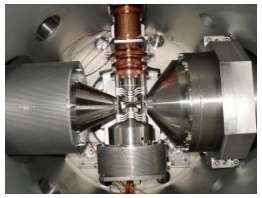
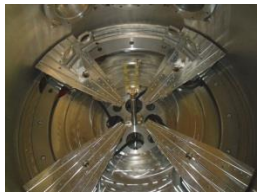


laboratoire de physique corpusculaire

EXEMPLE D'APPLICATION DANS LE PROJET FASTER

David Etasse

cnrs

ENSICAEN
École Nationale Supérieure d'Ingénieurs de Caen

Normandie Université

UNICAEN

AGENDA

FASTER

1. FASTERV2 → Real time acquisition system
2. FASTERV3 → Real Time acquisition platform
3. NVIDIA Jetson modules

THINK

1. AI vs ML vs ANN vs DL
2. THINK challenges
3. THINK challenges benchmark
4. From Pytorch to TensorRT with ONNX
5. First results

FASTER-V2

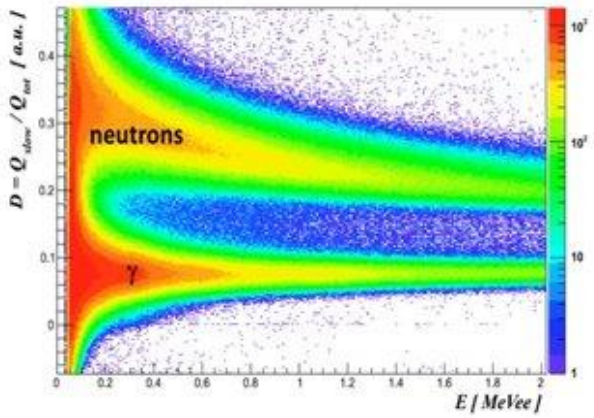
Offline Analysis

Ubuntu repository

Real Time Algorithms

Modular Electronic

RHB

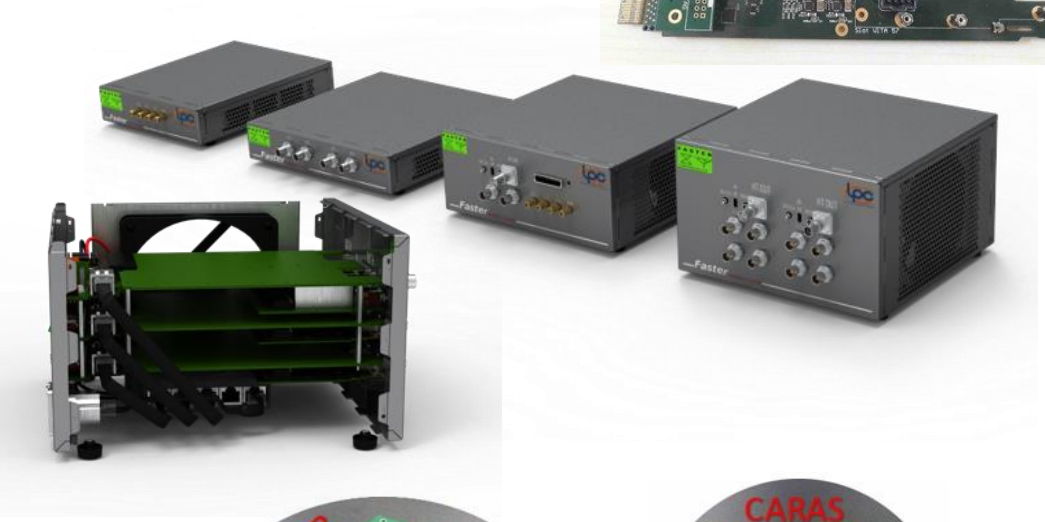


Based on Root

Carniol Benjamin, **Chaventré** Thiérry, **Cussol** Daniel,
Etasse David, **Fontbonne** Cathy, **Fontbonne** Jean-Marc,
Harang Julien, **Hommet** Jean, **Langlois** Jérome, **Poincheval** Jérome

HARDWARE

STANDALONE SYSTEM



MULTI-CHANNEL SYSTEM

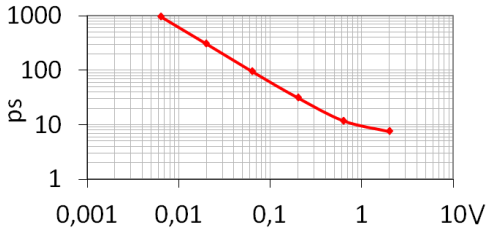
- 4 FADC (125MHz@14bits)
- ±1V, ±2V, ±5V, ±10V input range
- 25 MHz Bandwidth

- 2 FADC (500MHz@12bits)
- ±1V input range
- ±1V input adjustable Offset
- 100 MHz Bandwidth

- DDC316 from TI
- 32 channels
- I-TO-V conversion front end
- 3pC to 12 pC (full scale)
- Integration time range from 10us to 10 ms

