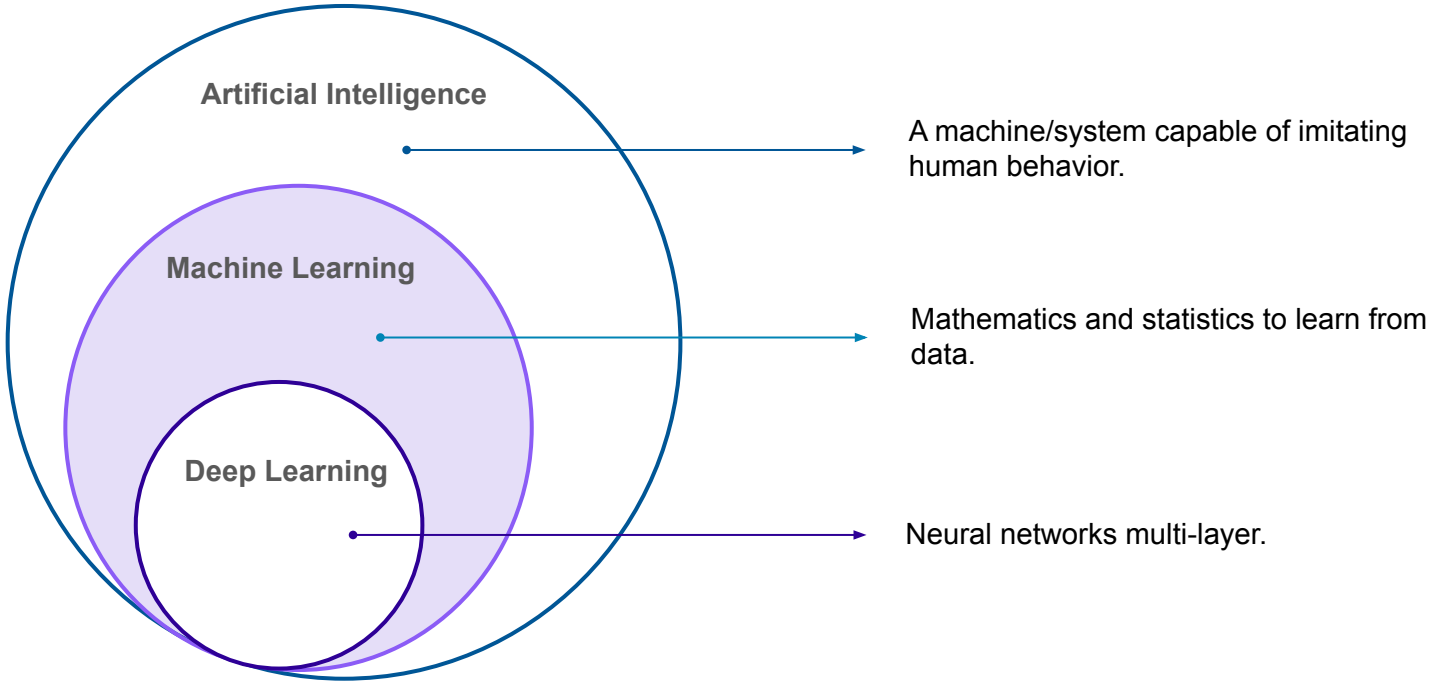
School on Applied AI for Sustainable Development | smr 4210 | Trieste, Italy
March 2026

Outline

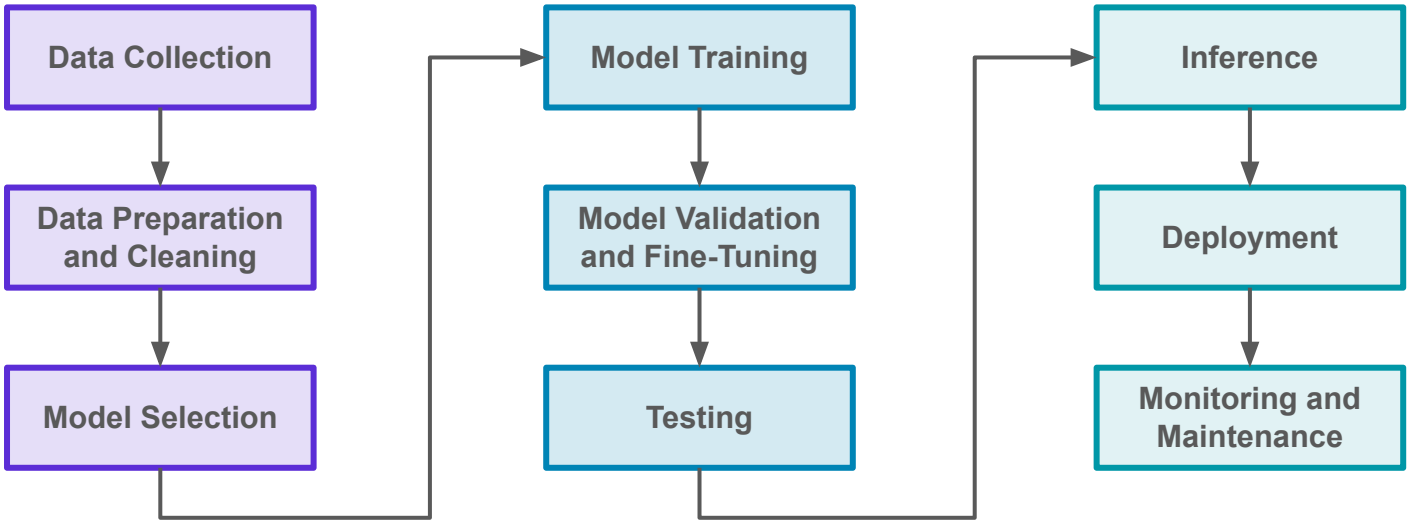
- Introduction to Machine Learning and Edge AI
- Edge Artificial Intelligence
- ML for Real-World Deployment: What truly matters once a model leaves the notebook
- Edge AI based on FPGA: From trained model to real-time hardware inference

Introduction to Machine Learning and Edge AI

Introduction to Machine Learning and Edge AI

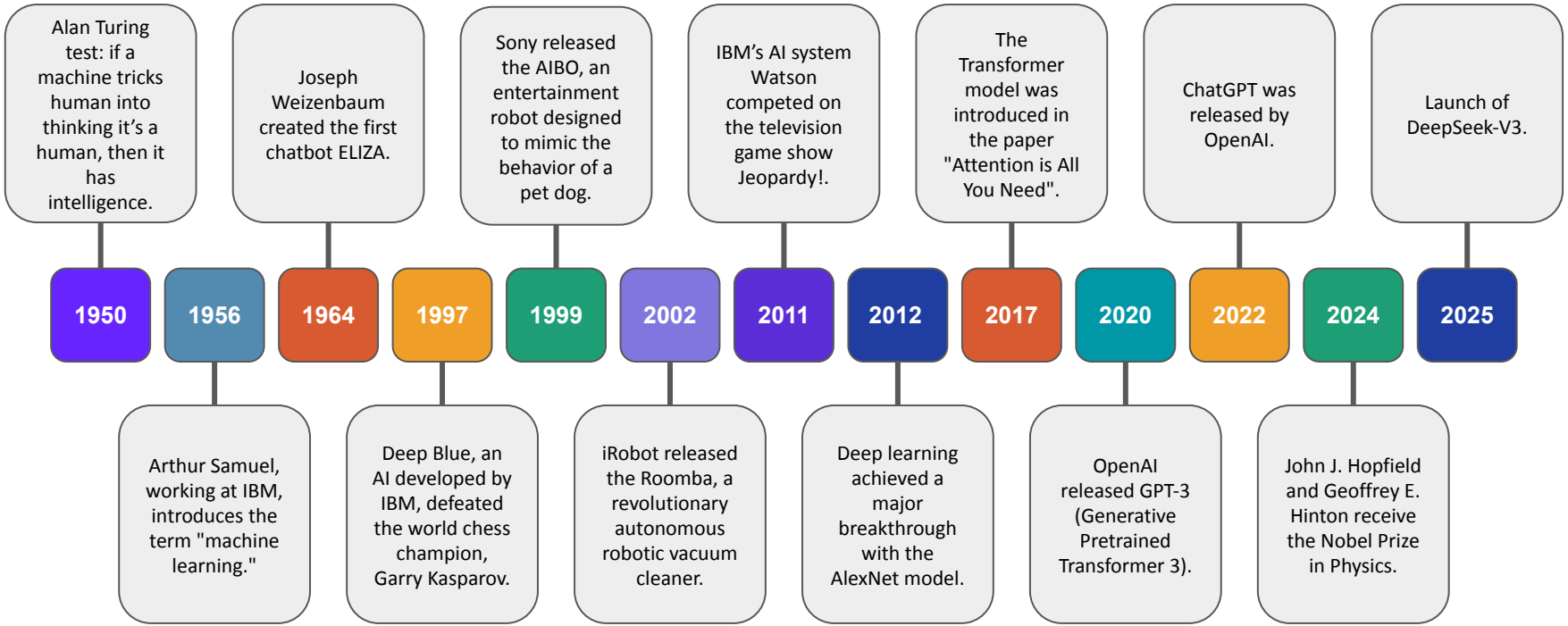


Introduction to Machine Learning and Edge AI



[Based on the original from Qualcomm Academy]

