Del Algoritmo al Hardware:
Aprendizaje Automático en Sistemas Embebidos

From Algorithm to Hardware: Machine Learning in Embedded Systems

1 al 11 de Abril, 2025. Universidad Nacional de Mar del Plata - Mar del Plata - Argentina.

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -

Universidad Nacional de Mar del Plata

Fundación Williams

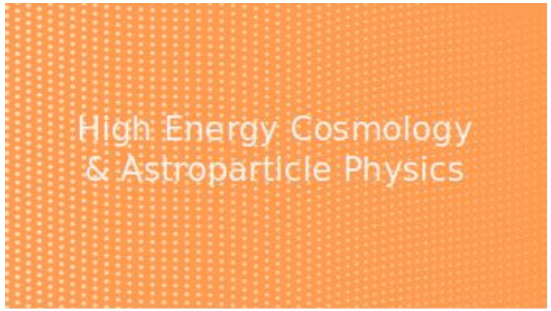
ICTP The Abdus Salam International Centre for Theoretical Physics

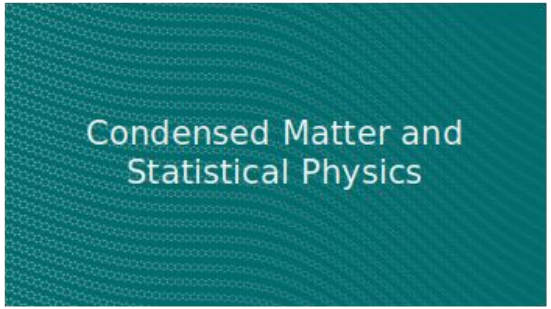
About ICTP

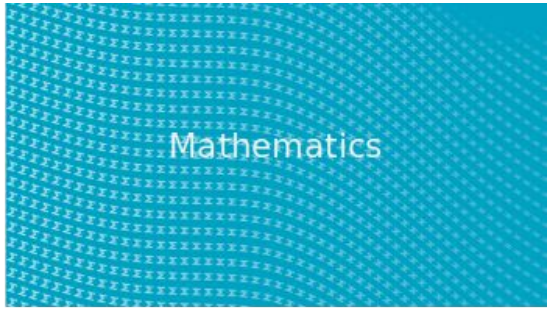
What is ICTP?

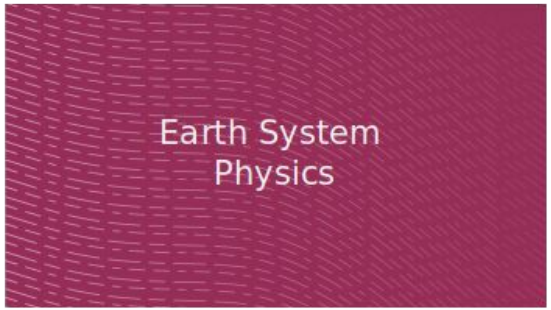
- Founded in 1964 by Nobel Laureate Abdus Salam to enhance international cooperation through science.
- Combines world-class research with a unique global mission of building science capacity in the developing world.
- Governed by tripartite agreement between Italy, UNESCO, and IAEA.
- Run by scientists for scientists.

Research Sections









Multidisciplinary Laboratory (MLab)

Research and Development of Advanced Scientific Instrumentation and Methods

- Particle Physics Experiments.
- Nuclear Applications.
- Applied Machine Learning.
- Multidisciplinary Experimental Projects.
- X-Ray Imaging for Cultural Heritage.
- Applied Optics and Lasers.

Multidisciplinary Laboratory (MLab)

Training

- PhD students (joint students with UniTS and STEP fellows).
- TRIL and postdoctoral fellows, associates and other visitors.
- Organized several schools and workshops on FPGA and SoC for Scientific Instrumentation, held at ICTP.
- Organized conferences and workshops in many countries: Argentina, Bangladesh, Brazil, Colombia, Costa Rica, Cuba, India, Malaysia, Mexico, Pakistan, Peru, among others.

Del Algoritmo al Hardware: Aprendizaje Automático en Sistemas Embebidos

From Algorithm to Hardware: Machine Learning in Embedded Systems

1 al 11 de Abril, 2025. Universidad Nacional de Mar del Plata - Mar del Plata - Argentina.

Course Presentation

Course presentation

Day	Time	Topic
01/04	17:30- 19:30	Machine Learning and FPGA: Evolution and Current State of These Technologies. Edge AI (R. M.)
02/04	10:00 - 11:00	Machine Learning: From Theory to Practice (R. M.)
	11:00 - 11:15	Coffee break
	11:15-13:00	Machine Learning: From Theory to Practice (R. M.)
		Machine Learning: From Theory to Practice (R. M.)
	16:00 - 19:00	Model Compression For Machine Learning-based Models: Pruning, Quantization, and Knowledge Distillation (R. M.)
	17:30 - 17:45	Coffee break
		Model Compression For Machine Learning-based Models: Pruning, Quantization, and Knowledge Distillation (R. M.)
03/04	13:00 - 14:00	System-On-Chip on based on FPGA: Architecture and workflow (R. M.)
	18:00 - 19:00	High-level synthesis (R. M.)
04/04	9:00 - 11:00	Hands-on: Deep Neural Network Training and Verification (R. M.)
	11:00 - 11:15	Coffee break
	11:15 - 12:00	Hands-on: Deep Neural Network Training and Verification (R. M.)
	12:00 - 13:00	Hands-on: Deep Neural Network Model Compresion (R. M.)
	18:00 - 19:00	Hands-on: SoC-based FPGA Bring-Up: "Hello World" (R. M.)
		Hands-on: High-level synthesis (R. M.)
07/04	13:00 - 14:00	High-level Synthesis for Machine Learning (hls4ml) (R. M.)
		Workflow for Deep Neural Network Deployment On Embedded Architectures (R. M.)
	14:00 - 14:15	Coffee break
	14:15 - 15:00	Hands-on: High-Level Synthesis for Machine Learning (hls4ml) (R. M.)
08/04	11:00 - 13:30	Communication Block (ComBlock) (M. B.)
		HyperFPGA: Enhancing Education with Remote Laboratory Access (M. B.)

Course presentation

	13:00 - 13:45	Lauch break
	13:45 - 15:00	Hands-on: High-Level Synthesis for Machine Learning (hls4ml) (R. M.)
09/04	10:00 - 13:00	Machine Learning and SoC-based FPGA for real-case applications (R. M.)
		Hands-on: Deploying Machine Learning on HyperFPGA and SoC-FPGA Boards (R. M.)
	16:00 - 17:30	Hands-on: Deploying Machine Learning on HyperFPGA and SoC-FPGA Boards (R. M.)
	17:30 - 17:45	Coffee break
	18:00 - 18:50	Overview of Embedded Platform Architectures and Key Hardware Components for Machine Learning Applications (N. J.)
	18:50 - 19:00	Break
	19:00 - 19:50	Methodological Approach to Designing Embedded Platforms for Machine Learning (N. J.)
	19:50 - 20:00	Break
	20:00 - 21: 00	Addressing Signal Integrity Challenges in Embedded Platforms for Machine Learning Applications (N. J.)
10/04	18:00 - 18:50	Managing Power Integrity Issues in Embedded Platforms for Machine Learning Applications (N. J.)
	18:50 - 19:00	Coffee break
	19:00 - 19:50	Optimizing Electromagnetic Compatibility (EMC) and Mitigating Electromagnetic Interference (EMI) in Embedded Platforms for Machine Learning Applications (N. J.)
	19:50 - 20:00	Break
	20:00 - 21: 00	Practical Hardware Design Considerations for Embedded Platforms in Machine Learning Applications (N. J.)
11/04	9:00-10:00	AMD Xilinx - AI Engines (G. S.) [Confirmar horario]
	10:00-11:00	Project: SoC-FPGA & Machine Learning: A Deep Dive into Different Workflows
	11:00 - 11:15	Coffee break
	11:15 - 14:00	Project: SoC-FPGA & Machine Learning: A Deep Dive into Different Workflows
		Project: SoC-FPGA & Machine Learning: A Deep Dive into Different Workflows
	14:00 - 15:00	Project: SoC-FPGA & Machine Learning: A Deep Dive into Different Workflows - Participant Presentations
	18:00 - 18:50	Practical Hardware Design Considerations for Embedded Platforms in Machine Learning Applications (N. J.)
	18:50 - 19:00	Coffee break
	19:00 - 19:50	Design Exercise: Develop Architecture, Select Components, and create PCB Floor Plan for Specified Machine Learning Platform Requirements (N. J.)
	19:50 - 20:00	Break
	20:00 - 21: 00	Interactive Discussion and Analysis of Participant-Proposed Solutions (N. J.)