- ISEG BPS-Serie - 4W
- ± 500 V to ± 6 KV

FIRMWARE



Time resolution VS signal amplitude

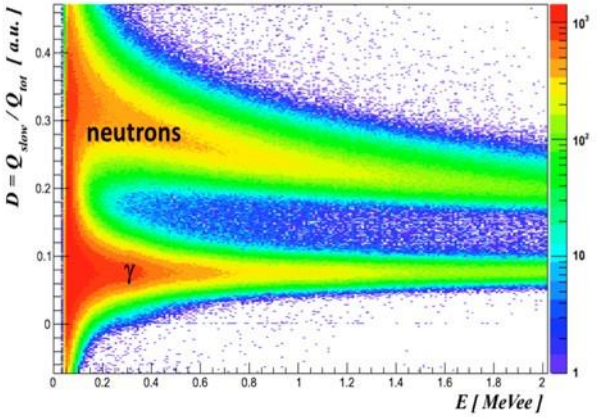

FASTER-CRRC4
FASTER-TRAPEZ



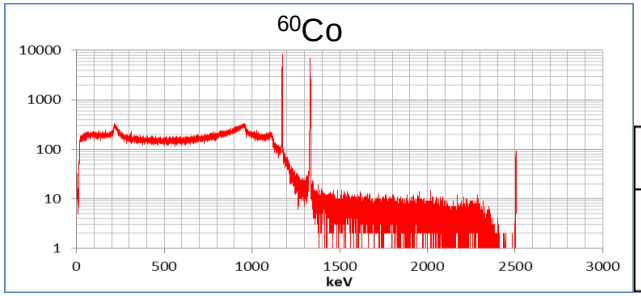
FASTER-QDC-TDC$_{HR}$
FASTER-RF
FASTER-SCALER
FASTER-SAMPLER

FASTER-ELECTROMETER


FASTER_HV



Demon detector, CARAS board,
FASTER-QDC-TDC$_{HR}$



pic keV	FWHM keV
1173,21	1,71
1332,48	1,90
2505,69	2,41

HPGe detector, MOSAHR board, FASTER_ADC

SOFTWARE

- Ubuntu 16.04 or 18.04 or 20.04 LTS 64 bits
- ADA, C and Python
- Software trigger (multiplicity or Boolean trigger)
- Faster repository on LPC Server
    - sudo apt_get install fasterv2
- Update the software and the FIRMWARE at the same time
- Offline analysis package

AGENDA

FASTER

1. FASTERV2 → Real time acquisition system
2. FASTERV3 → Real Time acquisition platform
3. NVIDIA Jetson modules

THINK

1. AI vs ML vs ANN vs DL
2. THINK challenges
3. THINK challenges benchmark
4. From Pytorch to TensorRT with ONNX
5. First results

FASTER-V3

A New real time, digital acquisition platform

Same real time FPGA algorithms as FASTER-V2

+

User algorithms

+

News digitizers (16bits@125 Msps-> 10bits@5 Gsps)

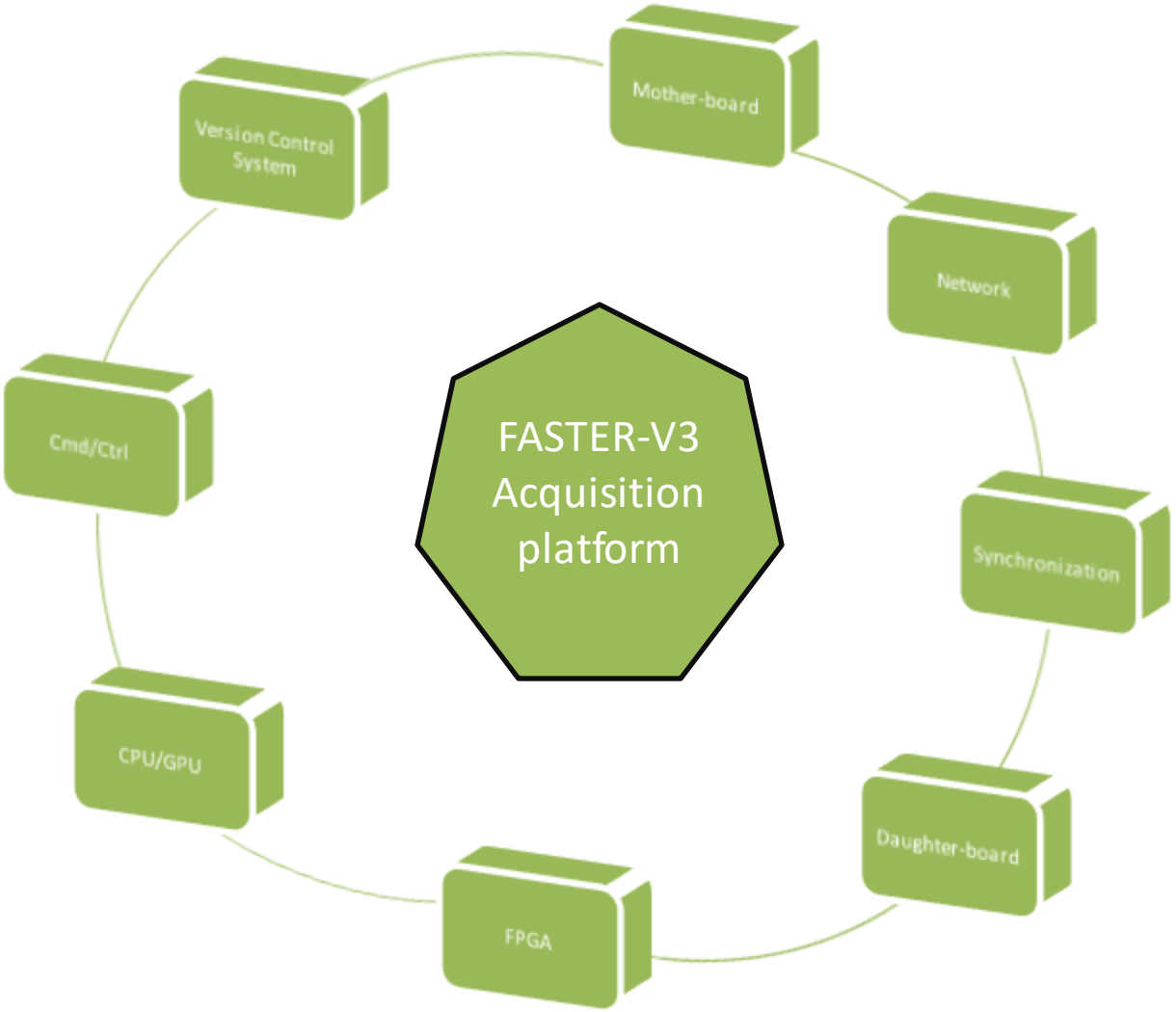
+

1588 V2.1 and SyncE

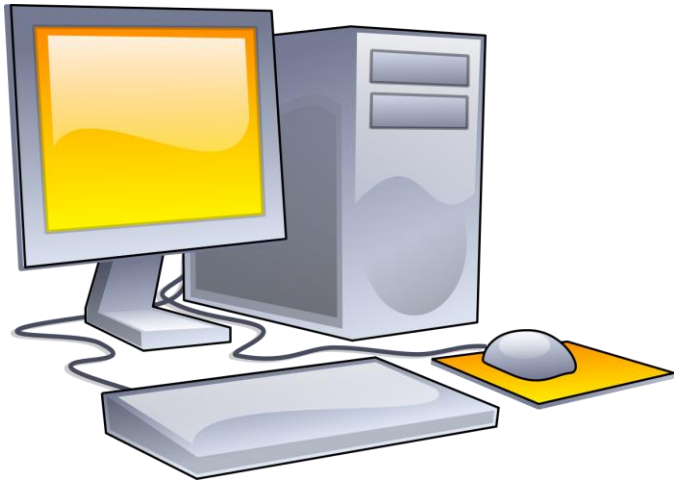
+

Epics compatible

FASTER-V3


FASTER-V3
Acquisition
platform
Mother-board
Network
Synchronization
Daughter-board
FPGA
CPU/GPU
Cmd/Ctrl
Version Control
System

CPU-GPU

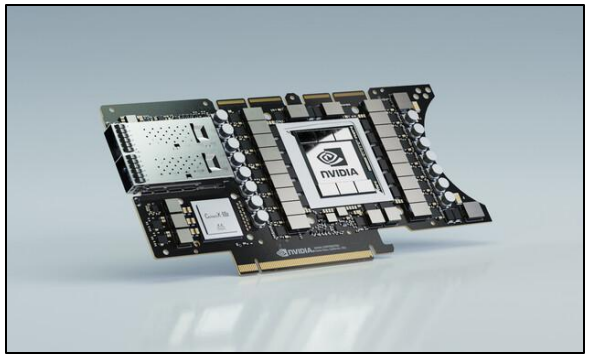


=

+

Software
Environnement
- Ubuntu 20.04
- Cuda 10
- PyTorch
- TensorFlow
- TensorRT
- C/C++

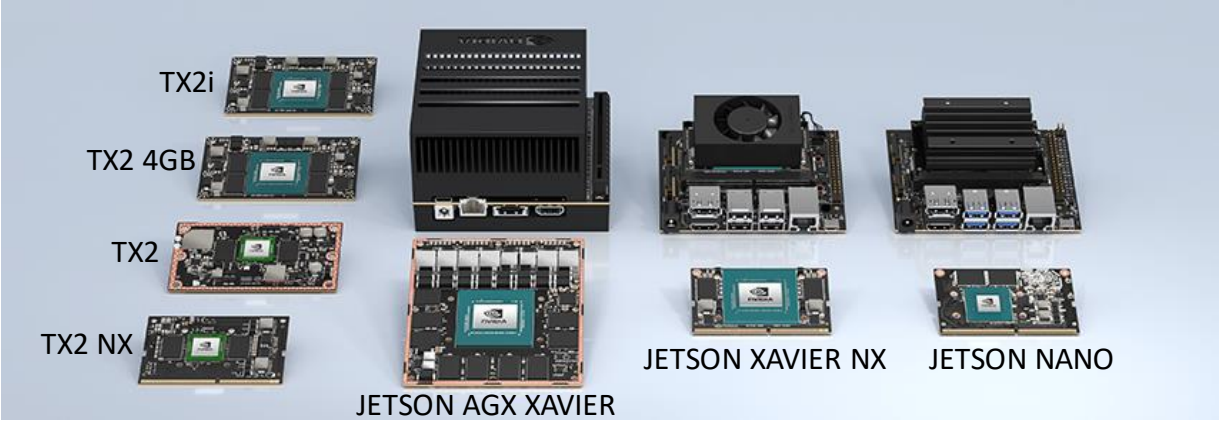
+



From Training to Deploying

TRAIN	OPTIMIZE	DEPLOY
	TENSOR-RT	
Train your model with the GPU NVIDIA A100 TENSOR CORE PCIE board	Dramatically speed up and reduce memory usage for your model on Jetson module	Deploy your model to your fleet of jetson module

JETSON MODULES



Module Name	Mechanical	AI performance	GPU	CPU (core)	Memory LPDDR4(x)	PCie	POWER	PRICE
Jetson Nano	69.6 * 45 (mm) 260-pin So-DIMM	472 GFLOPs	128-cores	4	4 GB 64-bits (25.6GB/s)	1 x4 (PCIe Gen2)	5W/10W	112 €
Jetson TX2 Series (Nx)	69.6 * 45 (mm) 260-pin So-DIMM	1.33 TFLOPs	256-cores	2 + 4	4 GB 128-bits (51.2GB/s)	1x1 + 1x2 (PCIe Gen2)	7.5W | 15W	172 €
Jetson Xavier NX	69.6 * 45 (mm) 260-pin So-DIMM	21 TOPs	384-cores with 48 Tensor Cores	6	8 GB 128-bits (59.7GB/s)	1x1 (PCIe Gen3) + 1x4 (PCIe Gen4)	10W | 15W | 20W	414 €
Jetson AGX Xavier	100 * 87 (mm) 699-pin connector	32 TOPS	512-cores with 64 Tensor Cores	8	32 GB 256-bits (136.5GB/s)	1 x8 + 1 x4 + 1 x2 + 2 x1 (PCIe Gen4)	10W | 15W | 30W	864 €

AGENDA

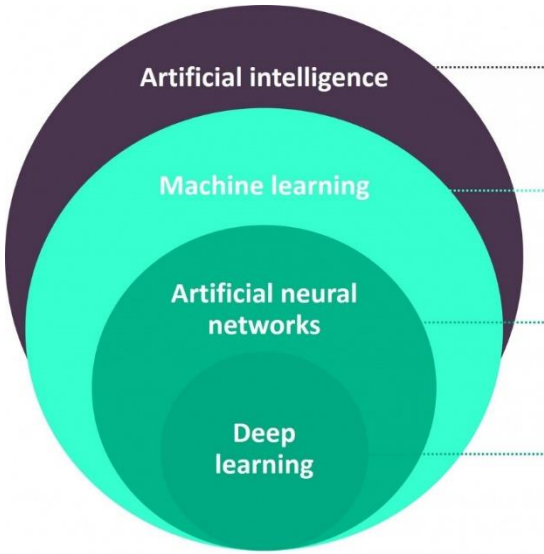
FASTER

1. FASTERV2 → Real time acquisition system
2. FASTERV3 → Real Time acquisition platform
3. NVIDIA Jetson modules

THINK

1. AI vs ML vs ANN vs DL
2. THINK challenges
3. THINK challenges benchmark
4. From Pytorch to TensorRT with ONNX
5. First results

AI vs ML vs ANN vs DL

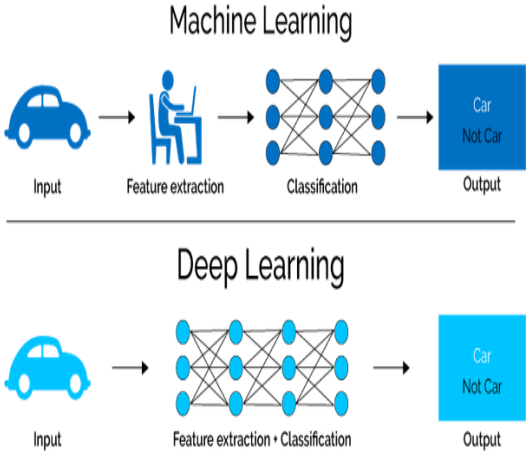


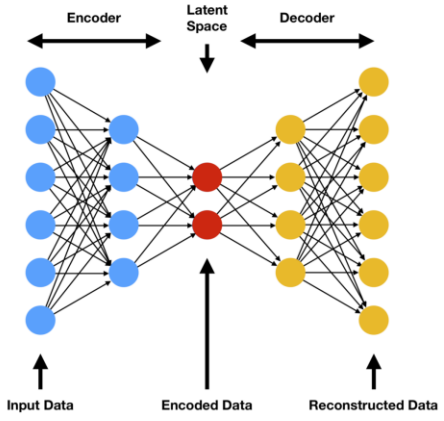
Artificial intelligence (AI)
Any techniques that enable machines to solve a task in a way like humans do

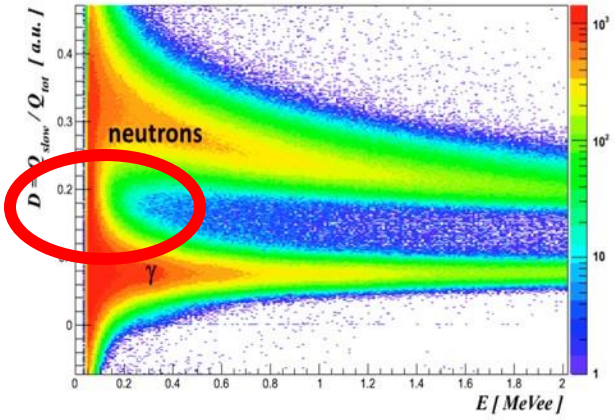
Machine learning (ML)
Algorithms that allow computers to learn from examples without being explicitly programmed

Artificial neural networks (ANN)
Brain-inspired machine learning models

Deep learning (DL)
A subset of ML which uses deep artificial neural networks as models and automatically builds a hierarchy of data representations

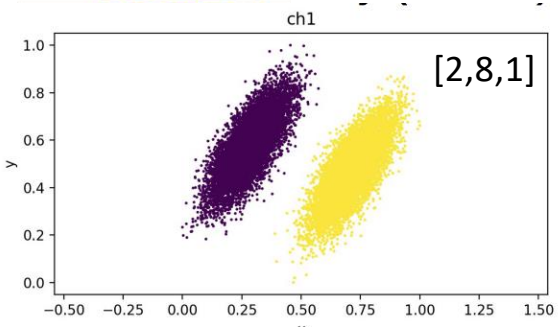


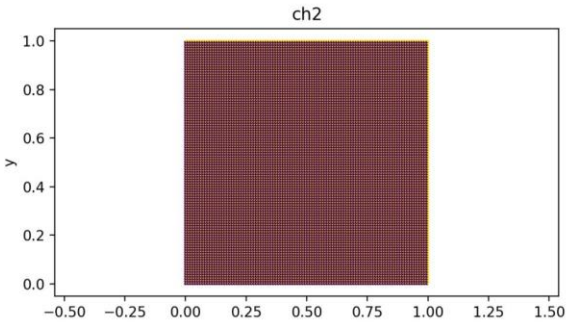




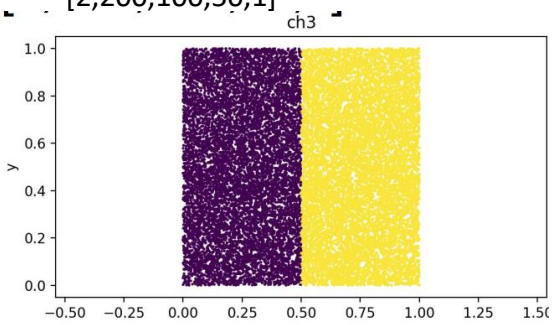
Frédéric Magniette
LLR - Think Project

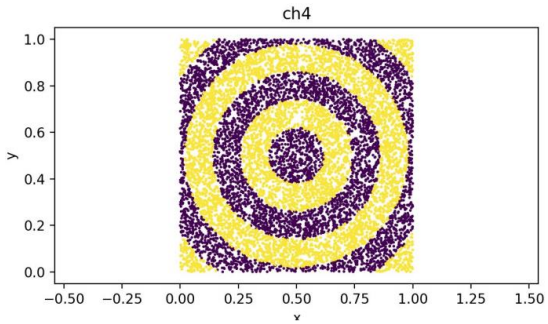
THINK Challenges



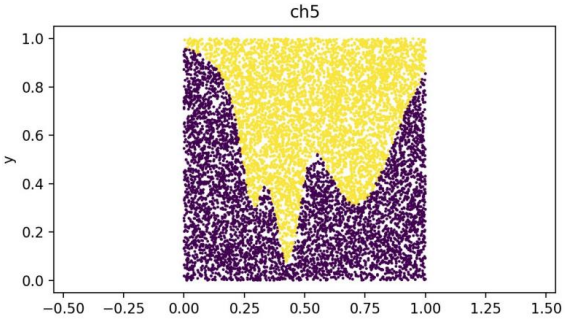


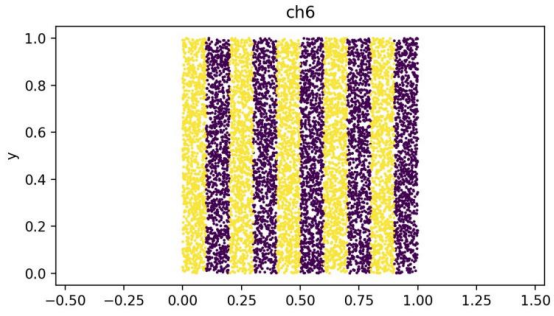
[2,200,100,50,1]

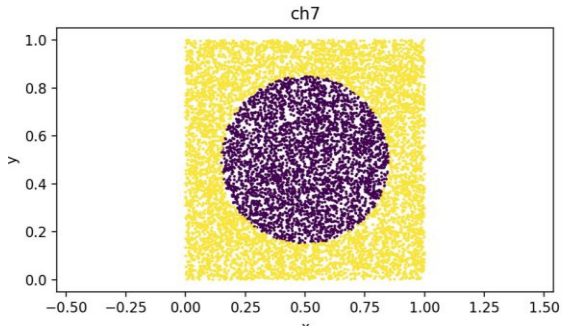


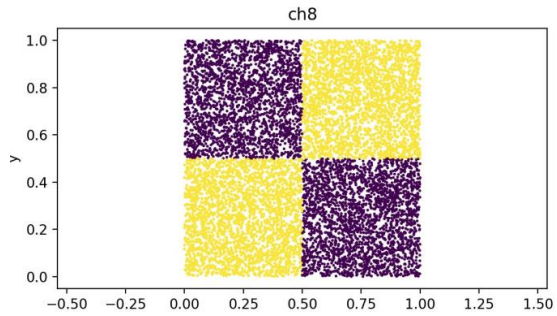


- 8 challenges are finalized
- 2 continuous inputs X and Y between 0 and 1 (points in a unitary square)
- 1 binary output (coloring) associated to the points
- 20000 points for each challenge
- A perceptron reference implementation for all of them (except ch2)









[2,500,250,100,50,1]

THINK Challenge benchmark

PyTorch
(Training mode)

ONNX

PyTorch
(Inference mode)

ONNX RUNTIME

TensorRT

JETSON NANO

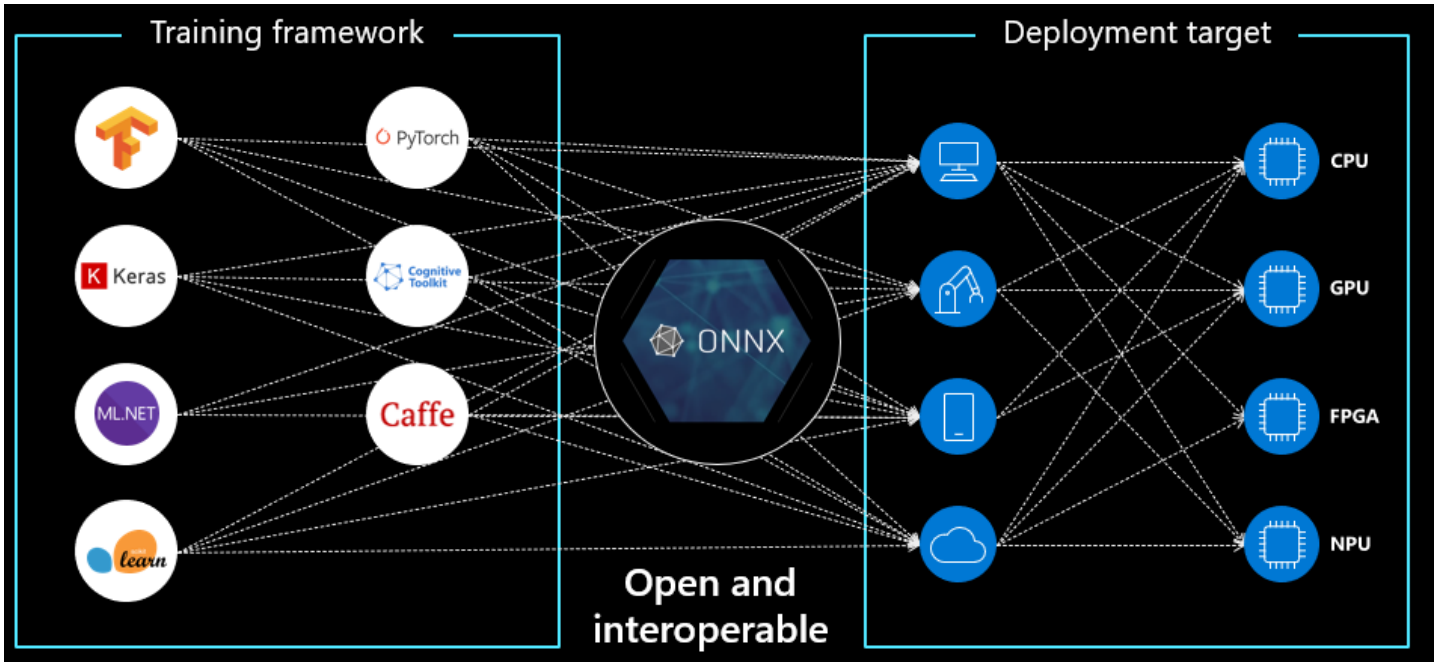
JETSON XAVIER NX

JETSON AGX XAVIER

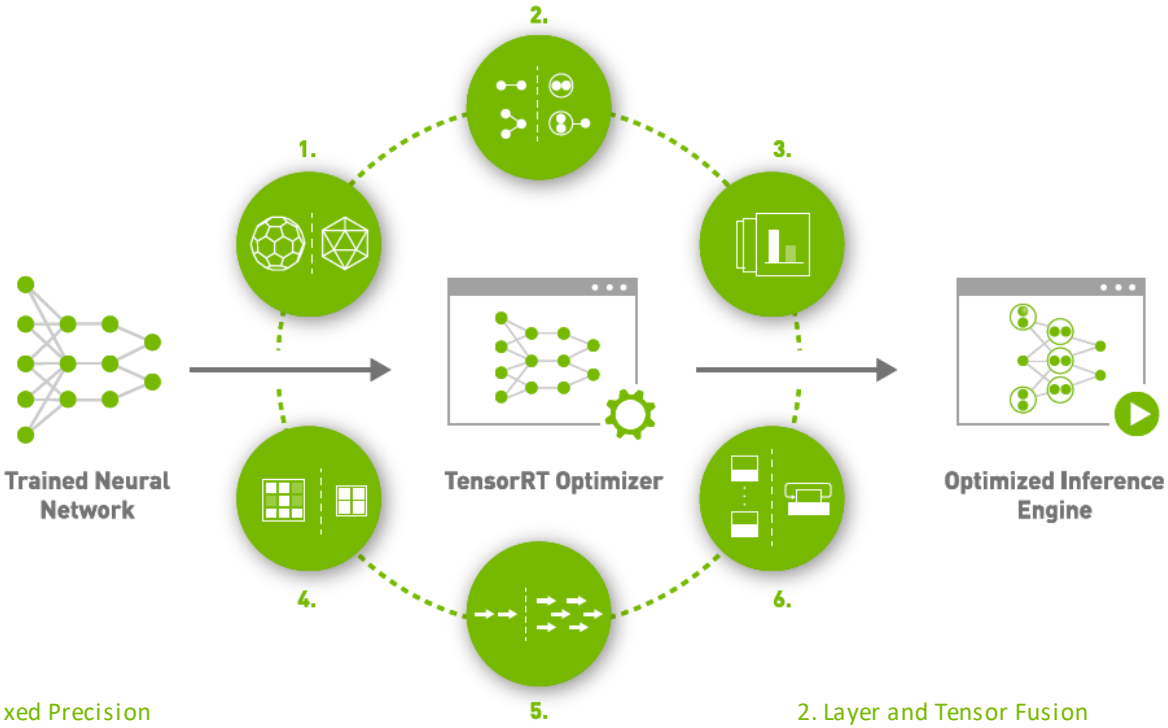
ONNX

Open Neural Network eXchange

ONNX is an open format built to represent machine learning models. ONNX defines a common set of operators - the building blocks of machine learning and deep learning models - and a common file format to enable AI developers to use models with a variety of frameworks, tools, runtimes, and compilers.



TensorRT



1. Reduce Mixed Precision
Maximizes throughput by quantizing models to INT8 while preserving accuracy

2. Layer and Tensor Fusion
Optimizes use of GPU memory and bandwidth by fusing nodes in a kernel

3. Kernel Auto-Tuning
Selects best data layers and algorithms based on the target GPU platform

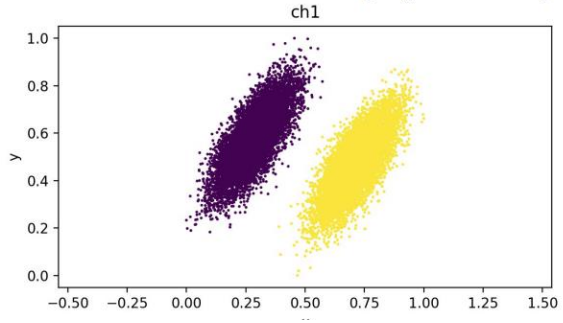
4. Dynamic Tensor Memory
Minimizes memory footprint and reuses memory for tensors efficiently

5. Multi-Stream Execution
Uses a scalable design to process multiple input streams in parallel

6. Time Fusion
Optimizes recurrent neural networks over time steps with dynamically generated kernels

First results

Challenge 1



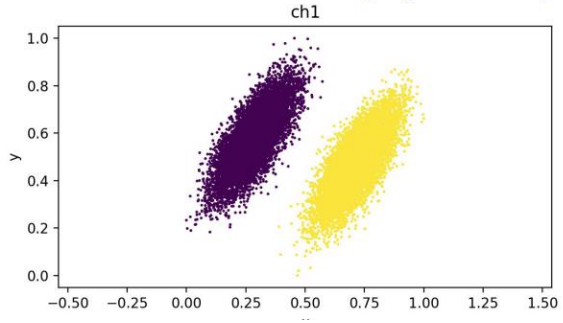
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,8,1]

Error = 0 %	Intel Core I7 (2.8Ghz 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 1



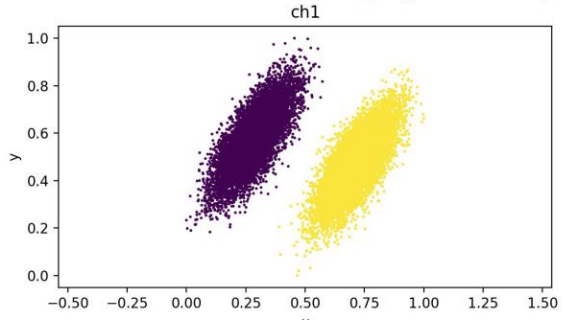
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,8,1]

Error = 0 %	Intel Core I7 (2.8Ghz 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~60ms	~650 ms							
Batch_size = 100	~1 ms	~13 ms							
Batch_size = 1000	~0.3 ms	~0.5 ms							
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 1



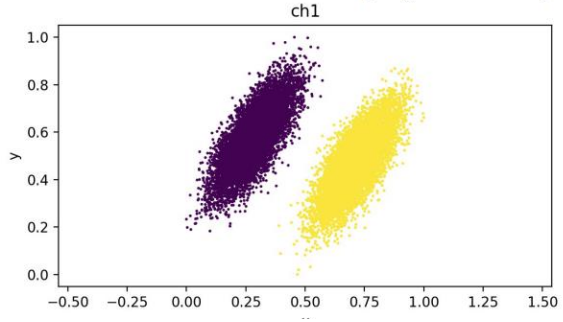
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,8,1]

Error = 0 %	Intel Core I7 (2.8Ghz 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~60ms	~650 ms		~500 ms	~170 ms				
Batch_size = 100	~1 ms	~13 ms		~10 ms	~2.5ms				
Batch_size = 1000	~0.3 ms	~0.5 ms		~3 ms	~0.8 ms				
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 1



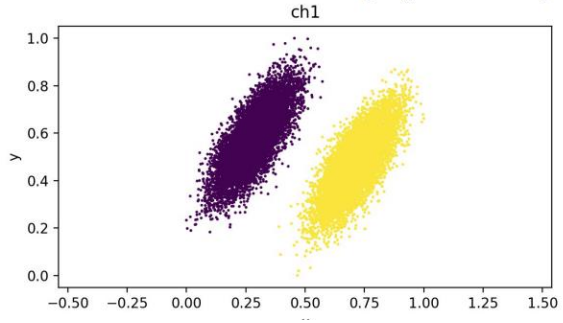
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,8,1]

Error = 0 %	Intel Core I7 (2.8Ghz 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~60ms	~650 ms		~500 ms	~170 ms		~500 ms	~110ms	
Batch_size = 100	~1 ms	~13 ms		~10 ms	~2.5ms		~50 ms	~1.5ms	
Batch_size = 1000	~0.3 ms	~0.5 ms		~3 ms	~0.8 ms		~25 ms	~0.5ms	
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 1



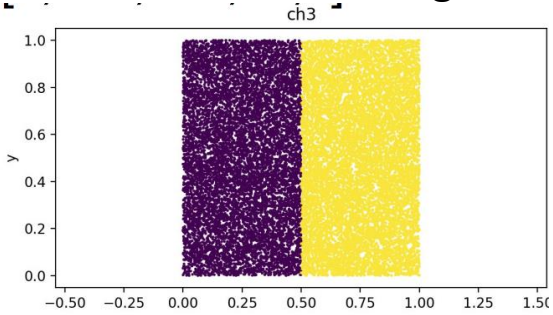
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,8,1]

Error = 0 %	Intel Core I7 (2.8Ghz 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~60ms	~650 ms		~500 ms	~170 ms		~500 ms	~110ms	
Batch_size = 100	~1 ms	~13 ms		~10 ms	~2.5ms		~50 ms	~1.5ms	
Batch_size = 1000	~0.3 ms	~0.5 ms		~3 ms	~0.8 ms		~25 ms	~0.5ms	
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1				~1800 ms					
Batch_size = 100				~800 ms					
Batch_size = 1000				~800 ms					

First results

Challenge 3



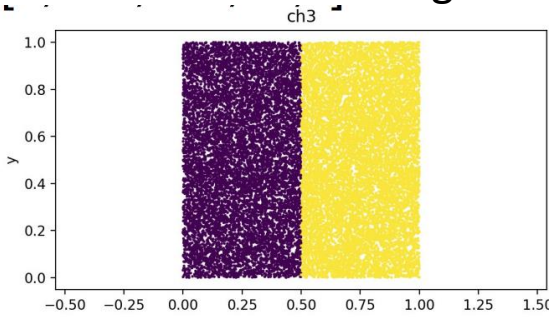
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,200,100,50,1]

Error < 0.2 %	Intel Core I7 (2.8G 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 3



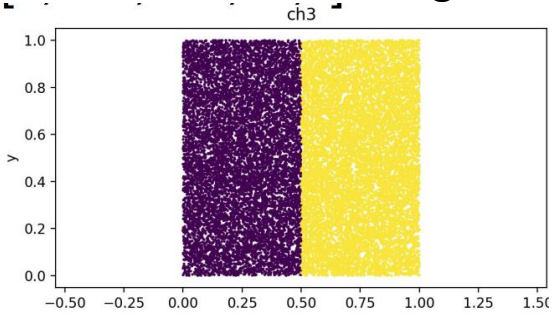
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,200,100,50,1]

Error < 0.2 %	Intel Core I7 (2.8G 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~70 ms	~700 ms							
Batch_size = 100	~1ms	~15ms							
Batch_size = 1000	~0.4 ms	~8 ms							
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 3



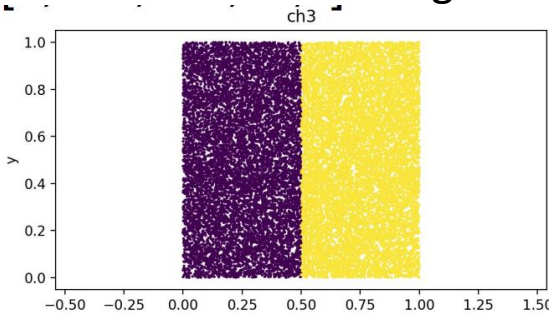
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,200,100,50,1]

Error < 0.2 %	Intel Core I7 (2.8G 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~70 ms	~700 ms		~1200 ms	~240 ms				
Batch_size = 100	~1ms	~15ms		~50 ms	~10ms				
Batch_size = 1000	~0.4 ms	~8 ms		~35 ms	~8 ms				
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 3



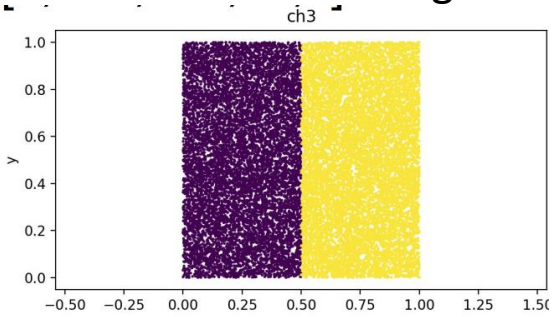
- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,200,100,50,1]

Error < 0.2 %	Intel Core I7 (2.8G 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~70 ms	~700 ms		~1200 ms	~240 ms		~500 ms	~150ms	
Batch_size = 100	~1ms	~15ms		~50 ms	~10ms		~50 ms	~8ms	
Batch_size = 1000	~0.4 ms	~8 ms		~35 ms	~8 ms		~25 ms	~6ms	
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1									
Batch_size = 100									
Batch_size = 1000									

First results

Challenge 3



- 20000 points for each challenge :
  - 18000 for the training
  - 2000 for testing
- A perceptron reference implementation

[2,200,100,50,1]

Error < 0.2 %	Intel Core I7 (2.8G 4 Cores)			JETSON XAVIER NX			JETSON AGX XAVIER		
CPU	Pytorch	ONNX runtime	TensortRT	Pytorch	ONNX Runtime	Tensort RT	Pytorch	ONNX Runtime	TensortRT
Batch_size = 1	~70 ms	~700 ms		~1200 ms	~240 ms		~500 ms	~150ms	
Batch_size = 100	~1ms	~15ms		~50 ms	~10ms		~50 ms	~8ms	
Batch_size = 1000	~0.4 ms	~8 ms		~35 ms	~8 ms		~25 ms	~6ms	
GPU	Pytorch	ONNX RT	TensortRT	Pytorch	ONNX RT	Tensort RT	Pytorch	ONNX RT	TensortRT
Batch_size = 1				~2300 ms					
Batch_size = 100				~830 ms					
Batch_size = 1000				~800 ms					

Next Step

- Running a ONNX model on GPU (onnxruntime_gpu)
- Running a ONNX model with TensorRT
- FP32 -> FP16 -> INT8
- Include Memory copy latency (CPU to GPU)
- …