Introduction to Machine Learning and Edge AI



Introduction to Machine Learning and Edge AI

Why has the explosion of AI happened in recent years? Three main components

Introduction to Machine Learning and Edge AI

Why has the explosion of AI happened in recent years? Three main components

Big data

The massive amount of data generated every day (from social media, sensors, and online activity) provides the fuel that AI systems need to learn and improve

Introduction to Machine Learning and Edge AI

Why has the explosion of AI happened in recent years? Three main components

Big data

The massive amount of data generated every day (from social media, sensors, and online activity) provides the fuel that AI systems need to learn and improve

Software

New algorithms, neural network architectures, and open-source frameworks have dramatically improved AI's capabilities and made development more accessible.

Introduction to Machine Learning and Edge AI

Why has the explosion of AI happened in recent years? Three main components

Big data

The massive amount of data generated every day (from social media, sensors, and online activity) provides the fuel that AI systems need to learn and improve

Software

New algorithms, neural network architectures, and open-source frameworks have dramatically improved AI's capabilities and made development more accessible.

Hardware

Advances in GPUs, TPUs, and cloud computing have made it possible to train large AI models faster and more efficiently than ever before.

Edge
Artificial Intelligence

Edge AI

“Edge artificial intelligence (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

“Edge artificial intelligence (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]

Low Latency	Privacy & security	Scalability & reliability	Lower Bandwidth Use	Energy efficiency
Real-time decisions, reducing cloud dependence	Data stays on the device.	Devices run autonomously, even offline.	Less data sent to the cloud.	Optimized for local processing.

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

“Edge artificial intelligence (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]

Low Latency	Privacy & security	Scalability & reliability	Lower Bandwidth Use	Energy efficiency
Real-time decisions, reducing cloud dependence	Data stays on the device.	Devices run autonomously, even offline.	Less data sent to the cloud.	Optimized for local processing.

Bringing intelligence closer to the data, enabling low-latency, privacy-preserving, and energy-aware AI.

[IBM] https://www.ibm.com/think/topics/edge-ai

Edge AI

Challenges

Edge AI
Challenges

Hardware & Energy

Limited power

Memory / storage

limits

Thermal constraints

Edge AI
Challenges

Hardware & Energy

Limited power

Memory / storage limits

Thermal constraints

Computation & Performance

Latency & real-time

Limited compute capacity

Hardware heterogeneity

Model size vs accuracy tradeoff

Edge AI
Challenges

Hardware & Energy

Limited power

Memory / storage limits

Thermal constraints

Computation & Performance

Latency & real-time

Limited compute capacity

Hardware heterogeneity

Model size vs accuracy tradeoff

Data & Deployment

Privacy & security

Network issues

Data distribution shift

Edge AI

Heterogeneous Computing: A Key Enabler for Edge AI

CPU

Few cores
Poor parallelism for ML inference

High latency
Slow for compute-intensive tasks

Not optimized for inference
General purpose, not ML-specific

GPU

High power consumption
Not suitable for battery-powered edge

Size and cost
Too large for small embedded systems

Memory bandwidth bottleneck
Data movement limits throughput

FPGA

Hard to program
Requires HDL expertise

Limited flexibility
Not general purpose

Longer development cycle
Synthesis and timing closure

What if we could combine the strengths of different processors CPUs, GPUs, and FPGAs to overcome these limits?

Edge AI

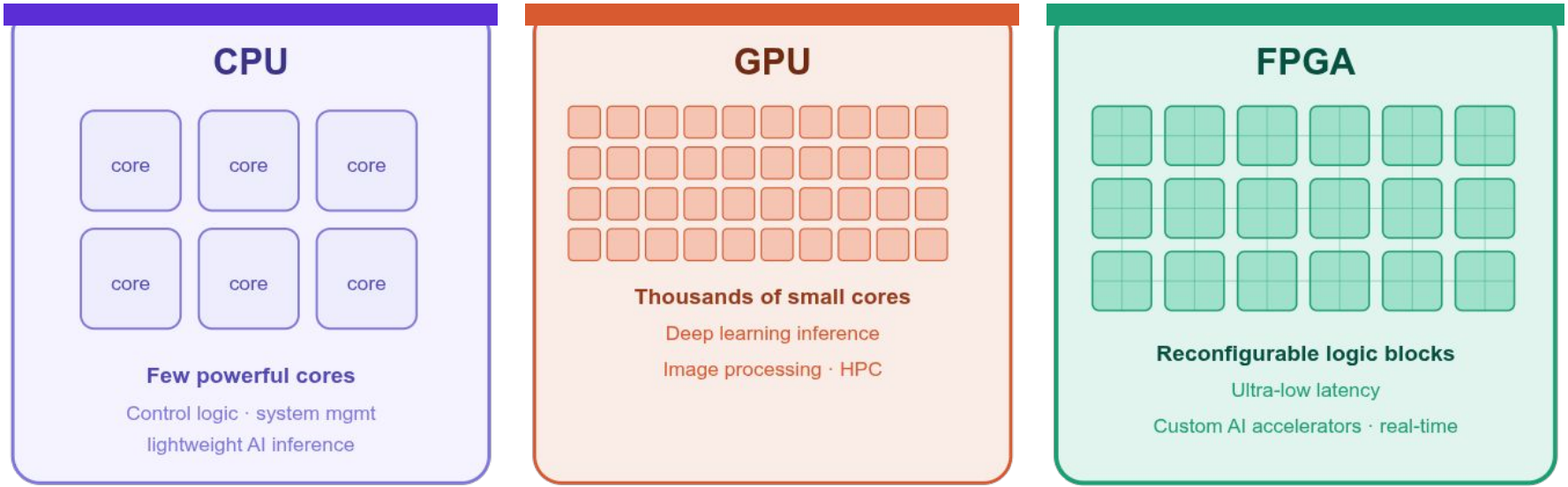
Heterogeneous Computing: A Key Enabler for Edge AI

By incorporating CPUs, GPUs, and FPGAs into a single system, **heterogeneous computing** becomes possible.

This approach maximizes the strengths of each component, efficiently distributing edge workloads to improve both performance and energy consumption.

Edge AI

Heterogeneous Computing: A Key Enabler for Edge AI



Edge AI

The Deployment Gap

Problem:

Modern ML models are often large, resulting in high memory consumption and slow inference.

The most accurate models (such as deep neural networks) are large and expensive in terms of memory and processing time.

Edge AI

The Deployment Gap

Model	Parameters	Disk size
BERT-large	~340M	~1.3 GB
VGG-19	~144M	~548 MB
VGG-16	~138M	~528 MB
BERT-base	~110M	~440 MB
YOLOv3	~62M	~236 MB
ResNet-50	~25.6M	~98 MB
SpineNet-49S	~11M	~45 MB
MobileNetV3-Large	**~5.4M**	**~20 MB**

Green = compressed / efficient model target · All others are candidates for compression

Edge AI

The Deployment Gap

Problem:

Modern ML models are often large, resulting in high memory consumption and slow inference.

Solution:

Apply model compression techniques to reduce size and improve efficiency.