Lecturers

Romina Soledad Molina, Ph.D. (R. M.) - International Centre for Theoretical Phyiscs
Nikola Jovalekic, Ph.D (N. J) - Teledyne Healthcare | X-Ray Solutions
Maynor Ballina, Ph. D student (M. B.) - International Centre for Theoretical Phyiscs and University of Trieste
Gustavo Sutter, Ph. D (G. S) - Universidad Autónoma de Madrid

Course presentation

- **Lecturers**
  - Romina Soledad Molina, Ph.D. - International Centre for Theoretical Phyiscs (MLab/STI).
  - Nikola Jovalekic, Ph.D. - Teledyne Healthcare | X-Ray Solutions.
  - Maynor Ballina, Ph. D. student - International Centre for Theoretical Phyiscs (MLab/STI) and University of Trieste.
  - Gustavo Sutter, Ph. D. - Universidad Autónoma de Madrid.

Course presentation

- Course material
  - https://github.com/RomiSolMolina/training-ML-FPGA-MarDelPlata-Argentina2025

- Hardware platforms:
  - Zedboard/PYNQ-Z1
  - HyperFPGA

- Approval conditions
  - Complete the exercises proposed during hands-on.
  - Completion of a final project and presentation.

Del Algoritmo al Hardware:
Aprendizaje Automático en Sistemas Embebidos

From Algorithm to Hardware: Machine Learning in Embedded Systems

1 al 11 de Abril, 2025. Universidad Nacional de Mar del Plata - Mar del Plata - Argentina.

Machine Learning and FPGA: Evolution, Current State of These Technologies, and Edge AI

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -

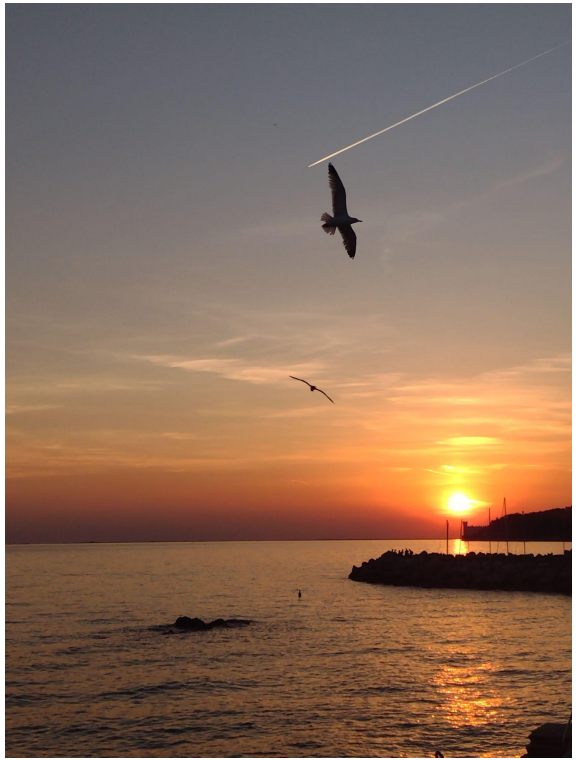
Universidad Nacional de Mar del Plata

Fundación Williams

ICTP The Abdus Salam International Centre for Theoretical Physics

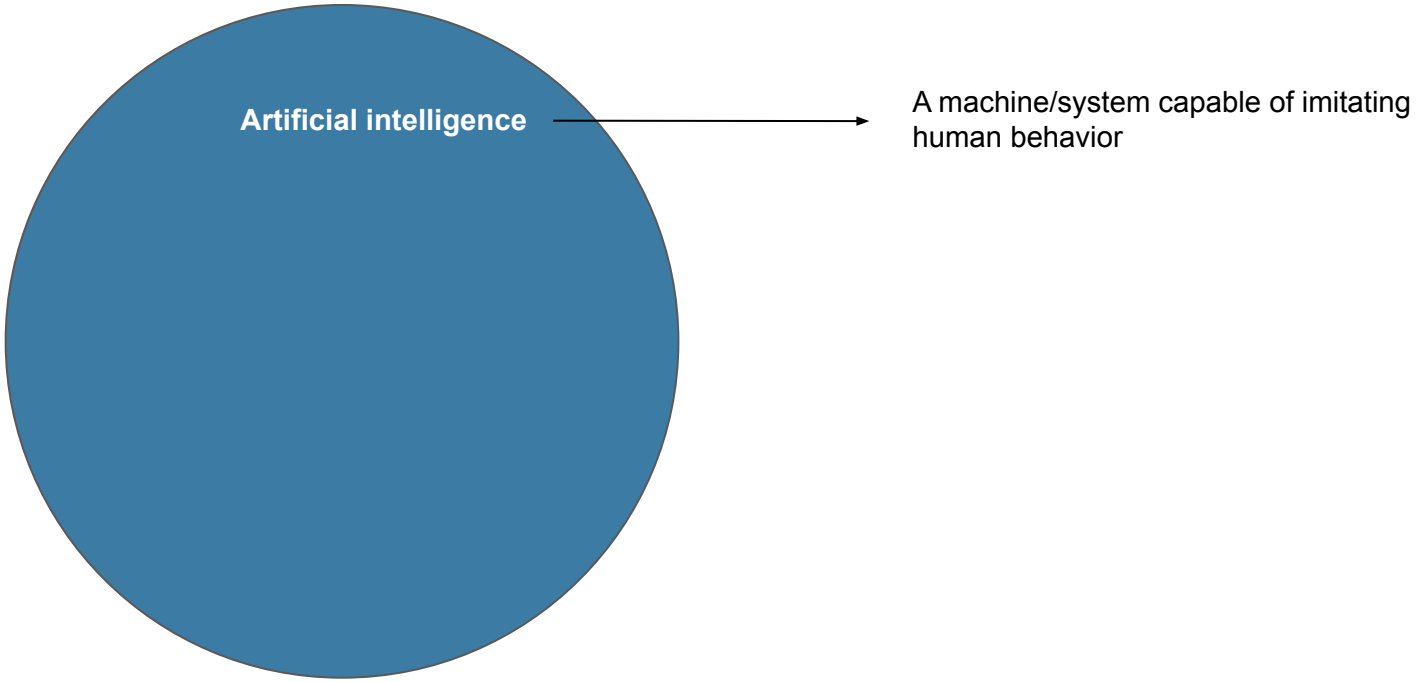
Outline

- Introduction.
- Edge AI.
- Remarks from the State-Of-The-Art.
- Optimizing every phase of the design and implementation process.
- MNIST-based binary classification.

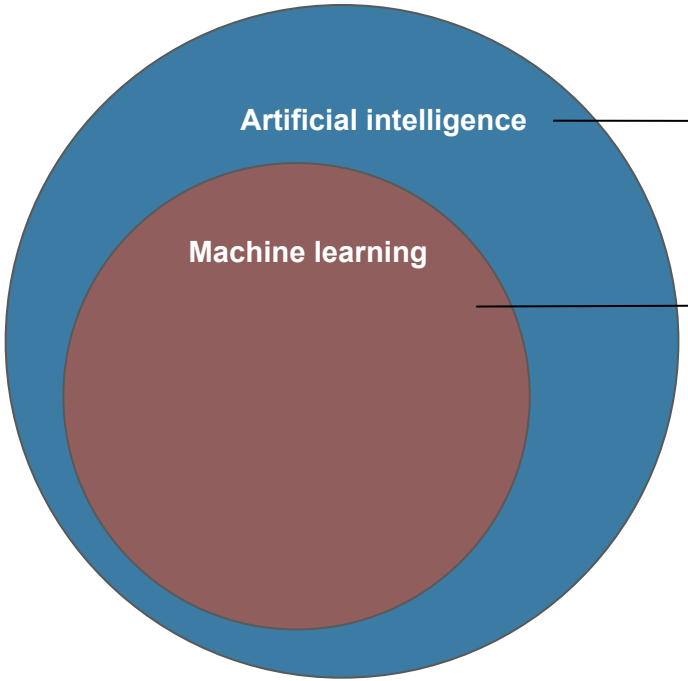


Introduction

Introduction



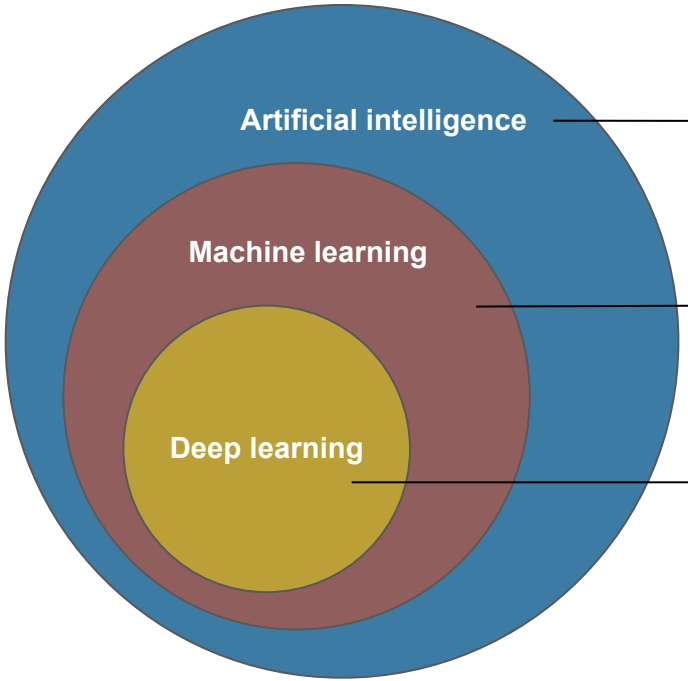
Introduction



A machine/system capable of imitating human behavior

Mathematics and statistics to learn from data.

Introduction

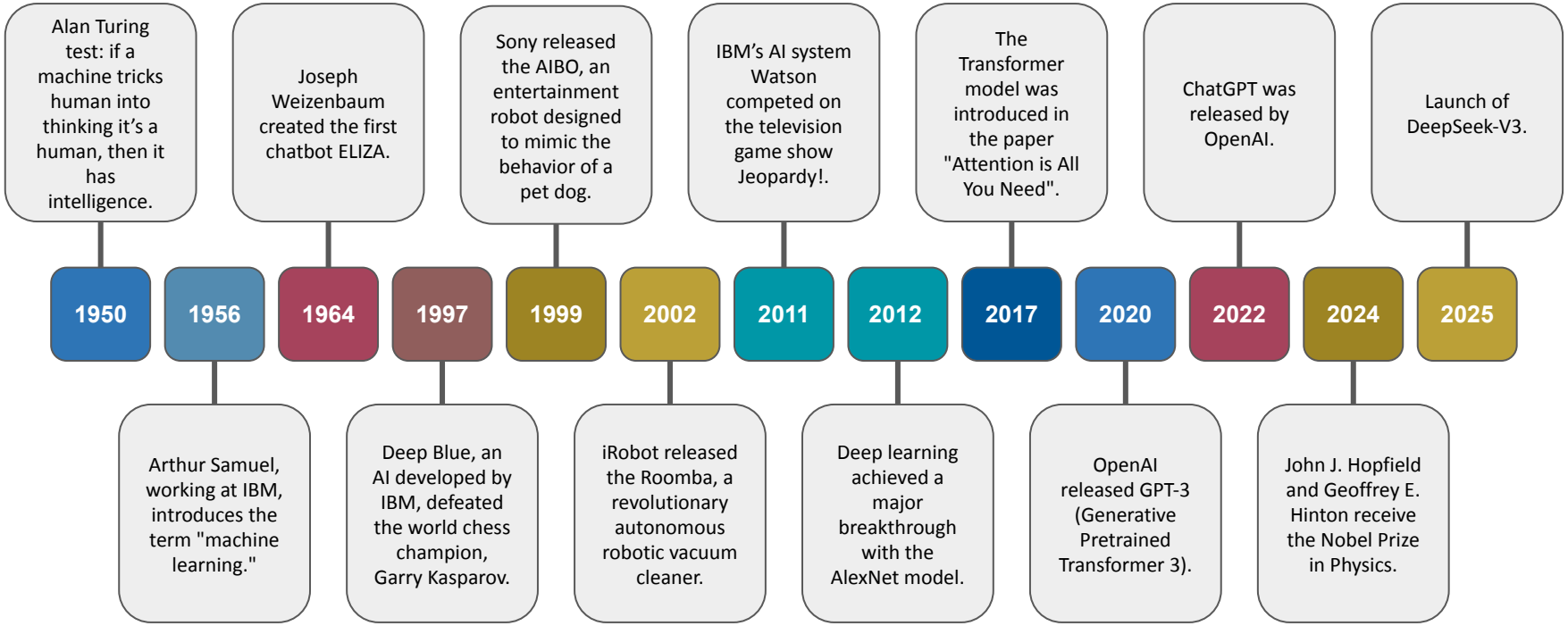


A machine/system capable of imitating human behavior

Mathematics and statistics to learn from data.

Neural networks multi-layer

Introduction



Introduction

- Why now? Three main components:

Introduction

- Why now? Three main components:

Big data

Introduction

- Why now? Three main components:

Big data

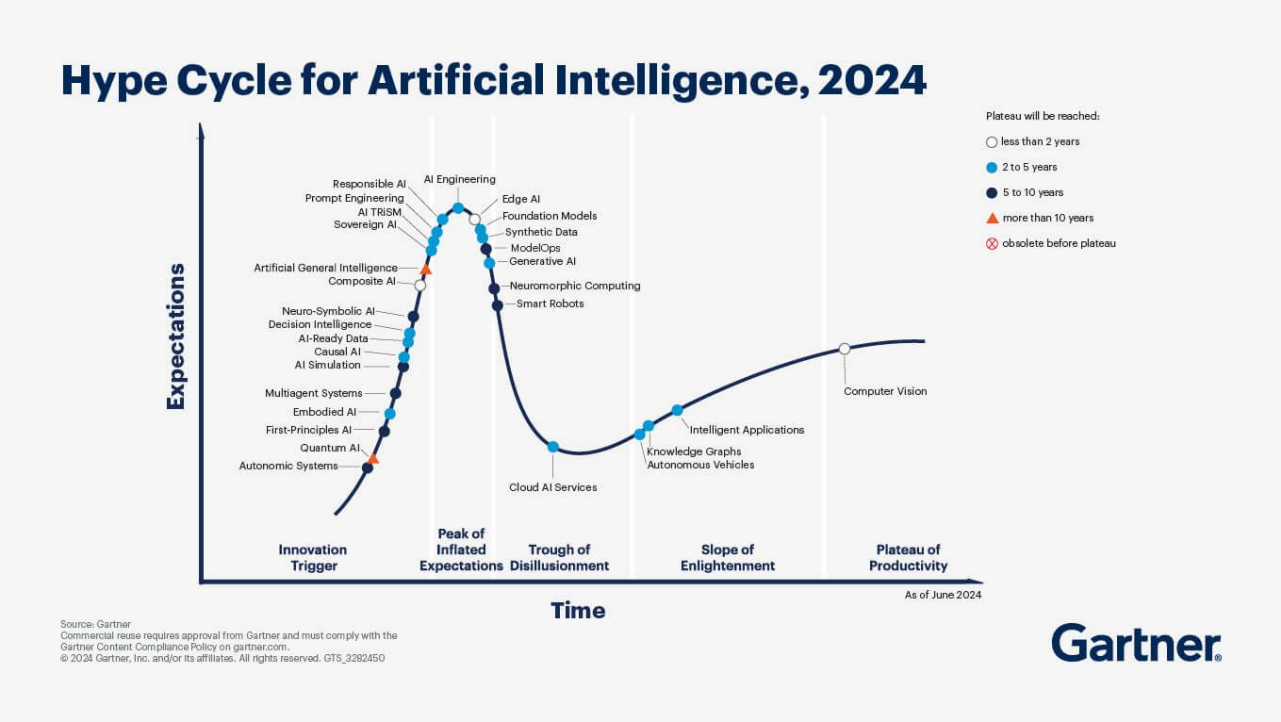
Software

Introduction

- Why now? Three main components:

Big data | **Software** | **Hardware**

Introduction



Graphically shows the maturity and adoption of technologies, helping to solve real problems and take advantage of opportunities.