Edge AI

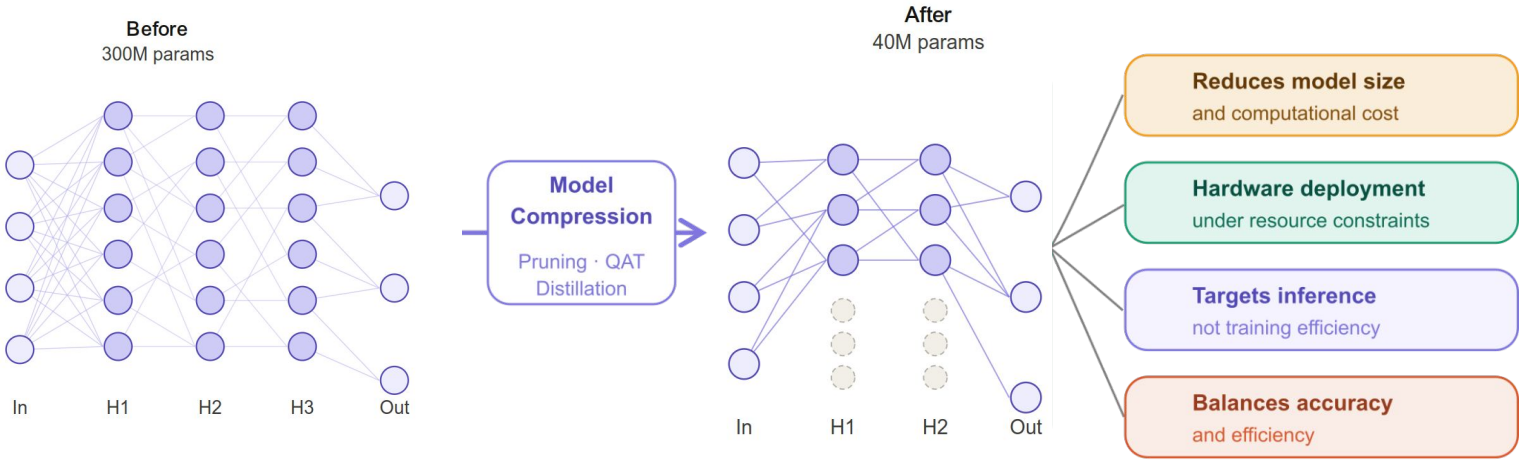
The Deployment Gap

Model	Parameters	Disk size	Accuracy
DistilBERT	66M ↓ 40%	~250 MB	**97% of BERT**
SqueezeNet 1.2M	1.2M ↓ 99%	~4.8 MB ↓ 99%	**Similar to VGG-16**
YOLOv8-Nano	3.2M ↓ 92%	~8 MB ↓ 90%	**Good trade-off with YOLOv8-L**
YOLO-Fastest	0.2M ↓ 99.5%	~1 MB ↓ 99%	**Lower accuracy**

These examples show that significant reductions in model size are possible with limited impact on accuracy

Edge AI

Compression



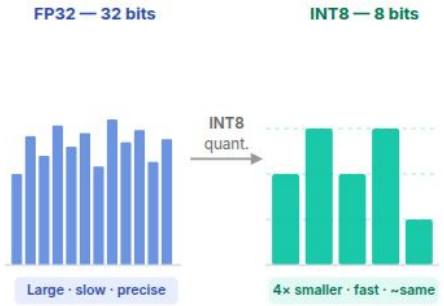
Model compression reduces size and complexity to enable efficient inference under real-world constraints

Edge AI

Compression

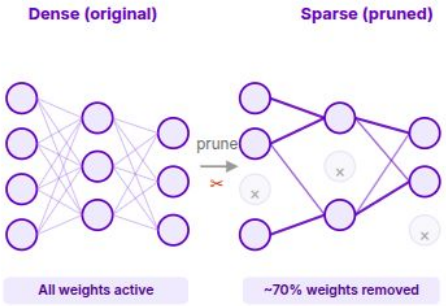
Quantization

Reduce weight precision to shrink model size and speed up inference



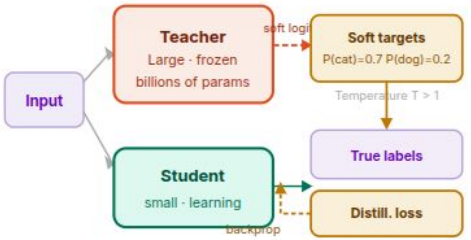
Pruning

Remove low-importance connections to create a sparse, leaner network



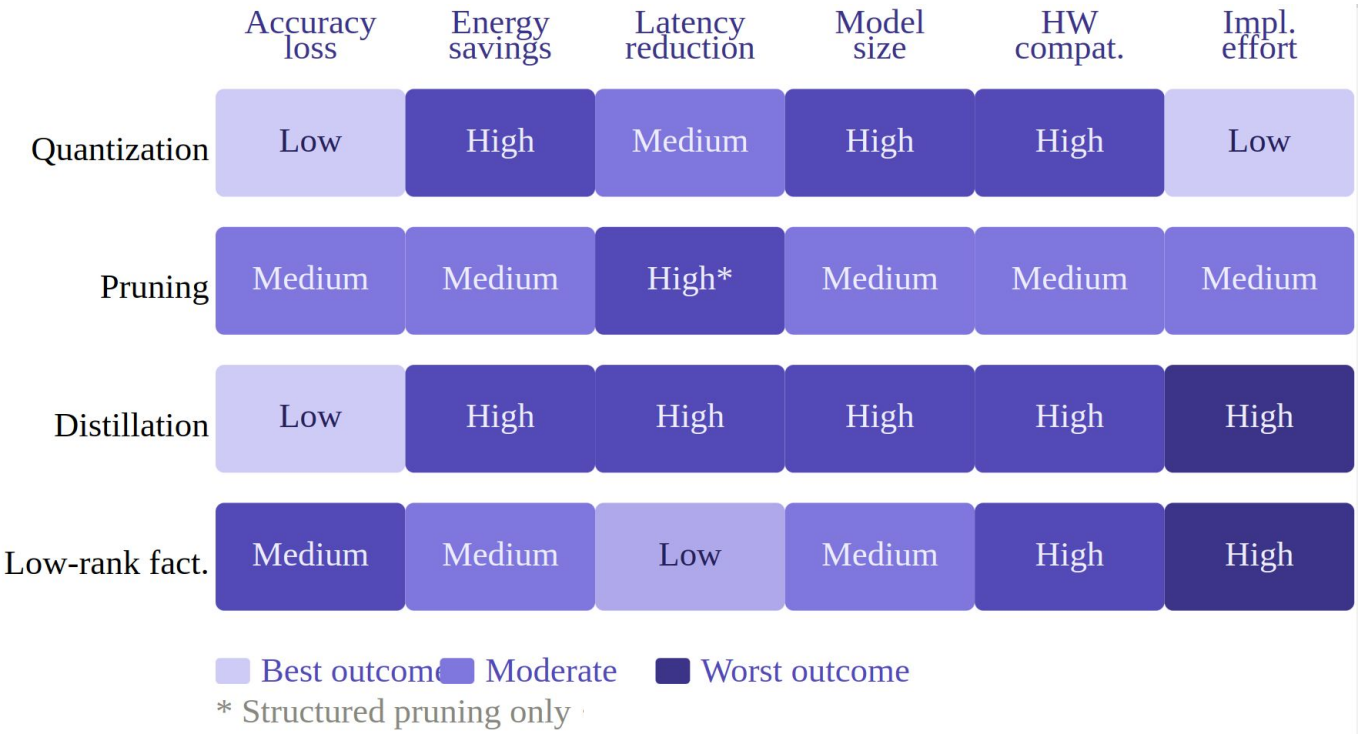
Knowledge Distillation

A large teacher model trains a compact student model via soft targets



Edge AI

Compression



Edge AI

Compression

Model compression is not a single technique,
but a design space where accuracy, efficiency, and hardware constraints must be balanced.

Edge AI

Compression

Model compression is not a single technique,
but a design space where accuracy, efficiency, and hardware constraints must be balanced.

Every optimization (pruning, quantization, or distillation) improves efficiency at the cost of something else. The goal is not to avoid trade-offs, but to choose the right ones.

ML for Real-World Deployment

What truly matters once a model leaves the notebook

Research vs Deployment

Research

The question:

Does it reach 98% accuracy on the test set?

Success metric:
accuracy, F1, AUC

Research vs Deployment

Research

The question:

Does it reach 98% accuracy on the test set?

Success metric:
accuracy, F1, AUC

vs

Deployment

The question:

Does it run in 10ms with 64 KB of RAM?

Success metric:
latency, memory, energy, accuracy within tolerance

The four deployment constraints

Latency

< 10ms real-time
< 100ms near real-time
< 1s batch acceptable

Memory

MCU: 64–256 KB flash
FPGA: 2–8 MB BRAM
Edge GPU: 256 MB+

Energy

Battery: µW–mW budget
Mains: less critical
Harvesting: µW limit

Accuracy tolerance

Safety-critical: > 99%
Industrial: > 95%
Consumer: > 90%

Conventional AI vs Edge AI

Conventional AI
Cloud/server-based inference

High-latency
Network round-trips add delay

High energy demand
TOPS-scale compute, not battery-friendly

Privacy exposure
Raw data sent to external servers

Deep neural networks
Billios of params, dense matrix operations

Ineffective at the extreme edge
Alternative paradigms needed

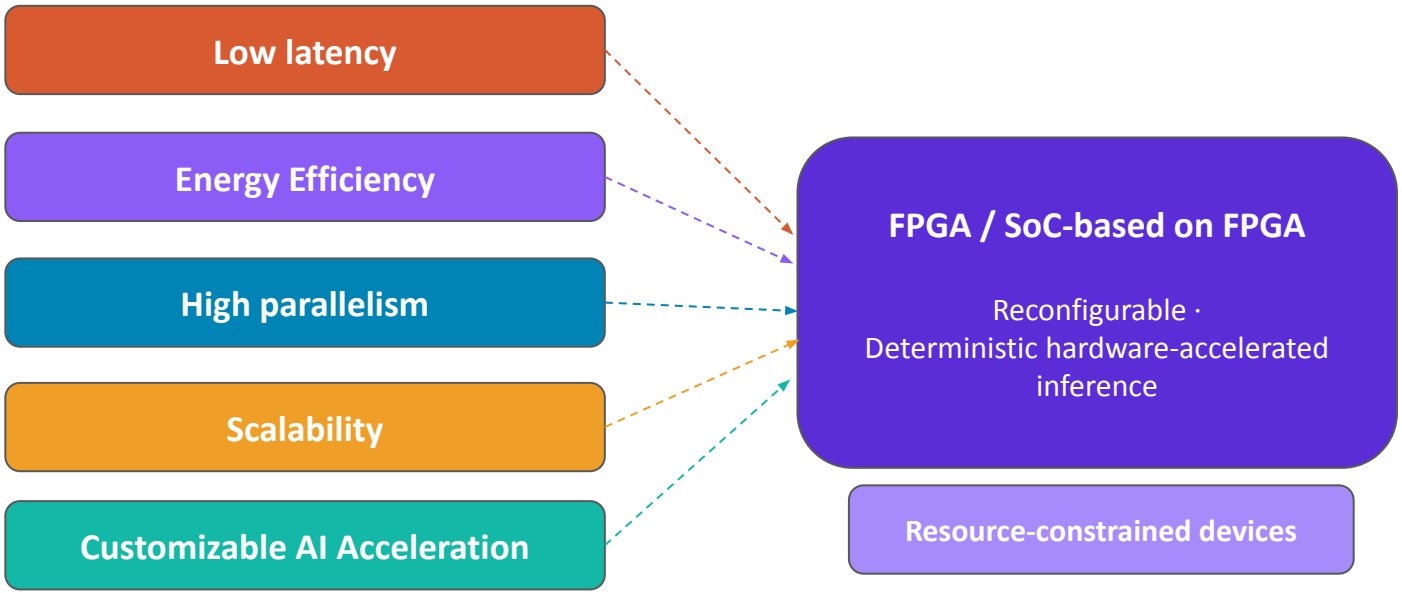
Conventional AI vs Edge AI

Conventional AI	Edge AI
Cloud/server-based inference	On-device, local processing
High-latency Network round-trips add delay	Real-time response Minimal latency, no network round-trip
High energy demand TOPS-scale compute, not battery-friendly	Ultra-low power consumption uW to mW range, long battery life
Privacy exposure Raw data sent to external servers	Privacy-aware computation Data never leaves the device
Deep neural networks Billios of params, dense matrix operations	Dynamic multisensory input Different sensors, processed locally
Ineffective at the extreme edge Alternative paradigms needed	Efficient local intelligence New paradigms required

Edge AI based on FPGA
From trained model to real-time hardware inference

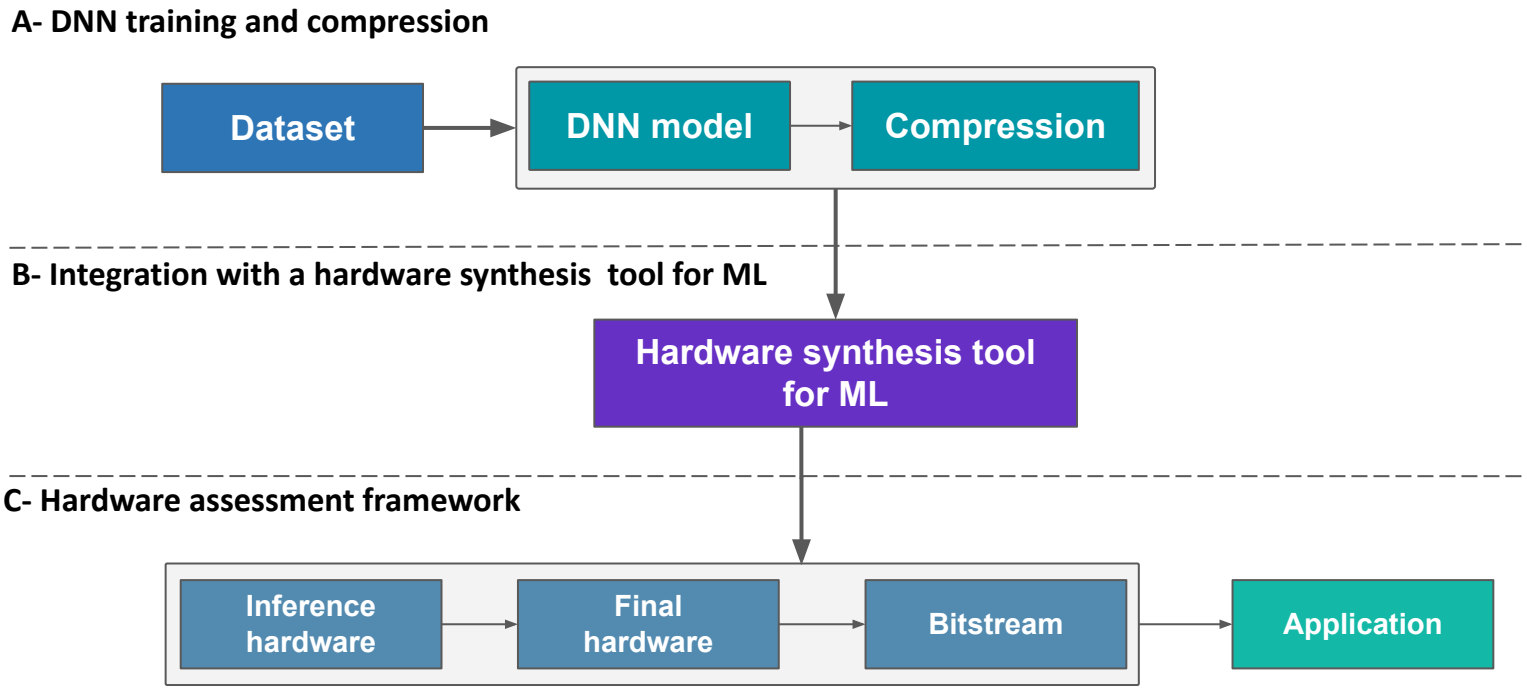
Edge AI based on FPGA

From trained model to real-time hardware inference



Edge AI based on FPGA

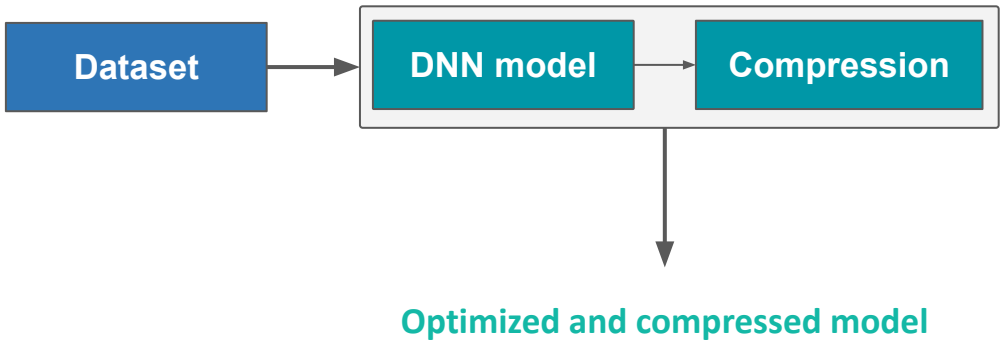
From trained model to real-time hardware inference