Source: Gartner Hype Cycle on Artificial Intelligence (2023-2024) - https://www.gartner.com/en/articles/hype-cycle-for-artificial-intelligence

AI technologies

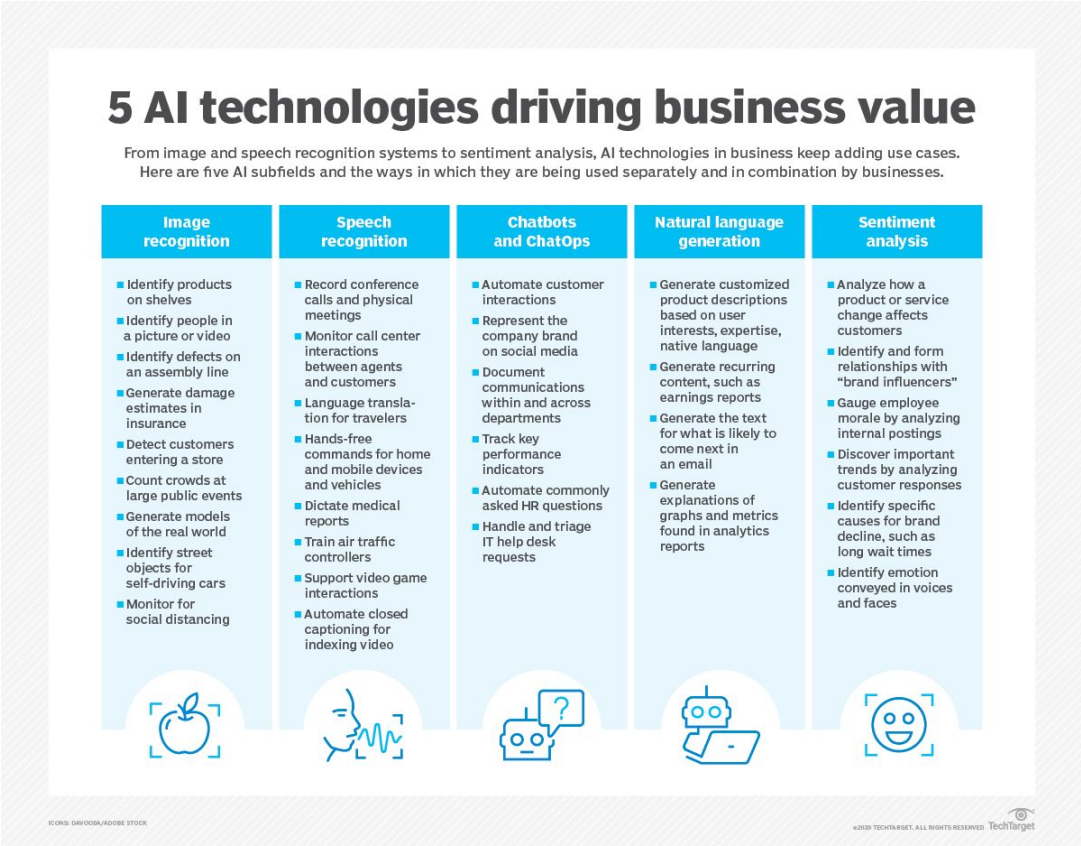


Image from https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- Advances in the following **technologies**:
  - Multimodal AI
  - AutoML
  - Embedded ML
  - MLOps
  - Low-code/No-code platforms
  - Reinforcement learning
  - Brain-computer interfaces
  - Neuromorphic processing
  - Digital Twins
  - Hardware platforms
  - Quantum computing
  - Among others

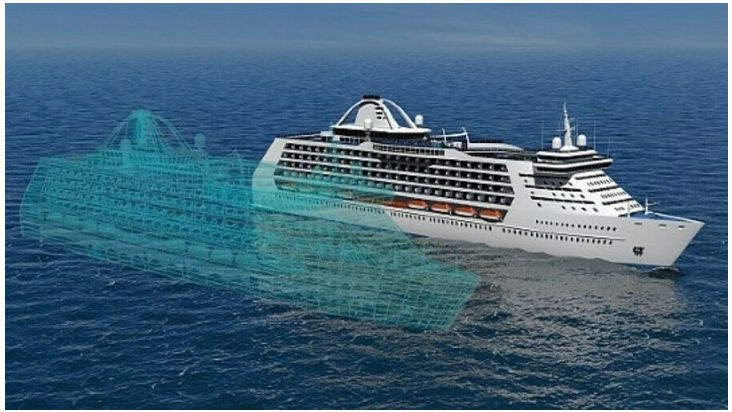


Image from
https://www.rivieramm.com/news-content-hub/news-content-hub/digital-twin-developed-to-model-green-ship-technologynbsp-59419

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- Advances in the following **areas**:
  - Games
  - Autonomous driving
  - Cybersecurity
  - Intelligent drones
  - Precision agriculture
  - Education
  - Renewable Energy

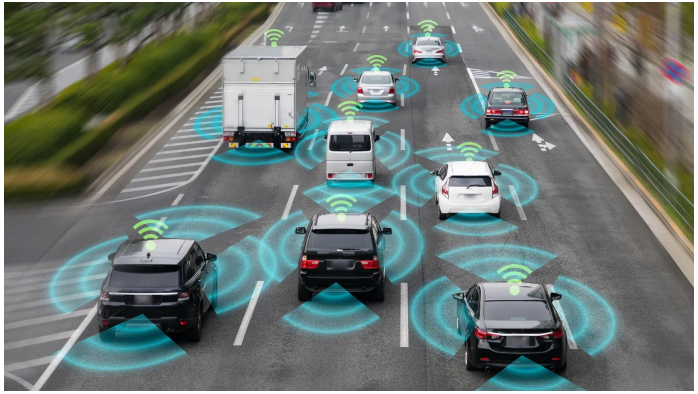


Image from
https://www.topgear.com/car%20news/what-are-sae-levels-autonomous-driving-uk

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

- **Business applications:**
  - Gartner predicts a significant share will embed **conversational AI**.
  - Some new applications will be **automatically generated by AI**, without human intervention.

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline

AI technologies beyond 2024

- **Market growth:**
  - The AI market is projected to grow at **35% annually**, surpassing **$1.3 trillion by 2030** (MarketsandMarkets).

- **Business applications:**
  - Gartner predicts a significant share will embed **conversational AI**.
  - Some new applications will be **automatically generated by AI**, without human intervention.

- **Impact on businesses & jobs:**
  - Business models and job roles may undergo **rapid and unpredictable transformations**.

From
https://www.techtarget.com/whatis/feature/History-and-evolution-of-machine-learning-A-timeline and https://www.marketsandmarkets.com/

AI technologies beyond 2024

- *Question: What do you think about the future of AI?*

AI-Specific Hardware Platforms

- **AI Chips & Processors**
  - **GPUs** (e.g., NVIDIA H100, AMD Instinct MI300) – Still the backbone of AI acceleration.
  - **TPUs** (Google Tensor Processing Units) – Optimized for deep learning workloads.
  - **NPUs** (Neural Processing Units) – Specialized for on-device AI in smartphones and edge devices.
  - **Analog & Optical AI Chips** – Emerging technology for ultra-low-power AI inference.

AI-Specific Hardware Platforms

- **AI Hardware Trends**
  - **Edge AI** – Custom low-power AI chips for real-time processing on IoT devices.
  - **RISC-V AI Processors** – Open-source architecture gaining traction for AI acceleration.
  - **Neuromorphic Computing** – Brain-inspired AI hardware mimicking human neural networks.

AI-Specific Hardware Platforms

- *Question: What do you think about FPGA and AI?*

AI-Specific Hardware Platforms

- **FPGA+AI Trends**
  - **AI-Specific FPGA Architectures** – New FPGA models optimized for deep learning (Xilinx Versal AI Core, Intel Stratix 10 NX, and Lattice Avant AI).
  - **Quantum + FPGA Hybrid Computing** – Research into integrating FPGA with quantum acceleration.
  - **FPGA as-a-Service (FaaS)** – Cloud-based FPGA solutions for scalable AI workloads.
  - **Embedded AI & Edge Computing** – FPGAs in IoT & industrial AI, enabling real-time decision-making with ultra-low power.

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]

[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]

On the edge 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]

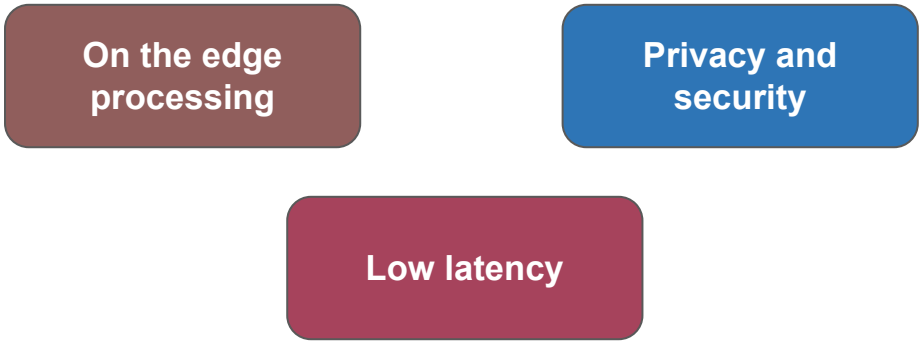
On the edge processing

Low latency

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

Edge AI

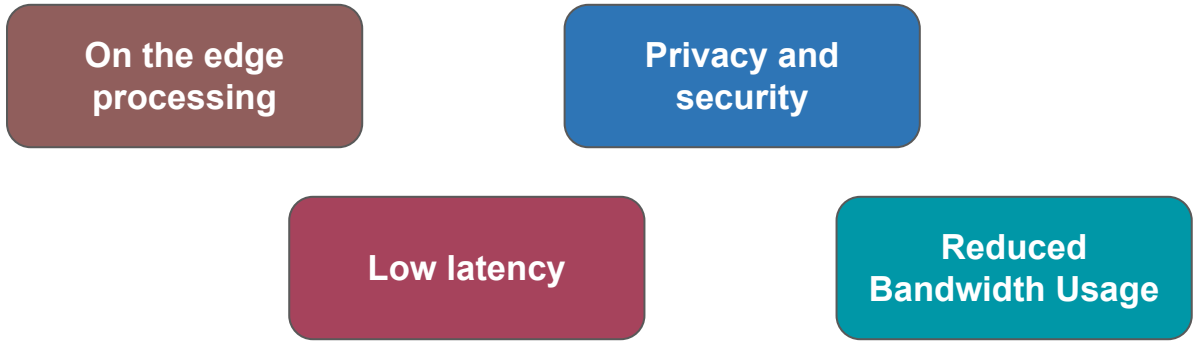
“Edge artificial intelligence (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]