Available at https://github.com/RomiSolMolina/workflowCompressionML

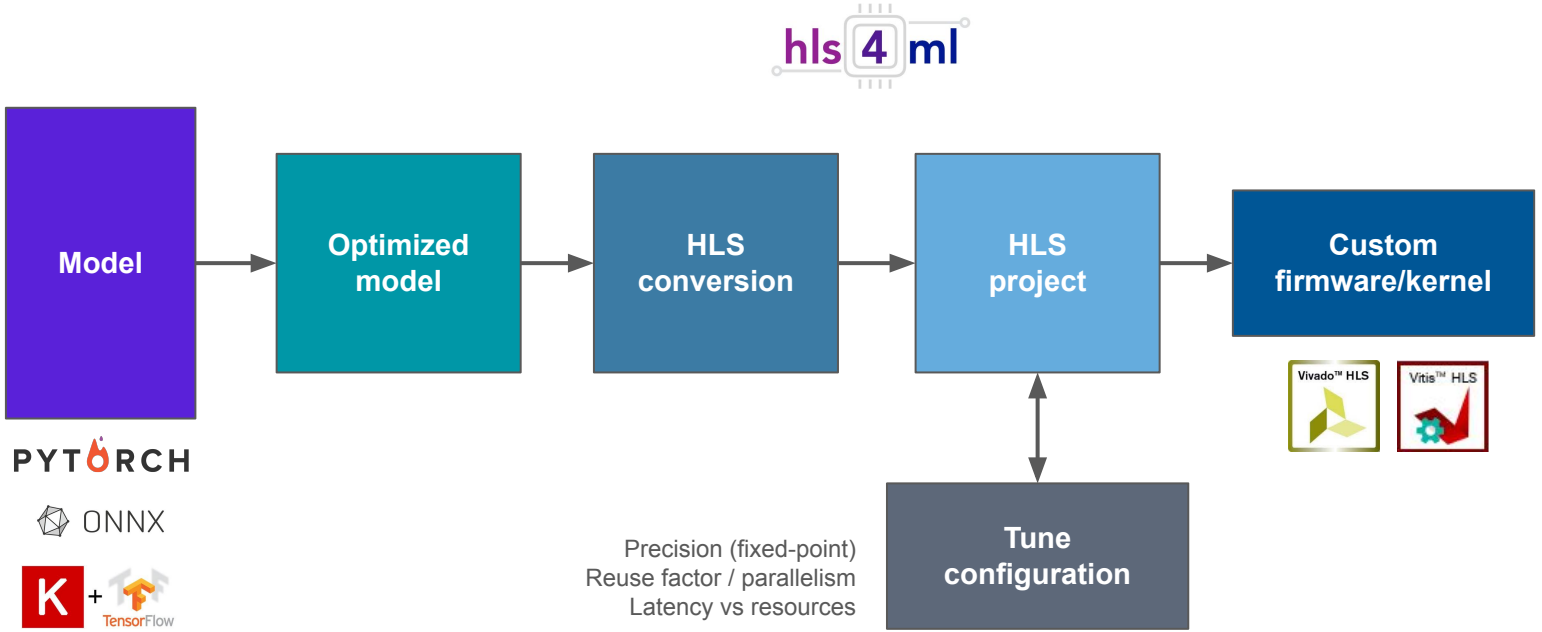
Edge AI based on FPGA

From trained model to real-time hardware inference



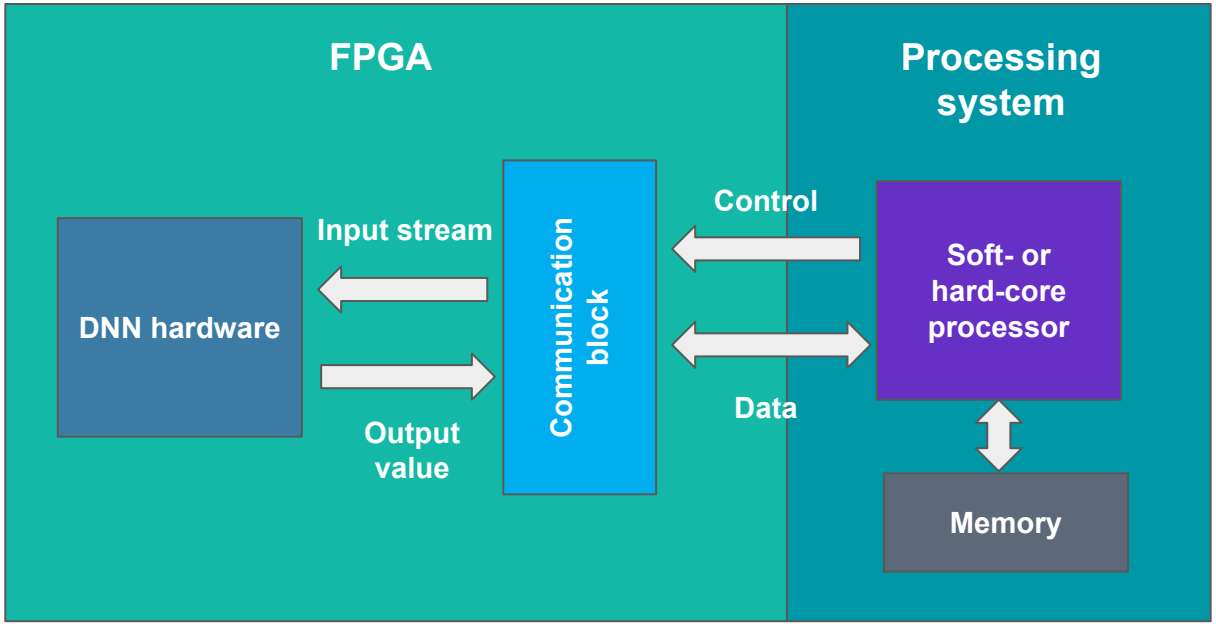
Edge AI based on FPGA

From trained model to real-time hardware inference



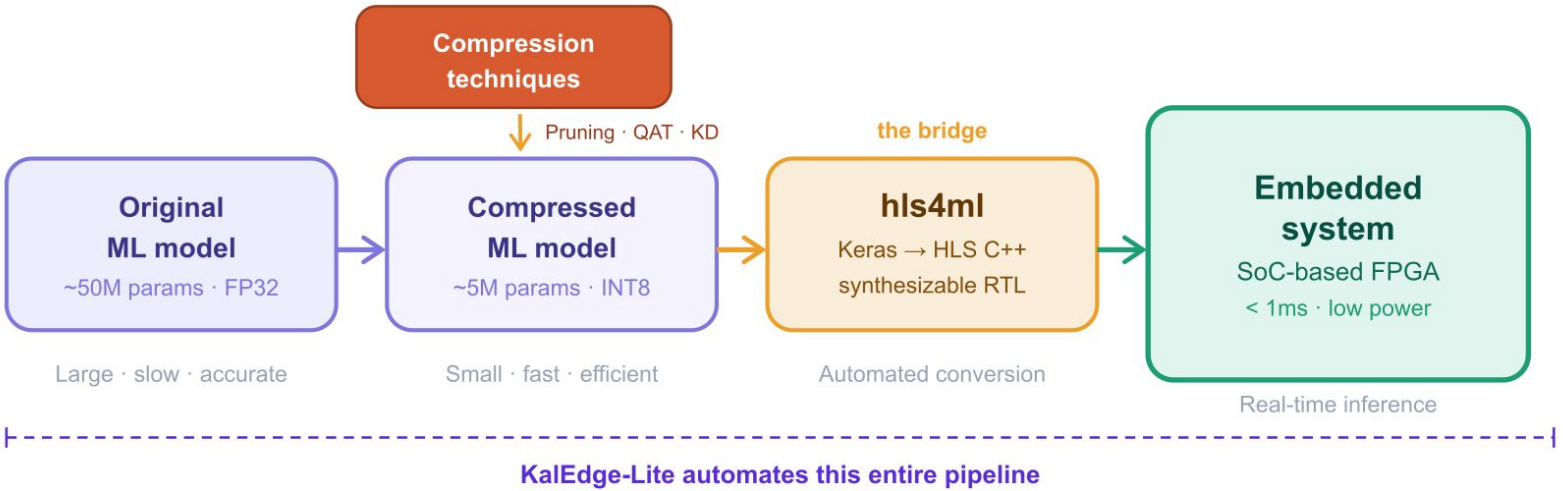
Edge AI based on FPGA

From trained model to real-time hardware inference



Edge AI based on FPGA

From trained model to real-time hardware inference



Edge AI based on FPGA

From trained model to real-time hardware inference

KalEdge Lite

Dataset | Architecture | Training | Pipeline | Reports & Export | hls4ml integration | Documentation

Load Dataset

Dataset: MNIST Input: MLP (Flatten)

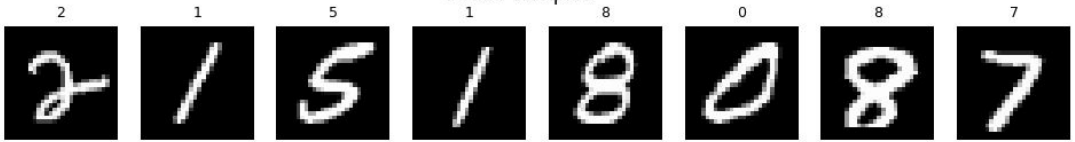
Load Dataset

MNIST loaded | Input shape **(784,)** · 10 classes

Trained models cleared. Retrain after loading a new dataset.