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

Edge AI

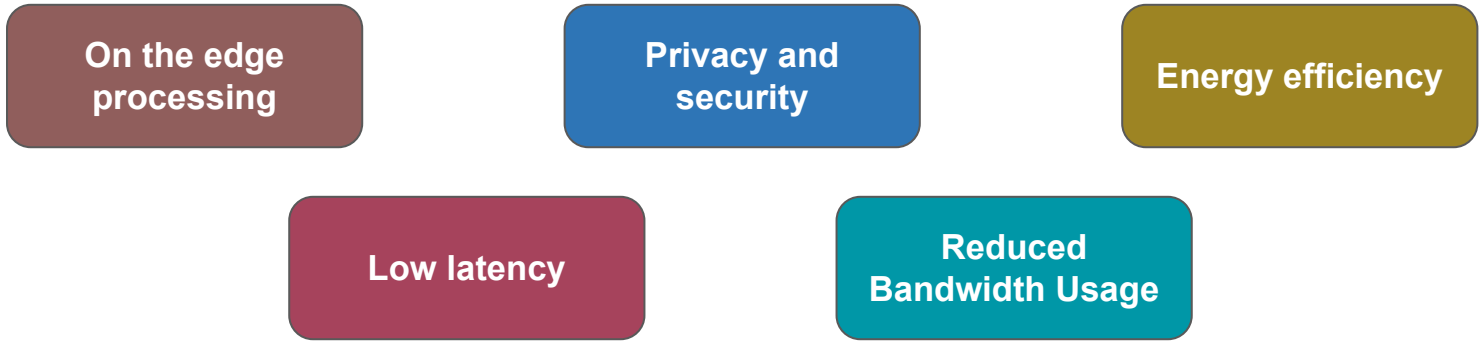
“Edge artificial intelligence (Edge AI) refers to the deployment of AI algorithms and AI models directly on local edge devices such as sensors or Internet of Things (IoT) devices, which enables real-time data processing and analysis without constant reliance on cloud infrastructure.” [IBM]



[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]



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

Edge AI

Challenges

- Limited computational resources.
- Memory and storage restrictions.
- Power constraints.
- Latency and real-time processing.
- Data privacy and security.
- Hardware heterogeneity.
- Network connectivity issues.

Edge AI

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

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

- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.

Edge AI

- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.

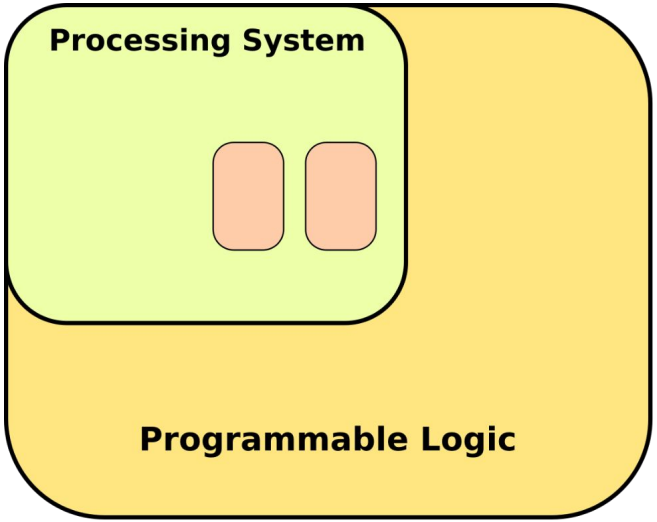
- **GPU:** accelerates parallelizable tasks such as deep learning inference, image processing, and high-performance computing.

Edge AI

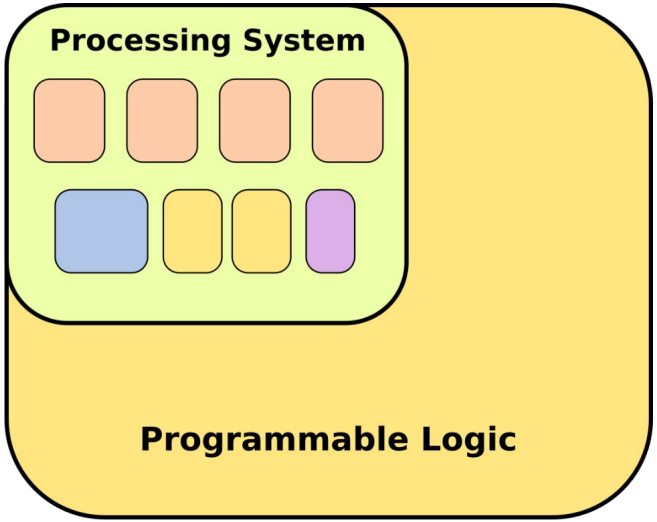
- **CPU:** handles general-purpose tasks like system management, control logic, and lightweight AI inference.
- **GPU:** accelerates parallelizable tasks such as deep learning inference, image processing, and high-performance computing.
- **FPGA:** provides real-time, ultra-low-latency processing for tasks like sensor fusion, encryption, and custom AI accelerators.

Edge AI

SoC-based FPGA



Zynq



Zynq MPSoC

Edge AI based on FPGA

FPGA / SoC-based on FPGA

Edge AI based on FPGA

Low latency

FPGA / SoC-based on FPGA

Edge AI based on FPGA

Low latency

Energy Efficiency

FPGA / SoC-based on FPGA

Edge AI based on FPGA

Low latency

Energy Efficiency

High parallelism

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability
- Customizable AI Acceleration

FPGA / SoC-based on FPGA

Edge AI based on FPGA

- Low latency
- Energy Efficiency
- High parallelism
- Scalability
- Customizable AI Acceleration