Train: (54000, 784) · Val: (6000, 784) · Test: (10000, 784)

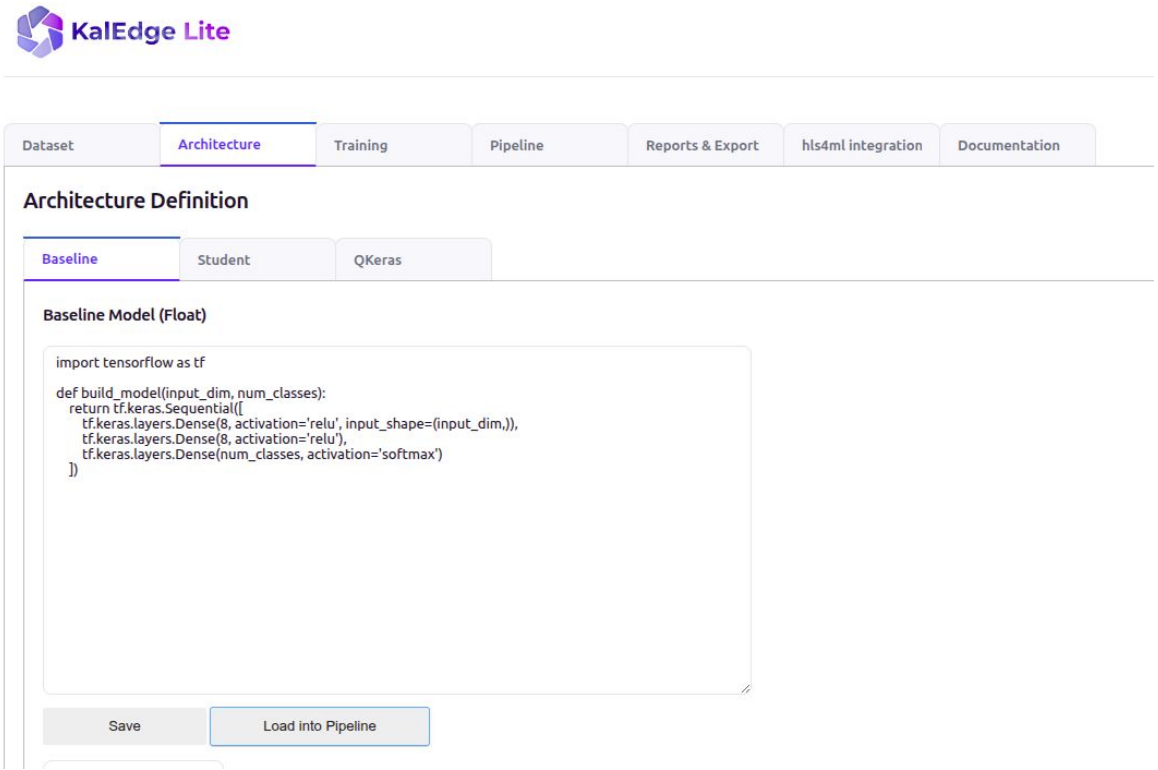
MNIST samples



MLP (Flatten) mode | each 28×28 image is reshaped into a **784-dimensional vector** before being fed to the network. Spatial structure is not preserved.