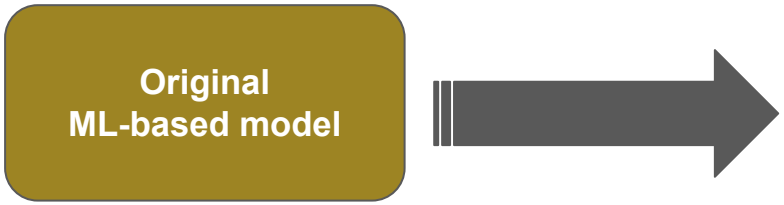
FPGA / SoC-based on FPGA

Resource-constrained devices

Edge AI based on FPGA

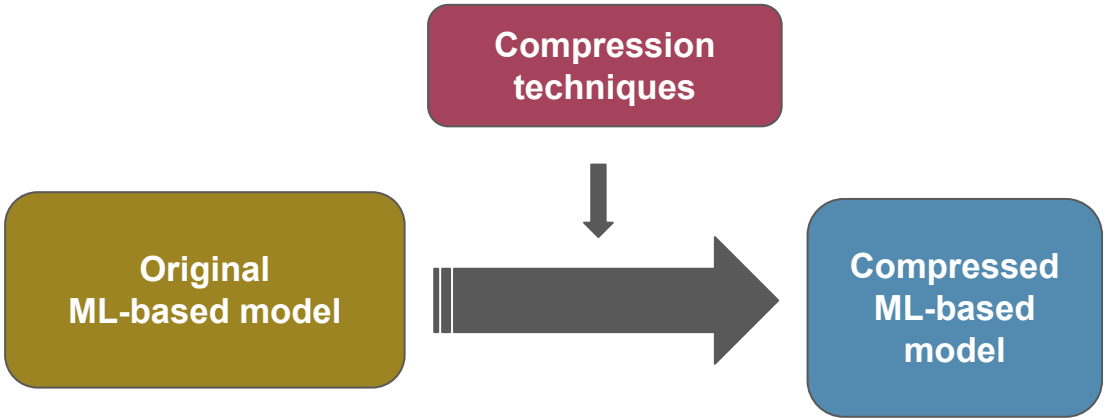
Original
ML-based model

Edge AI based on FPGA

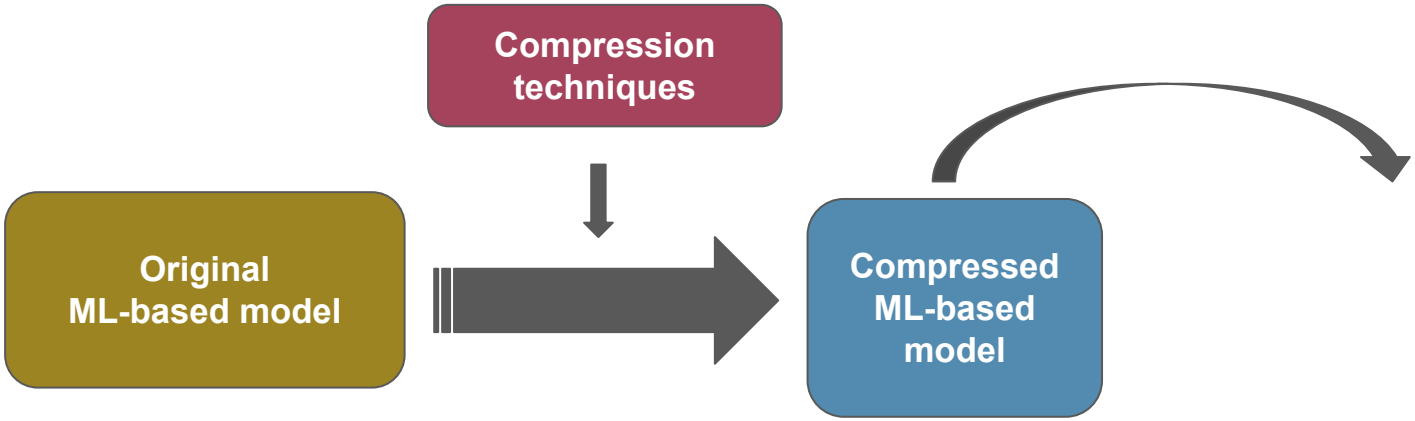


Edge AI based on FPGA

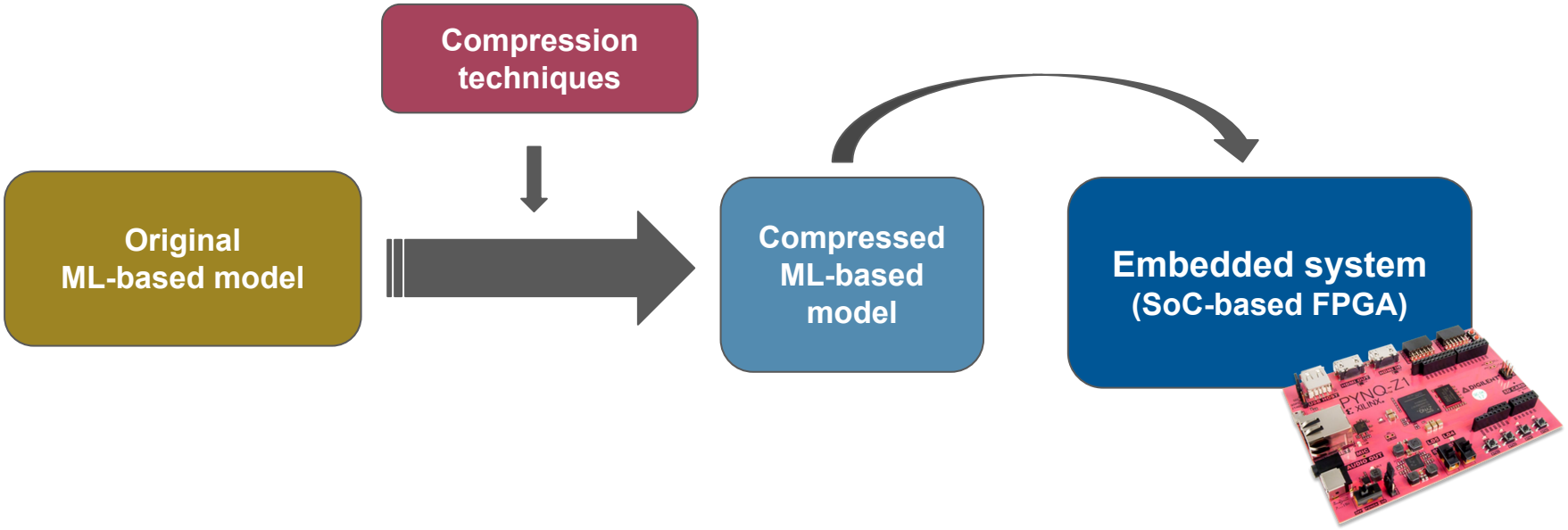
Edge AI based on FPGA



Edge AI based on FPGA



Edge AI based on FPGA



Remarks from the State-Of-The-Art

Remarks from the State-Of-The-Art

. Towards ML-based models implemented on resource-constrained devices.

Remarks from the State-Of-The-Art

. Towards ML-based models implemented on resource-constrained devices.

. SOTA models: VGG16, MobileNet V2, BERT, U-Net, YOLO.

Remarks from the State-Of-The-Art

. Towards ML-based models implemented on resource-constrained devices.

. SOTA models: VGG16, MobileNet V2, BERT, U-Net, YOLO.

. Compression: focused on pruning and quantization.

Remarks from the State-Of-The-Art

. Towards ML-based models implemented on resource-constrained devices.

. SOTA models: VGG16, MobileNet V2, BERT, U-Net, YOLO.

. Compression: focused on pruning and quantization.

. Workflows addressing some parts of the development cycle.

Remarks from the State-Of-The-Art

. Towards ML-based models implemented on resource-constrained devices.

. SOTA models: VGG16, MobileNet V2, BERT, U-Net, YOLO.

. Compression: focused on pruning and quantization.

. Workflows addressing some parts of the development cycle.

. Off-chip memory transactions.

Remarks from the State-Of-The-Art

Current state of the Tool-flows

TABLE I
COMPARISON OF TOOL-FLOWS

Tool-Flow	Platform	NN[1]	Compression[2]	Active[3]
HLS4ML [6]	Xilinx	F,C,R	P,Q	Yes
FINN [7]	Xilinx	F,C,R	P,Q	Yes
Vitis AI [8]	Xilinx	F,C,R	P,Q	Yes
VTA [9]	Xilinx	F,C	P,Q	Yes
Matlab DLP [10]	Xilinx	F,C	P,Q	Yes
OpenVino [11]	Intel	F,C	P,Q	Yes
NNGEN [12]	Xilinx	F	Q	No
CFU Playground [13]	Xilinx	C	-	Yes
VeriGOOD-ML [14]	Xilinx	C	-	Yes
DNNWeaver [15]	Xilinx	C	-	No
DL2HDL [16]	Xilinx	F,R	-	No
FPGAConvnet [17]	Xilinx	C	-	No
FINN-L [18]	Xilinx	LSTM	-	No
LeFlow [19]	Both	F,C	-	No
CaFGPA [20]	Xilinx	C	-	No
DNN Builder [21]	Xilinx	C	-	No
FP-DNN [22]	Xilinx	C	-	No
Snowflake [23]	Xilinx	C	-	No
FFTCodeGen [24]	Xilinx	C	-	No
Haddoc2 [25]	Xilinx	C	-	No
Angel-Eye [26]	Xilinx	C	-	No
Caffein [27]	Xilinx	C	-	No

[1] Supported NN architectures : F - Fully Connected Neural Network, C - Convolutional Neural Network, R - Recurrent Neural Network.
[2] Supported compression schemes : P - Pruning, Q Quantization.
[3] The tool-flow still receives regular updates.

Table from [1] Rahimifar, M. M., Granger, C. É., Wingering, Q., Gouin-Ferland, B., Rahali, H. E., & Therrien, A. C. (2022, November). A survey of machine learning to fpga tool-flows for instrumentation. In *2022 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)* (pp. 1-4). IEEE.