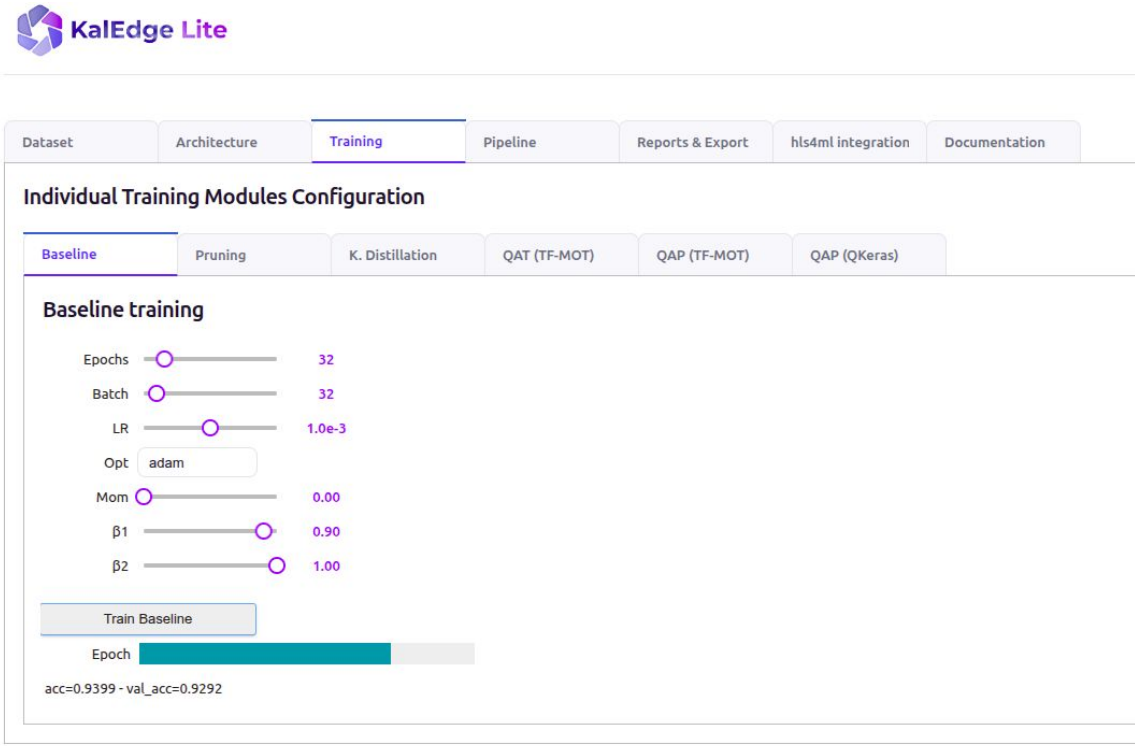
Edge AI based on FPGA

From trained model to real-time hardware inference



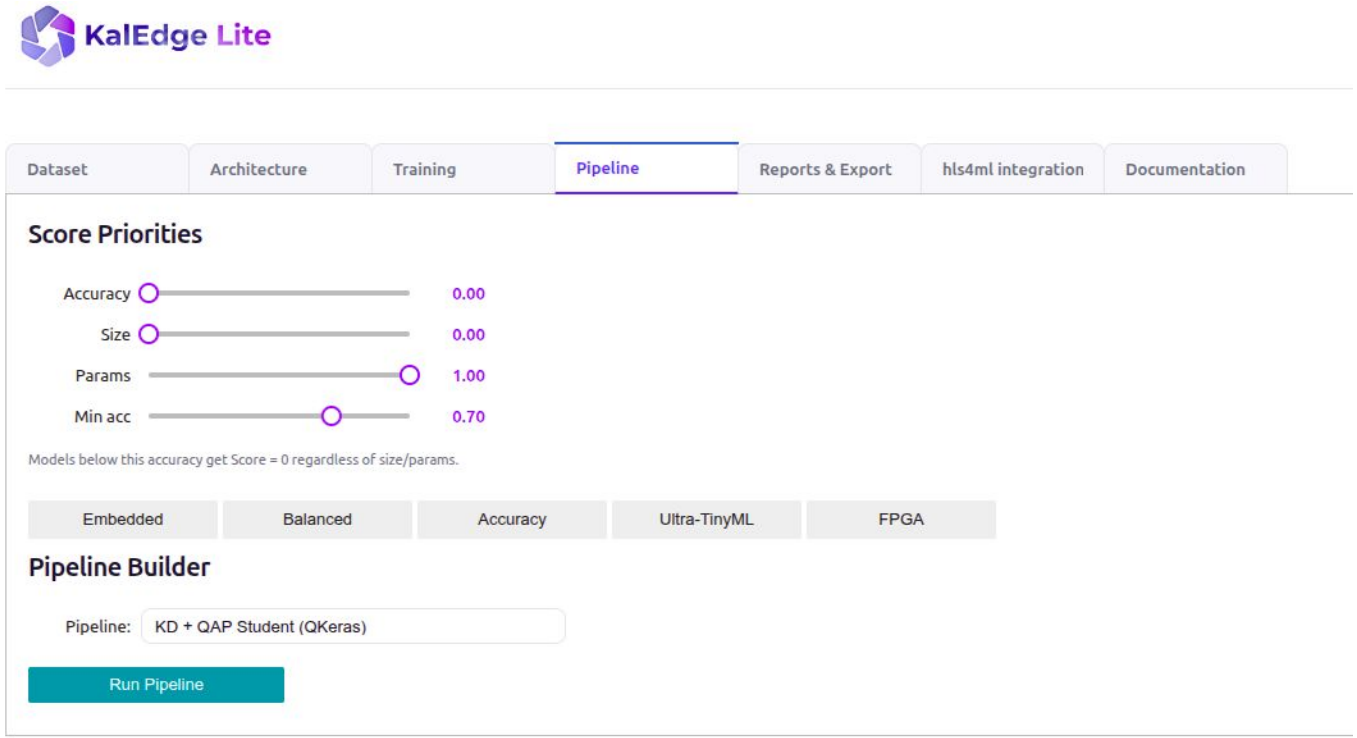
Edge AI based on FPGA

From trained model to real-time hardware inference



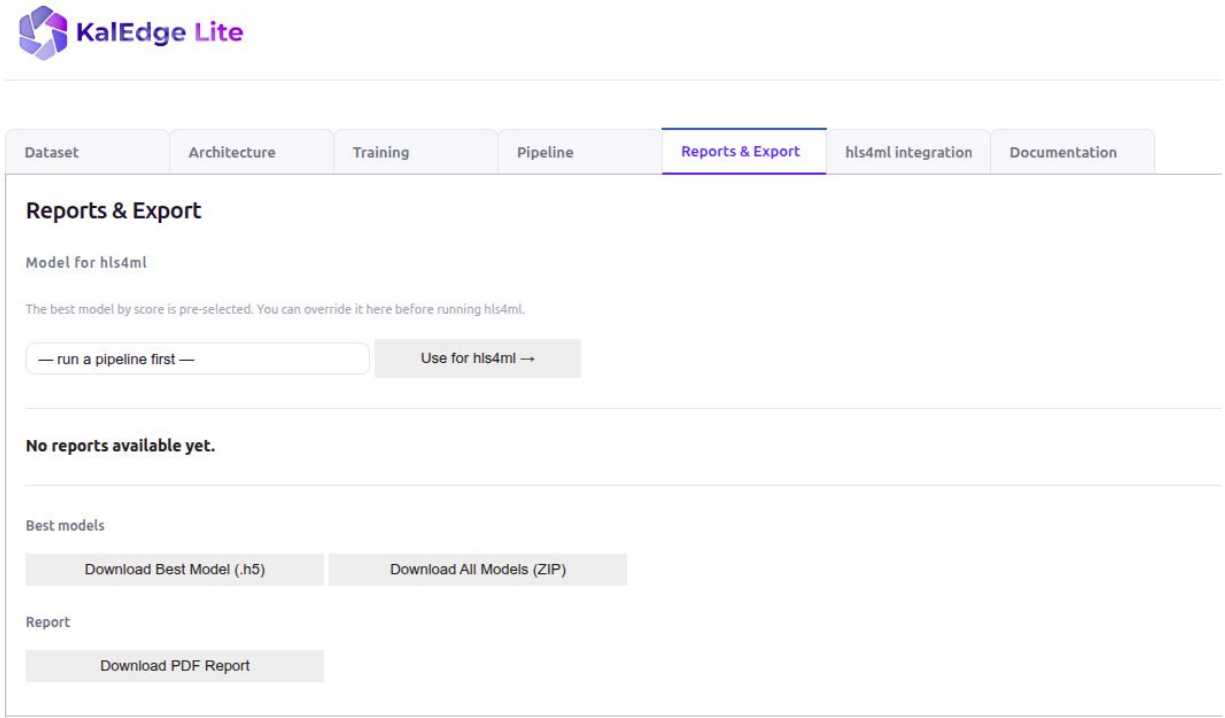
Edge AI based on FPGA

From trained model to real-time hardware inference



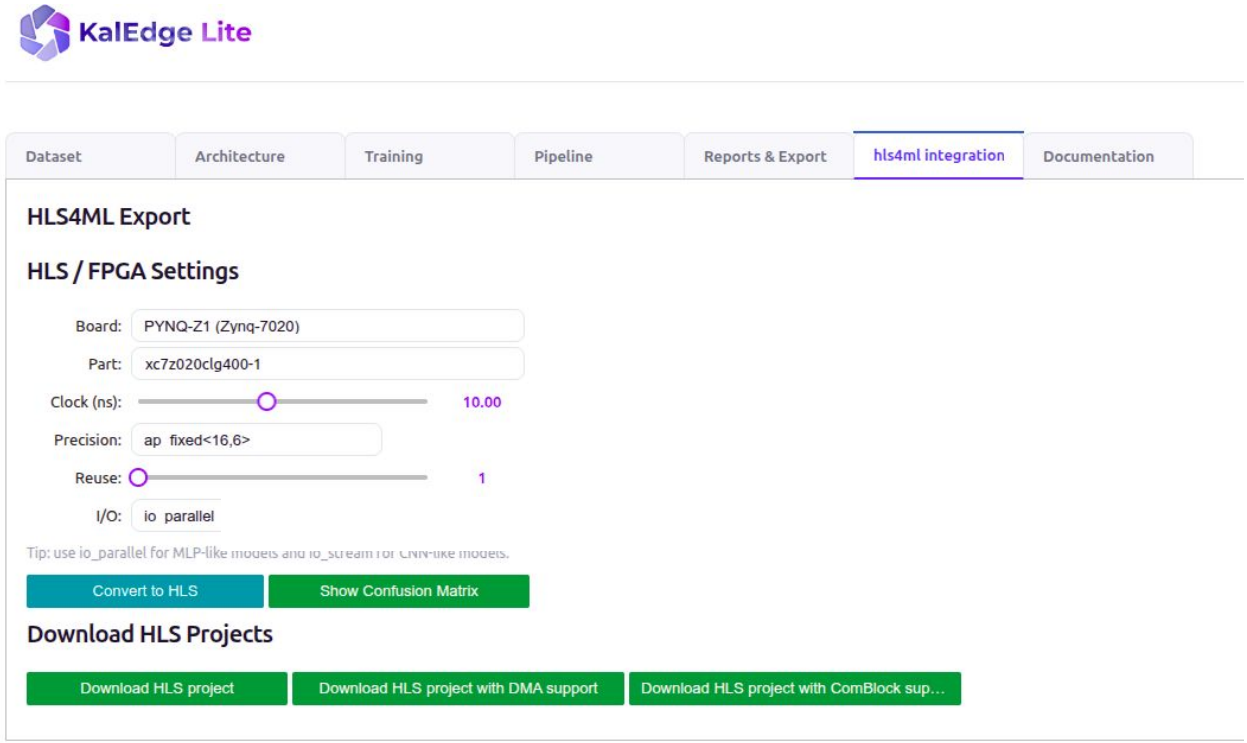
Edge AI based on FPGA

From trained model to real-time hardware inference



Edge AI based on FPGA

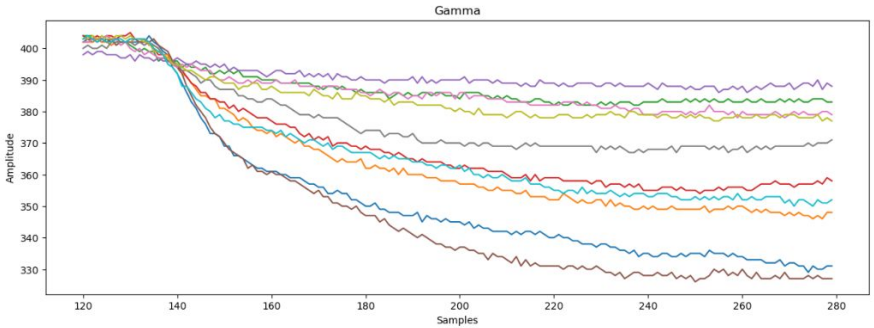
From trained model to real-time hardware inference

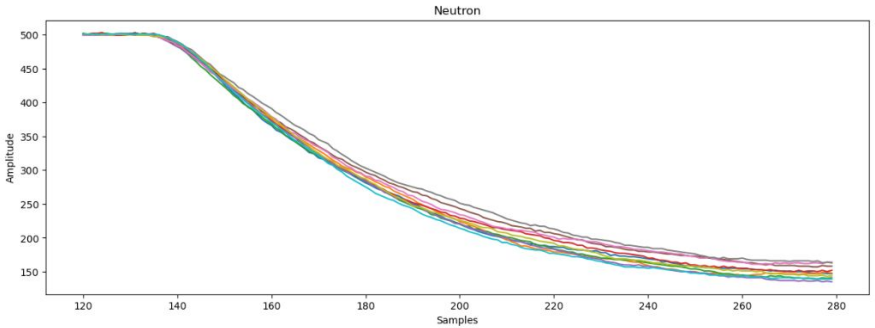


Edge AI based on FPGA

From trained model to real-time hardware inference

Gamma-Neutron Discrimination

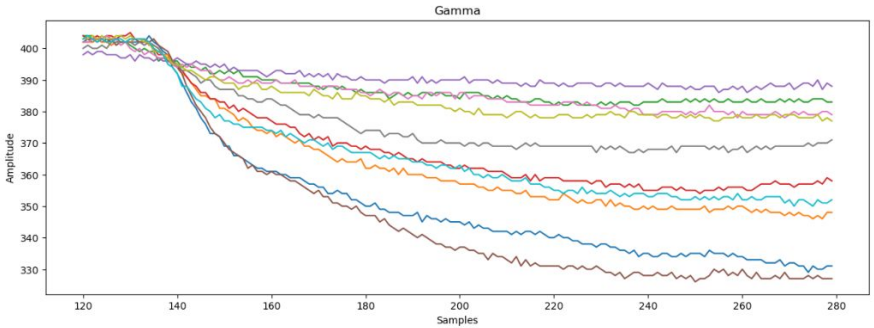


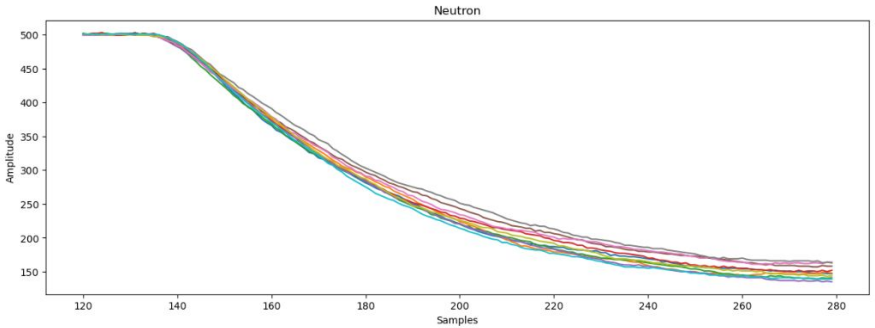


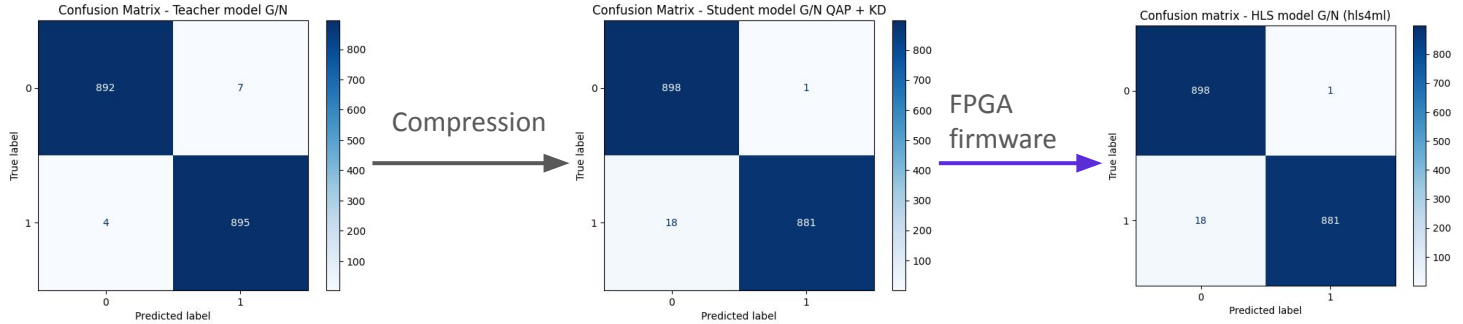
Edge AI based on FPGA

From trained model to real-time hardware inference

Gamma-Neutron Discrimination







Edge AI based on FPGA

From trained model to real-time hardware inference

HyperFPGA Cluster

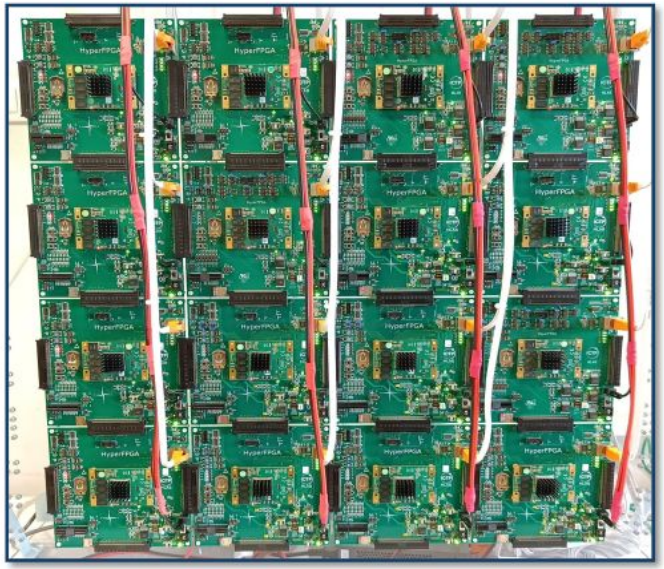
- 16 Nodos de MPSoC SOM with AMD Zynq UltraScale+
- Network: Ethernet connection TCP/IP, HP, HD GTH.
- Debian Linux OS

System-on-Module (Zynq UltraScale+ MPSoC-FPGA)

- CPU: Quad Arm Cortex, Dual Arm Cortex
- GPU: ARM Mali 400 MP2
- FPGA: ZU4EG

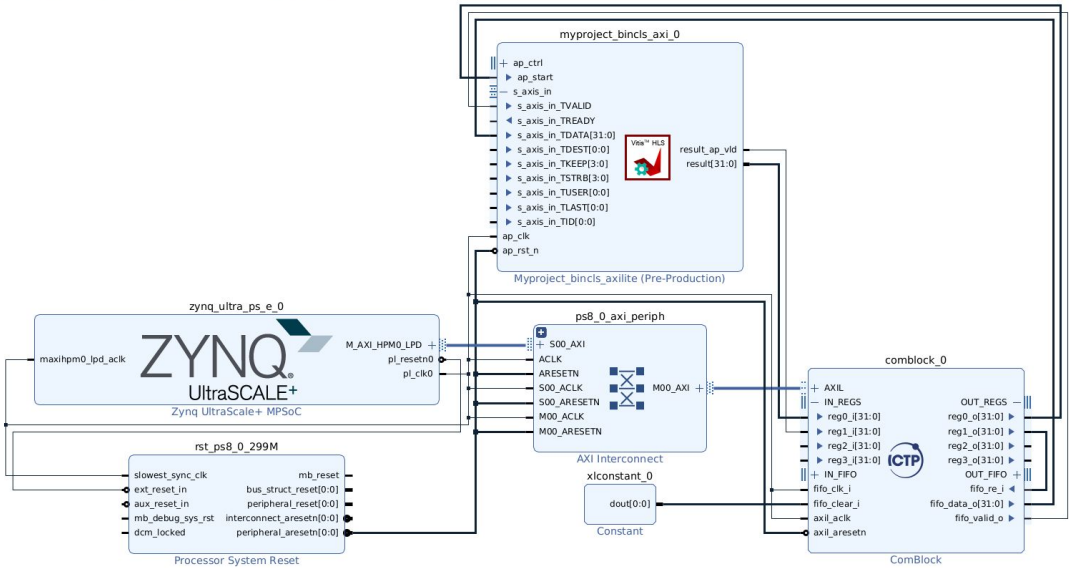
Carrier board

- x4 PCIe expansion port, Jtag, USB-UART
- microSD card
- 4 High-speed connectors for FPGA direct Interconnection



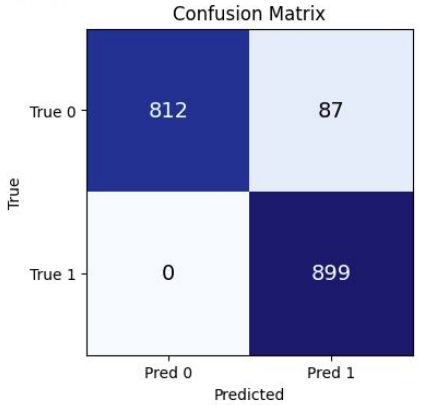
Edge AI based on FPGA

From trained model to real-time hardware inference




```
Confusion Matrix (rows=true, cols=pred)
[[812  87]
 [  0 899]]
Total: 1798

Accuracy : 0.9516
Precision: 0.9118
Recall   : 1.0000
F1-score : 0.9538
```



On Wednesday,
we build it together!

Edge AI and Machine Learning

Romina Soledad Molina, Ph.D.

School on Applied AI for Sustainable Development | smr 4210 | Trieste, Italy
March 2026