Remarks from the State-Of-The-Art

Memory footprint and latency

Remarks from the State-Of-The-Art

Memory footprint and latency

Ensemble of compression techniques

Remarks from the State-Of-The-Art

Memory footprint and latency

Ensemble of compression techniques

On-chip memory deployment

Remarks from the State-Of-The-Art

- Memory footprint and latency
- Ensemble of compression techniques
- On-chip memory deployment
- End-to-end workflow

Remarks from the State-Of-The-Art

- Memory footprint and latency
- Ensemble of compression techniques
- On-chip memory deployment
- End-to-end workflow
- Productivity

Remarks from the State-Of-The-Art

- *Question: What other features or challenges would you include?*

Optimizing every phase of the design and implementation process.

Optimizing every phase of the design and implementation process.

Software development

Model training

Model compression

Optimizing every phase of the design and implementation process.

Software development

- Model training
- Model compression

HW platform selection

- Edge devices
- Cloud
- GPU/CPU
- Others

Optimizing every phase of the design and implementation process.

Software development	HW platform selection	Firmware creation
Model training	Edge devices	C/C++ application
Model compression	Cloud	Python application
	GPU/CPU	HDL
	Others	

Optimizing every phase of the design and implementation process.

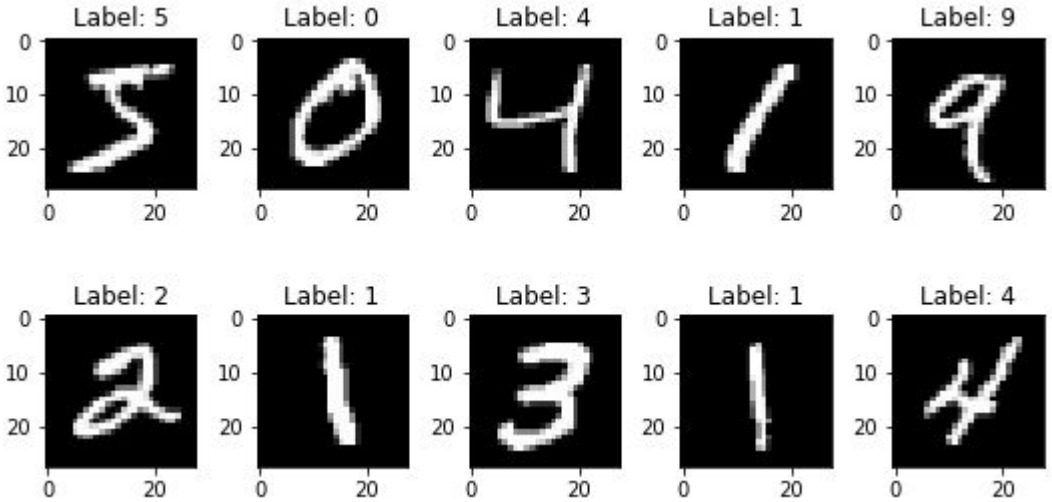
Software development	HW platform selection	Firmware creation	PCB design
Model training	Edge devices	C/C++ application	Key components
Model compression	Cloud	Python application	Signal and power integrity
	GPU/CPU	HDL	Methodological design
	Others		EMC and EMI

Demo:
MNIST-based binary classification

ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

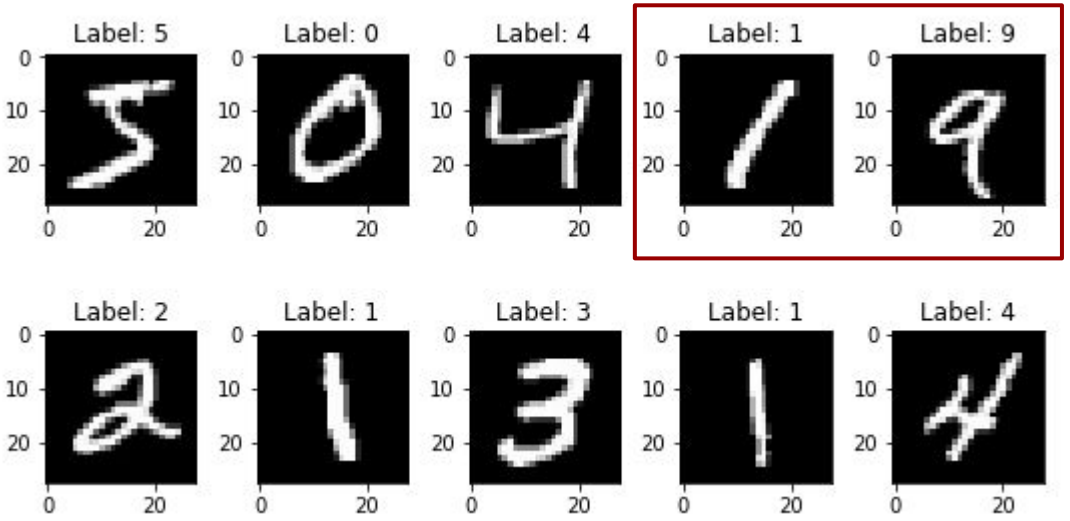
MNIST-based binary classification



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification

- **Quantization-Aware Pruning**
  - 8-bits fixed point precision
  - 20% target sparsity
  - QKeras for model definition

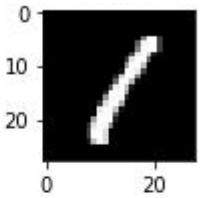
Layer (type)	Output Shape	Param #
fc1_input (QDense)	(None, 5)	3925
relu_input (QActivation)	(None, 5)	0
fc1 (QDense)	(None, 7)	42
relu1 (QActivation)	(None, 7)	0
fc2 (QDense)	(None, 10)	80
relu2 (QActivation)	(None, 10)	0
output (QDense)	(None, 2)	22
sigmoid (Activation)	(None, 2)	0

Total params: 4,069
Trainable params: 4,069
Non-trainable params: 0

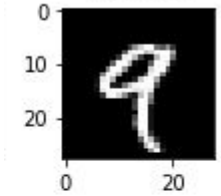
ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

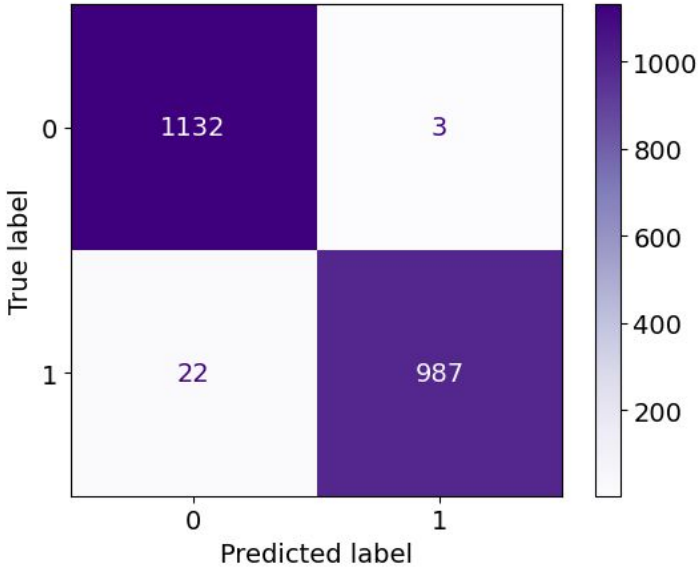
MNIST-based binary classification



Class 0



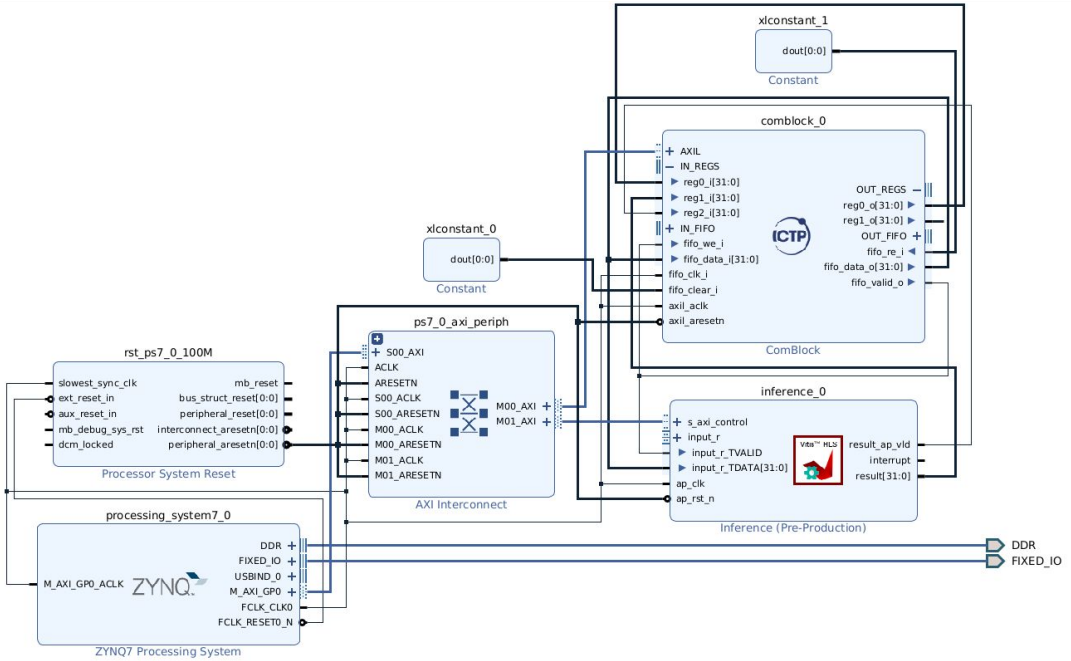
Class 1



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

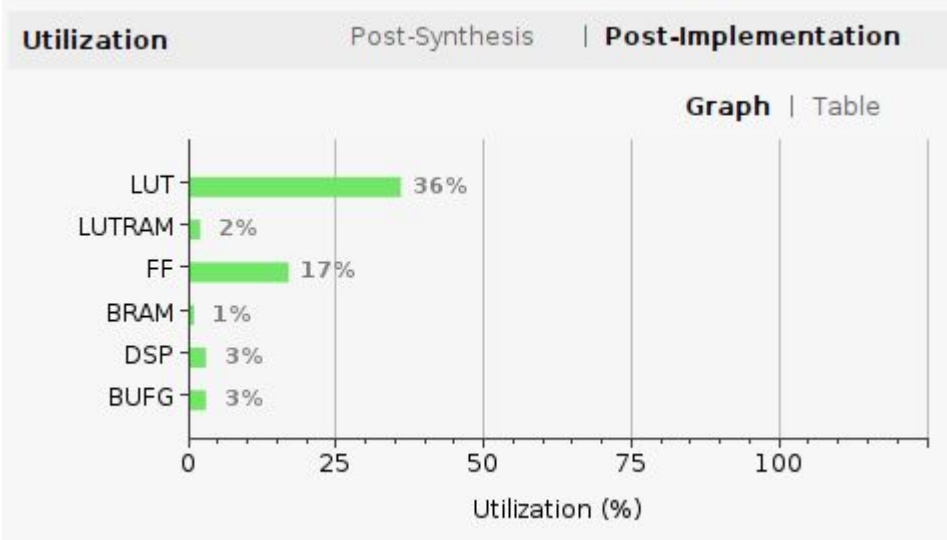
MNIST-based binary classification



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

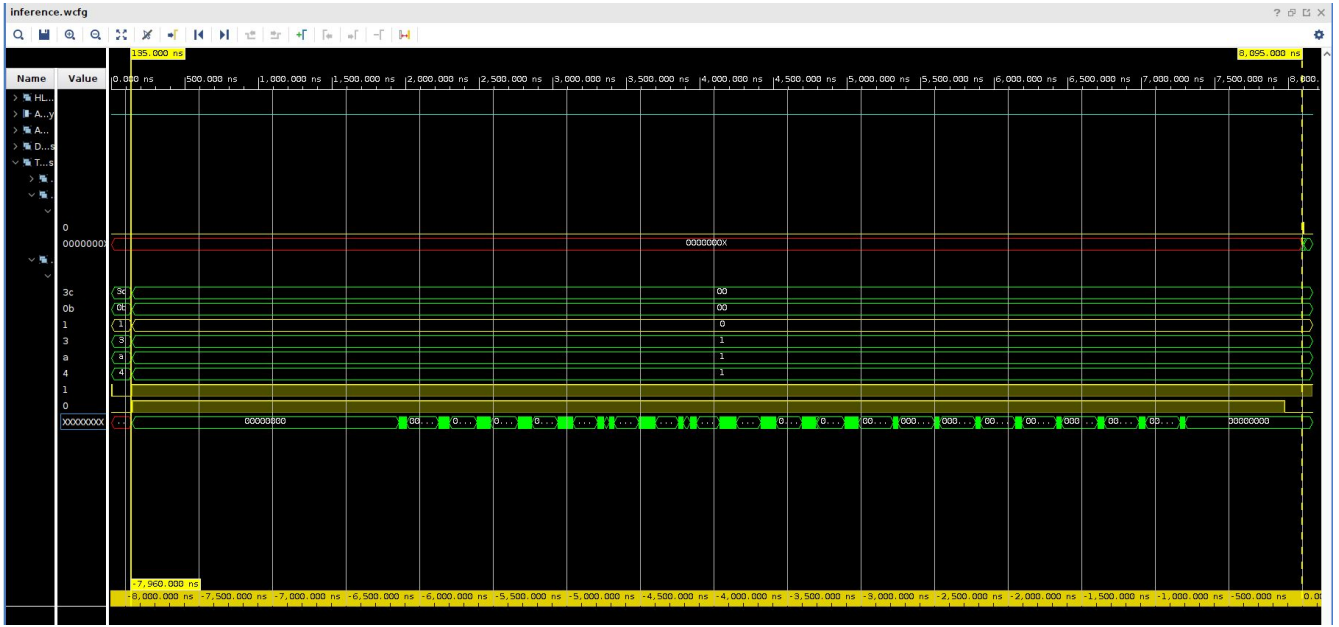
MNIST-based binary classification



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification



ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification

IP core based on ML integrated with PYNQ framework

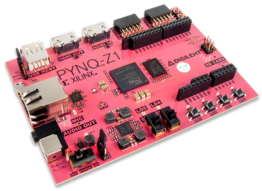
In [ ]:
```
from pynq import Overlay
from pynq import MMIO
import comblock as cbc

import numpy as np
import matplotlib.pyplot as plt
```

Load Overlay

In [ ]:
```
# Load the overlay (bitstream) onto the FPGA.

ol = Overlay("design_1_wrapper.xsa")
```

The information from the **comblock_0** block is read to verify everything that is obtained. Since the object is mapped to AXI Lite, it is noted that the AXI Full address is omitted.

In [ ]:
```
## Overlay information

ol.ip_dict
```



PYNQ™

ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

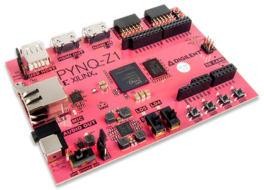
MNIST-based binary classification

ComBlock information

ComBlock for PYNQ: https://github.com/Mballina42/PynQ_ComBlock

For convenience, the *comblock.py* Python script is established which contains useful constants for interacting with the ComBlock.

```
In [ ]: ol.ip_dict['comblock_0']

In [ ]: # The object is created based on the comblock_0 IP
        cb = ol.comblock_0
```



PYNQ

ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification

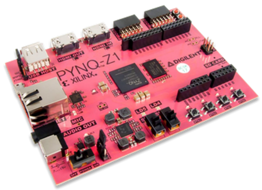
Data preparation

```
In [ ]: signal_1 = [

            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

        ]
```

```
In [ ]: imageArray = np.array(signal_1)

        image_2d = imageArray.reshape((28, 28))

        # Display as an image
        plt.imshow(image_2d, cmap='gray', interpolation='nearest')
        plt.colorbar()  # Optional: Show color scale
        plt.show()
```

```
In [ ]: signal_2 = [

            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

        ]
```

```
In [ ]: imageArray = np.array(signal_2)

        image_2d = imageArray.reshape((28, 28))

        # Display as an image
        plt.imshow(image_2d, cmap='gray', interpolation='nearest')
        plt.colorbar()  # Optional: Show color scale
        plt.show()
```



PYNQ™

ML and model compression techniques for SoC/FPGA

Demo: MNIST-based binary classification

MNIST-based binary classification

Interacting with ComBlock

Signal 1

Write FIFO - Send image to the FPGA

```
In [ ]: cb.write(cbc.OREG1, 1)

        # Send data to the ComBlock's FIFO
        data_size = 28*28
        for i in range(data_size):
               cb.write(cbc.OFIFO_VALUE, signal_1[i])
```

Read registers - Read inference result from the FPGA

```
In [ ]: # Read IREG1 to obtain the result of the inferece process
        cb.read(cbc.IREG1)
```

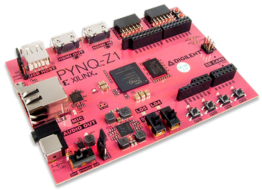
Signal 2

Write FIFO - Send image to the FPGA

```
In [ ]: cb.write(cbc.OREG1, 1)

        # Send data to the ComBlock's FIFO
        data_size = 28*28
        for i in range(data_size):
               cb.write(cbc.OFIFO_VALUE, signal_2[i])
```

Read registers - Read inference result from the FPGA

```
In [ ]: cb.read(cbc.IREG1)
```



PYNQ™

Thank you!

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -