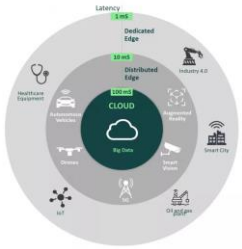
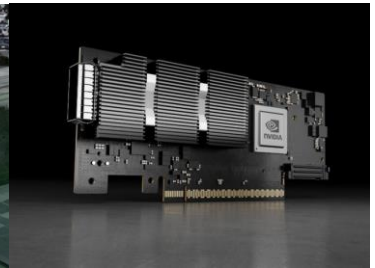
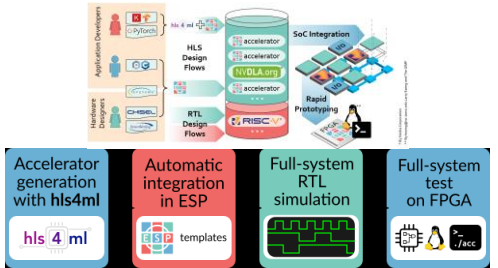
L'intelligence Artificielle dans les systèmes embarqués
– Aspects Matériels –

➔Bilan Matériels

➔Mise en œuvre







SCIENCES EXPERIMENTALES | SCIENCES THEORIQUES | SCIENCES CALCULATOIRES | SCIENCES PILOTEES PAR LES DONNEES | SCIENCES PILOTEES PAR L'INTELLIGENCE ARTIFICIELLE

Approche déductive

Symbolisme

Données d'entrée

ALGORITHMES

Données de sortie

EXPERTS:
LOIS
Règles

Boucles
Tests logiques

Traitement de cas particuliers:
Logiciels spécialisés

Approche inférentielle (Inductive)

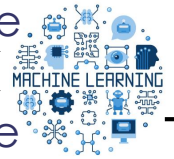
connectivisme

+ Probabilisme

+ Analogisme

Données d'entrée

Données de sortie

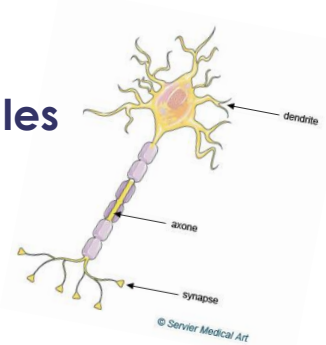


Modèle d'inférence

MESURES
FAITS
Informations

Algorithmes apprenants

Lois
Règles

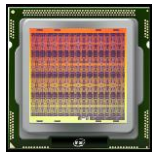


L'intelligence Artificielle dans les systèmes embarqués

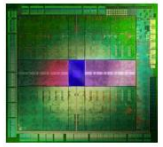
Nouvelles architectures matérielles

CPUs+ NNproc

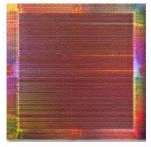
DSPs



NMC

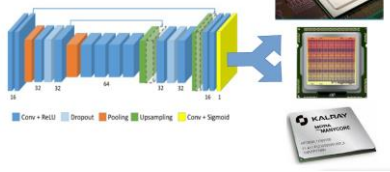


GPUs



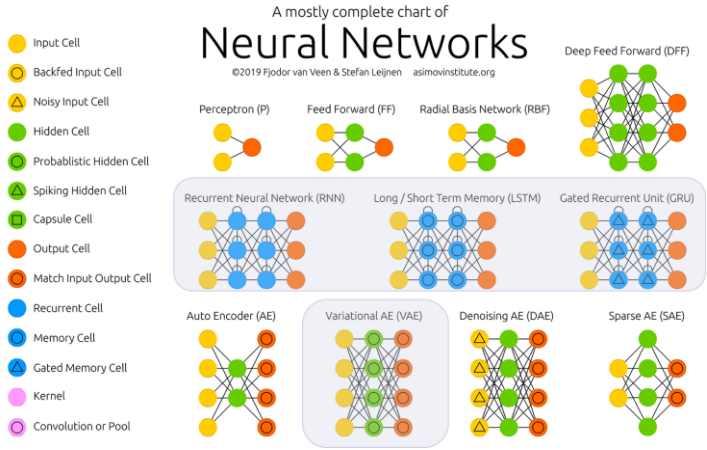
FPGAs

Frederic Druillole LP2I Bd›

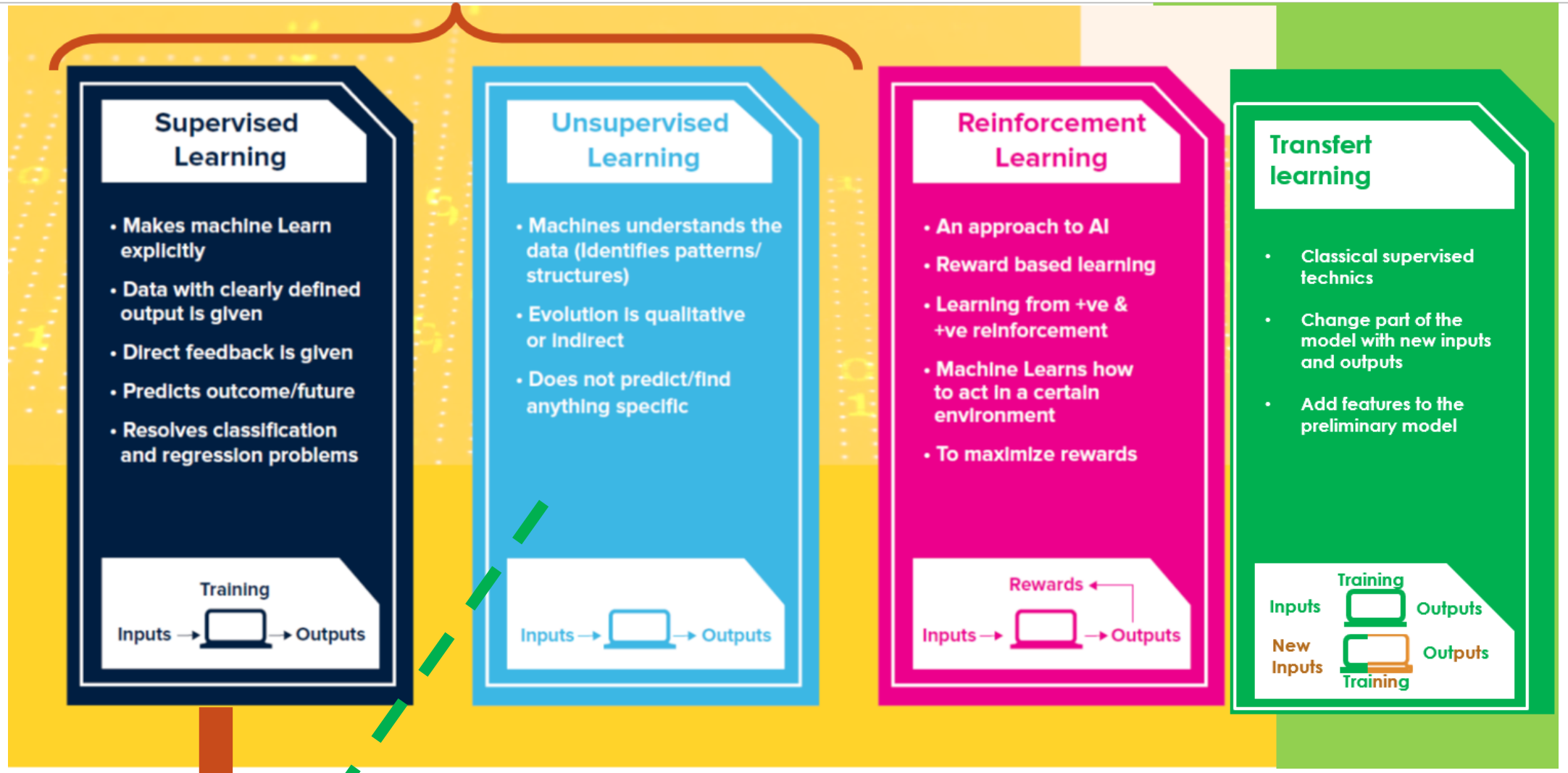


Outils de générations de données

Réseaux de neurones et descente de gradient



Les quatre catégories d'apprentissage profond



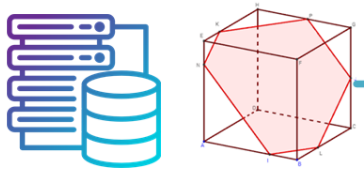
Système Embarqués

Algo Embarqué

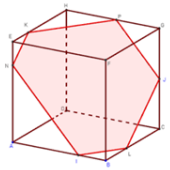
Méthodes

MLOps

TRAIN THE MODEL

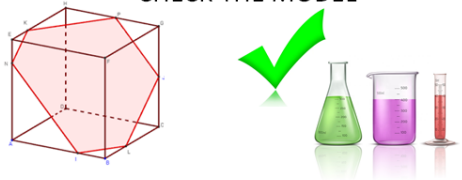


PACKAGE THE MODEL

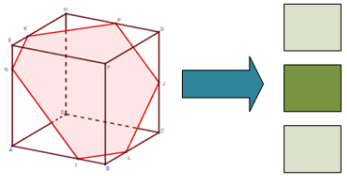


PACKAGE
Model
Framework

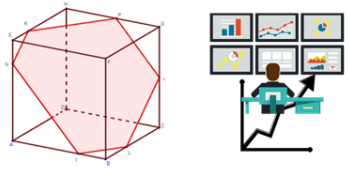
CHECK THE MODEL



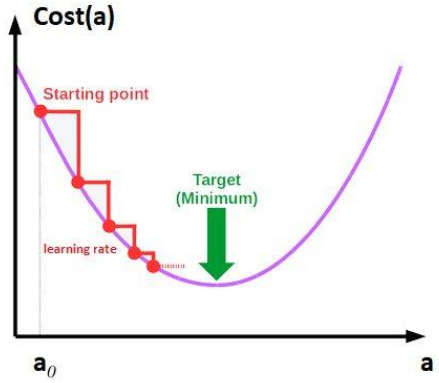
DEPLOY THE MODEL



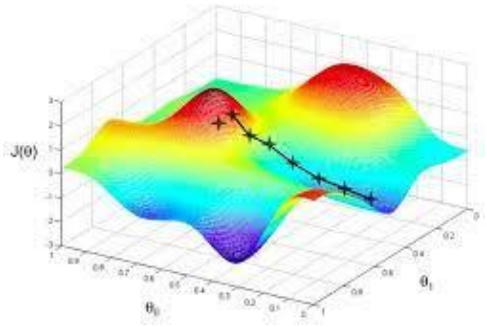
MONITOR THE MODEL

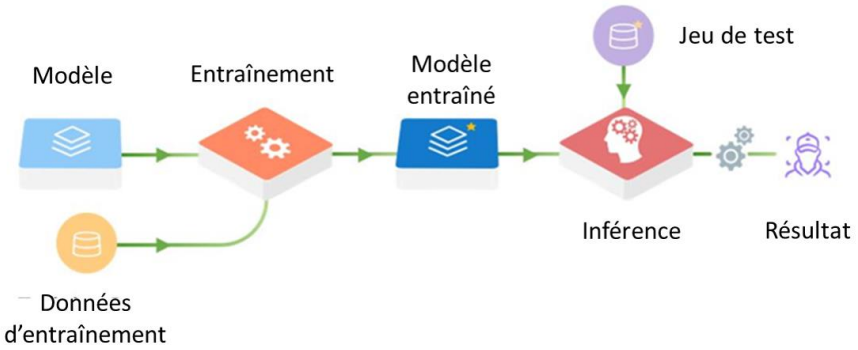


RETRAIN THE MODEL



Gradient Descent





IA cloud centralisé

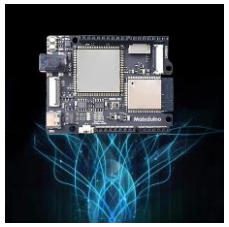
Grandes compagnies

- Gourmant en puissance electrique
- Gourmant en bande passante
- Gourmant en quantité de données

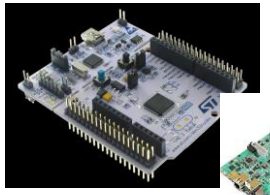
Migration

Réseau de composants (SE) pour de l'IA frontale

Kendryte K210



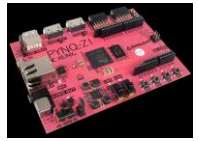
STM32 Cube AI



AMD-Xilinx

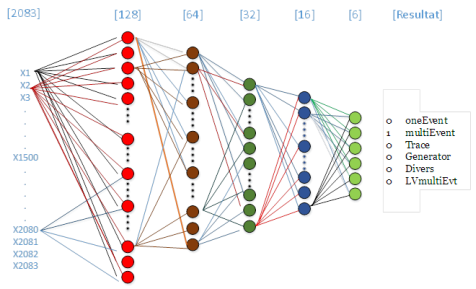
Intel

Digilent

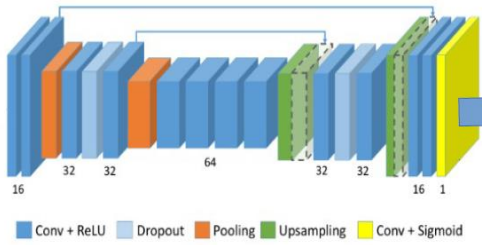


Les briques de bases

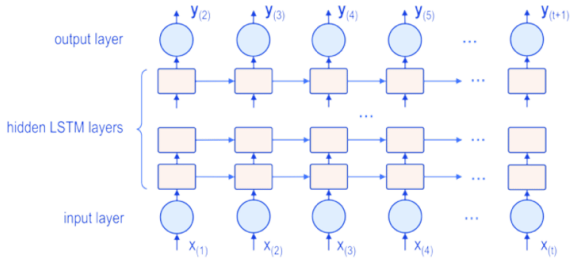
DEEP NEURAL NETWORK

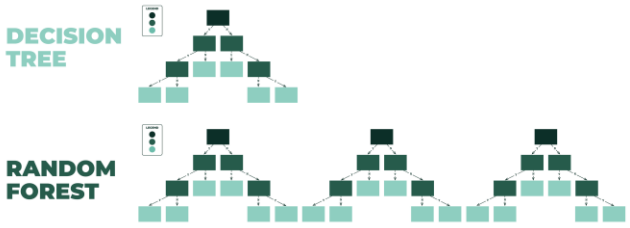


CONVOLUTIONAL NEURAL NETWORK



RECURRENT NEURAL NETWORK





Classique

- Séries temporelles (LLM, ChatGPT)
- Données Audio
- Générateur d'information en apprentissage profond
- Apprentissage par renforcement
- Traitement des données avec graphes (chaînes)
- Traitement des images et des vidéos (computer Vision)
- Traitement du langage naturel
- Structuration des données

Embarqué

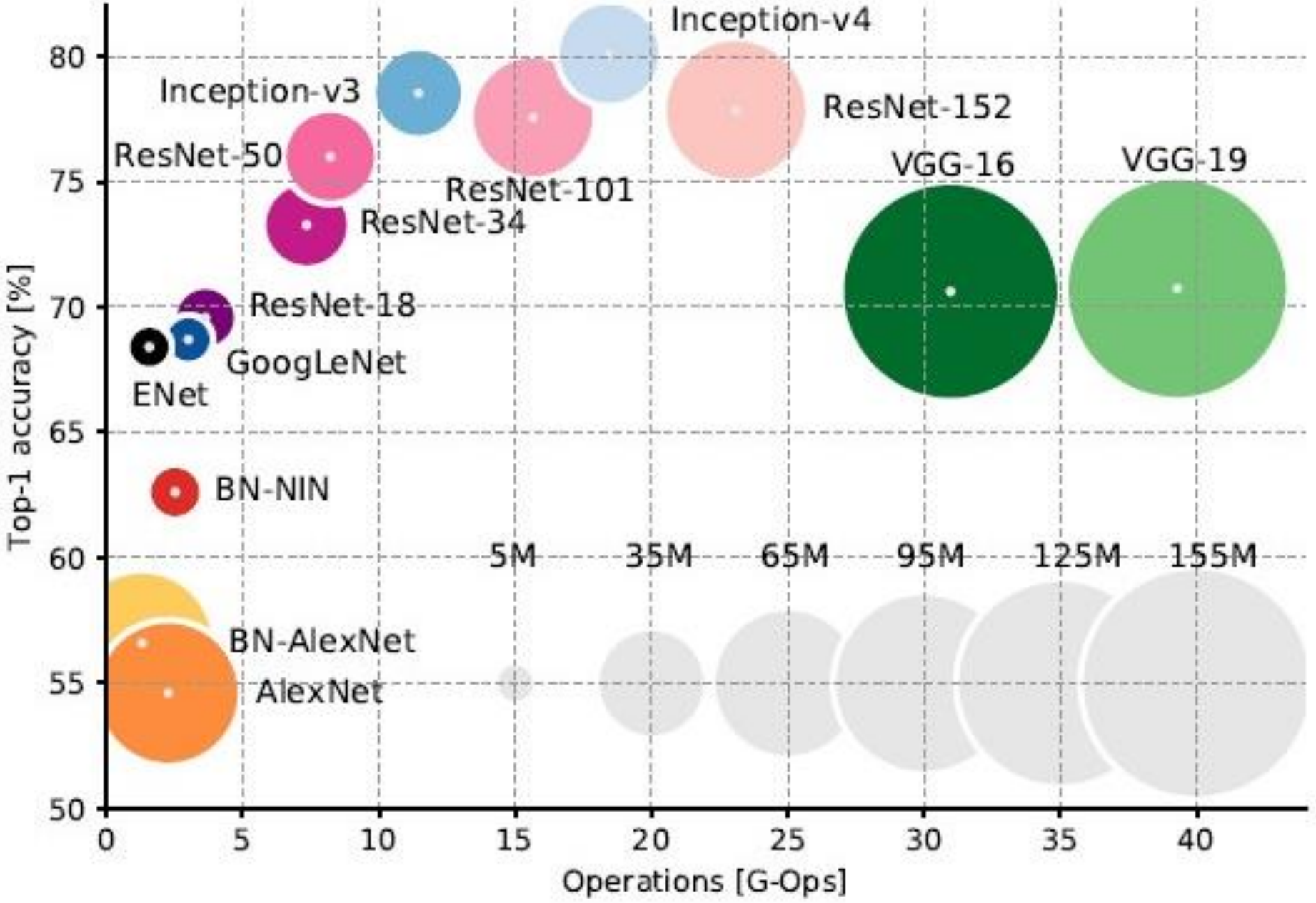
Off-line

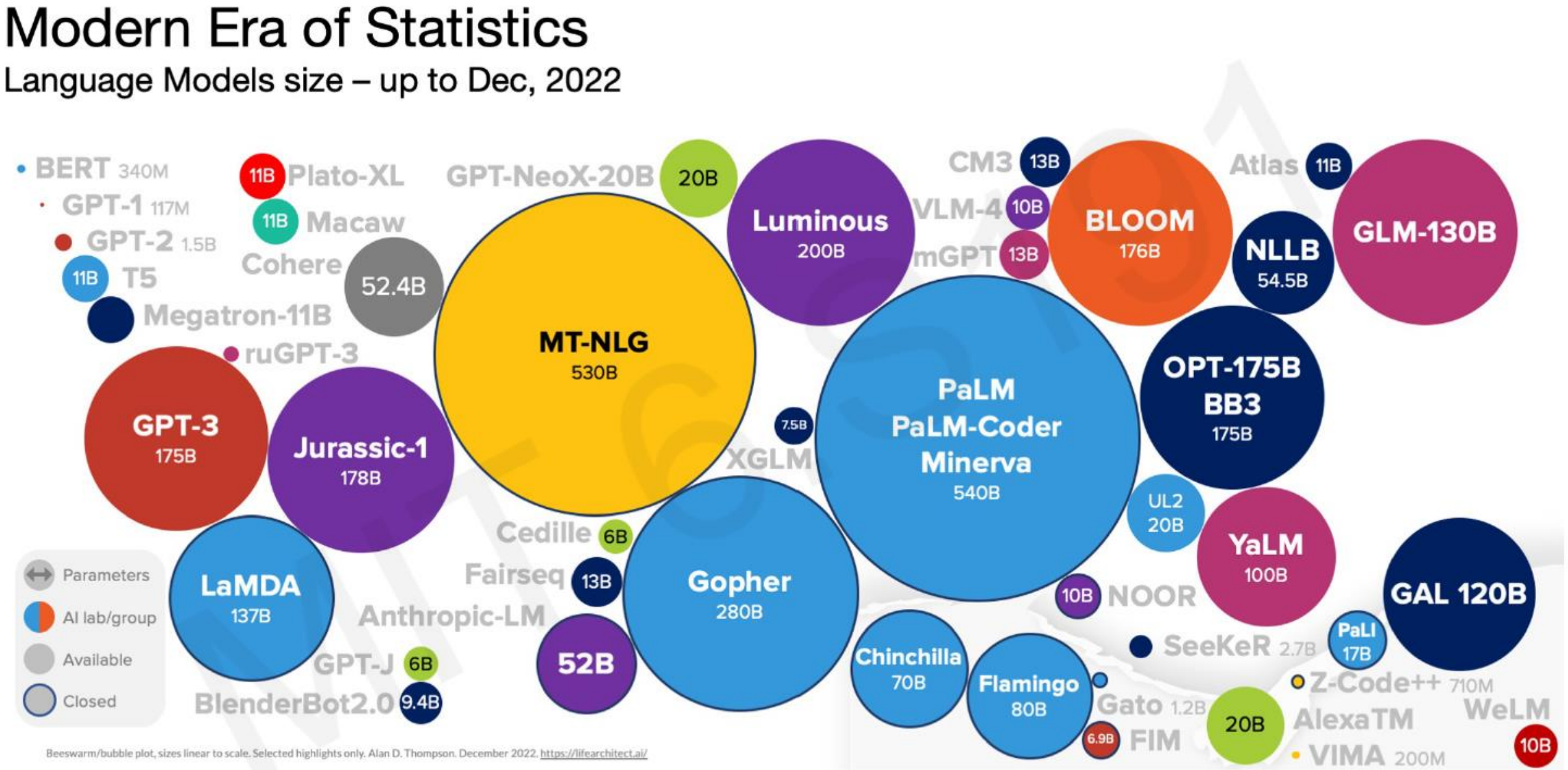
- Signal generation
- Design Optimisation

On-line

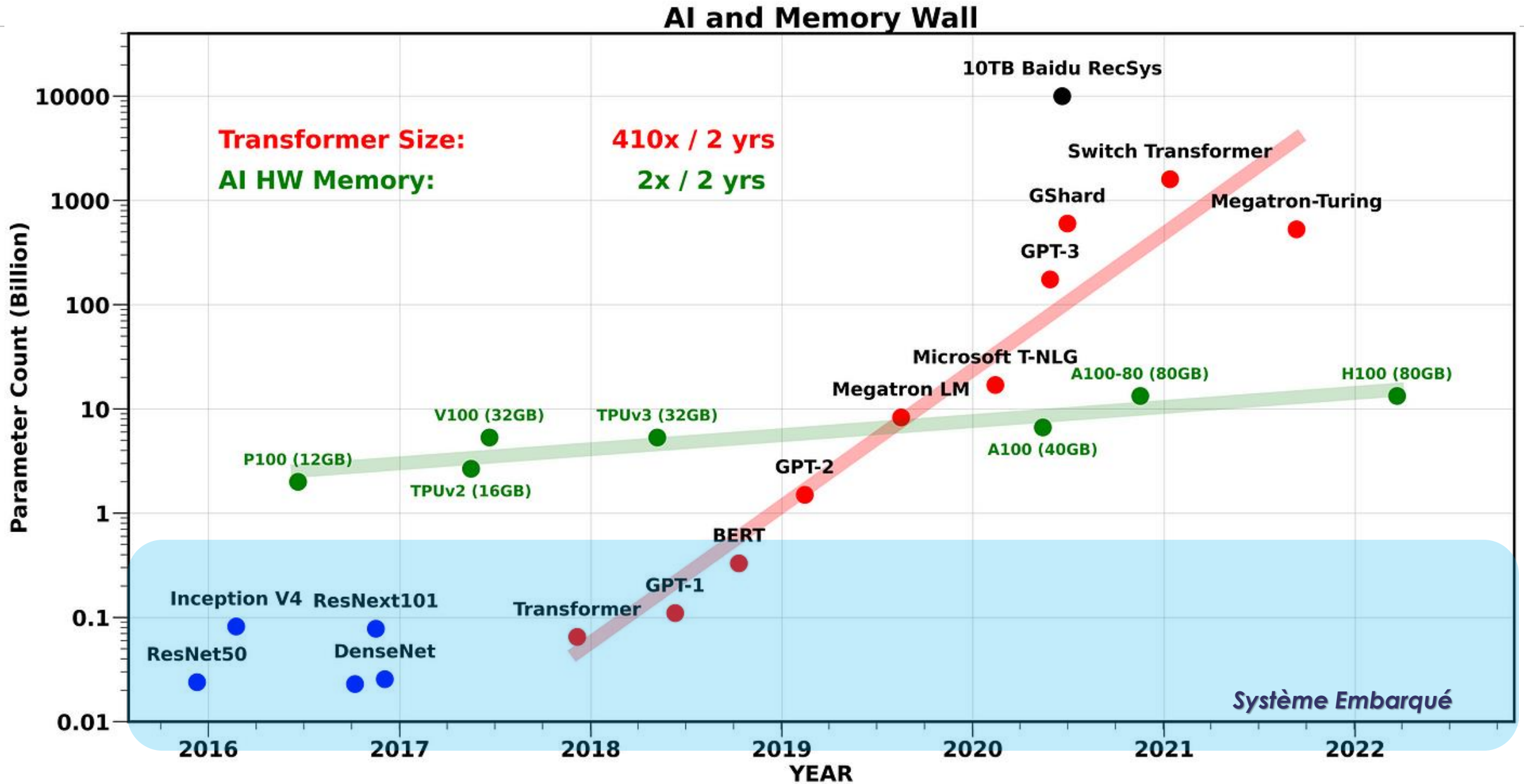
- Optimisation des Instruments
  - Signal recognition
  - Pile-Up recovery
  - Signal deconvolution
- Selection/ Classification/ Decision
  - Data selection
  - Data parameters prediction
  - Denoizing
- Data Compression
  - Reduced data format

Les autres réseaux : Performances (avant 2020)

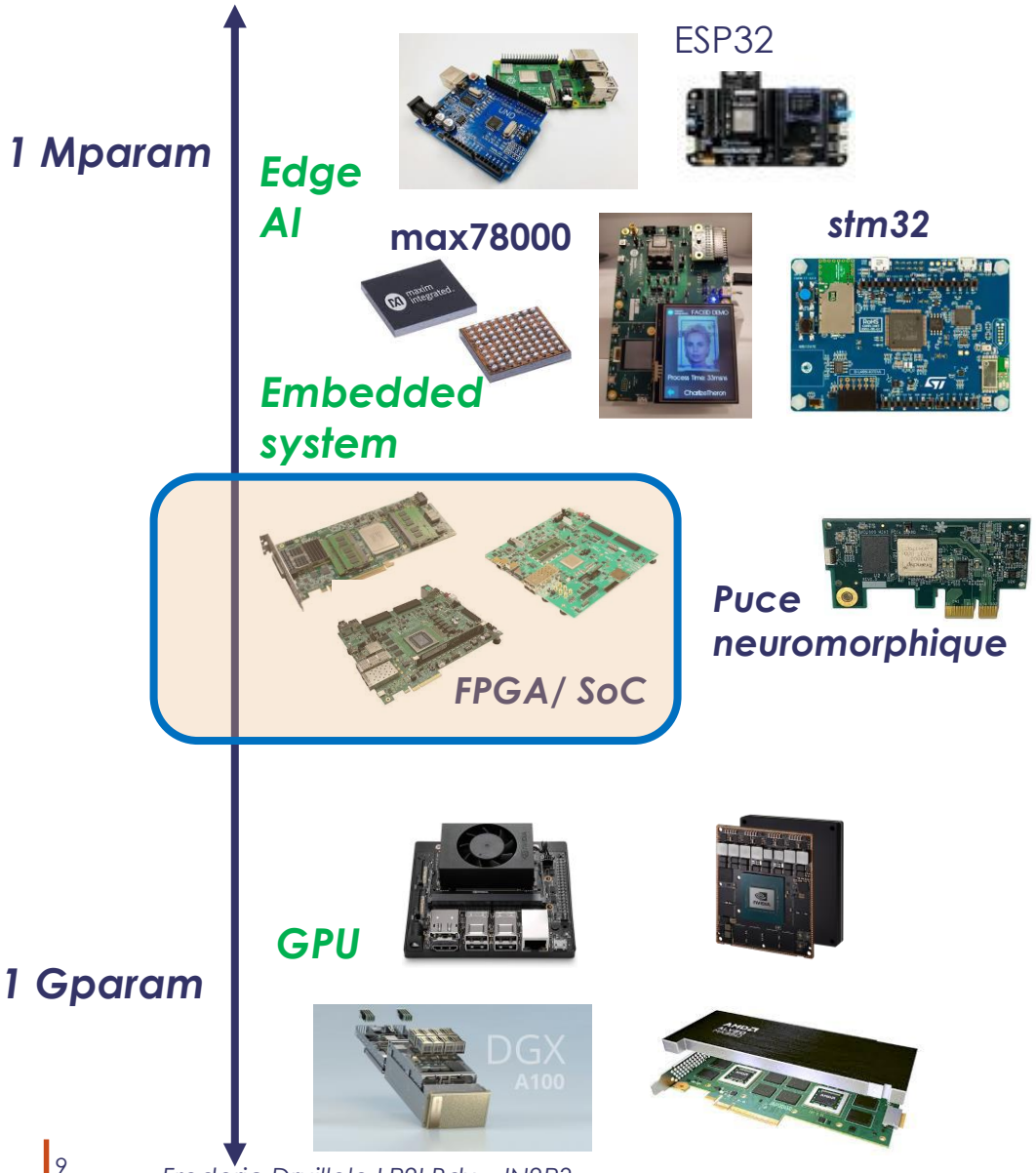



Modern Era of Statistics
Language Models size – up to Dec, 2022
BERT 340M
GPT-1 117M
GPT-2 1.5B
11B T5
Megatron-11B
11B Plato-XL
11B Macaw
Cohere
52.4B
ruGPT-3
GPT-3 175B
Jurassic-1 178B
LaMDA 137B
GPT-NeoX-20B 20B
MT-NLG 530B
Luminous 200B
CM3 13B
VLM-4 10B
mGPT 13B
BLOOM 176B
Atlas 11B
NLLB 54.5B
GLM-130B
7.5B XGLM
PaLM PaLM-Coder Minerva 540B
OPT-175B BB3 175B
UL2 20B
YaLM 100B
10B NOOR
SeeKeR 2.7B
PaLI 17B
GAL 120B
Z-Code++ 710M
AlexaTM 20B
WeLM 10B
VIMA 200M
Cedille 6B
Fairseq 13B
Anthropic-LM 52B
Gopher 280B
Chinchilla 70B
Flamingo 80B
Gato 1.2B
6.9B FIM
GPT-J 6B
BlenderBot2.0 9.4B
Parameters
AI lab/group
Available
Closed
Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. Alan D. Thompson. December 2022. https://lifearchitect.ai/

https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8



Choisir la plateforme et Optimiser le modèle



Entrées

RNN : modèle avec un énorme nombre de paramètres

Aspects matériels:

Créer un modèle synthétisable
Gestion des ressources

Objectifs

Transférer l'implémentation sur calculateur du réseau au niveau matériel

Optimiser le réseau pour trouver un compromis entre la performance (précision) et ressources matériels

Méthodes

Elagage (Pruning):
Reduire la taille du réseau entrainé

Trouver un réseau optimisé en taille en même temps que l'entrainement (quantification)

IA embarqué : 2 solutions

~~CPU~~ ⟷ Calcul implémenté dans une solution matérielle dédiée

Spatial Accelerators

Basé sur des fonctions matérielles dédiées aux calculs sans logiciel (calcul matriciel)

ASIC
Circuit neuromorphique
FPGA

2 En cours de développement

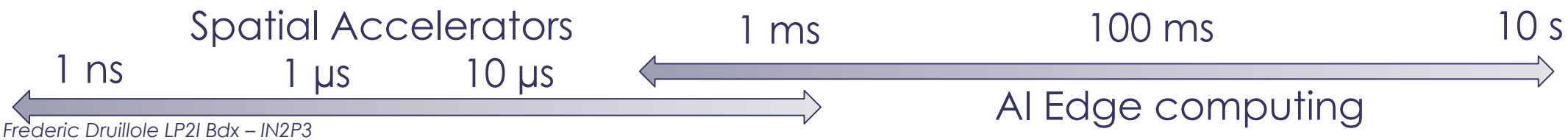
CPU ⟷ Architecture de calcul annexe

Edge computing

Reste basé sur de la programmation logicielle

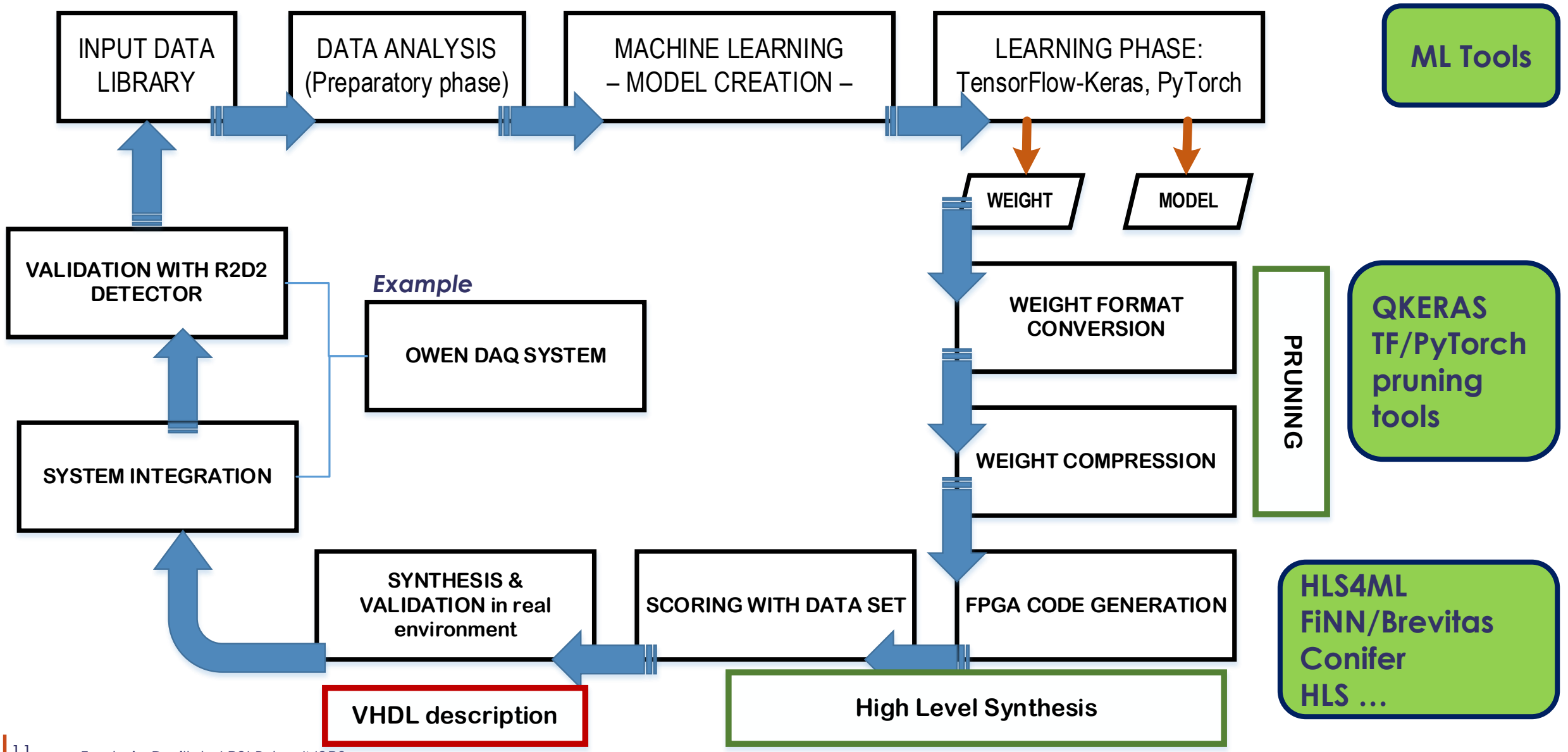
MMPA
GPU
FPGA
PU redesigned pour l'IA (TPU, KPU…)

1 Pilote les développements actuels



Methodology of design

➔ Outils différents suivant la cible

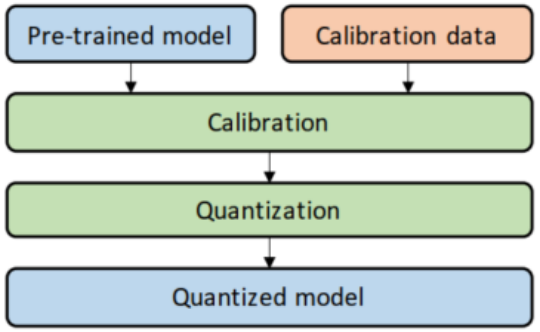


Optimisation de l'inférence pour l'embarqué

QUANTIZATION

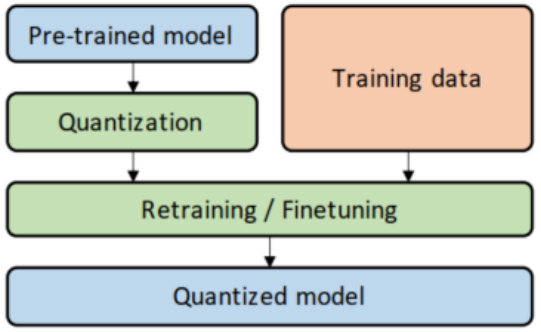
Floating Point → integer Fixed Point
32bits → 16,8,4,2 bits

PTQ: Post-Training Quantization



Conversion des données et du modèle après apprentissage.

QAT: Quantization Aware Training



Réentrainement du modèle en ayant convertit la précision des données et du modèle

FP32

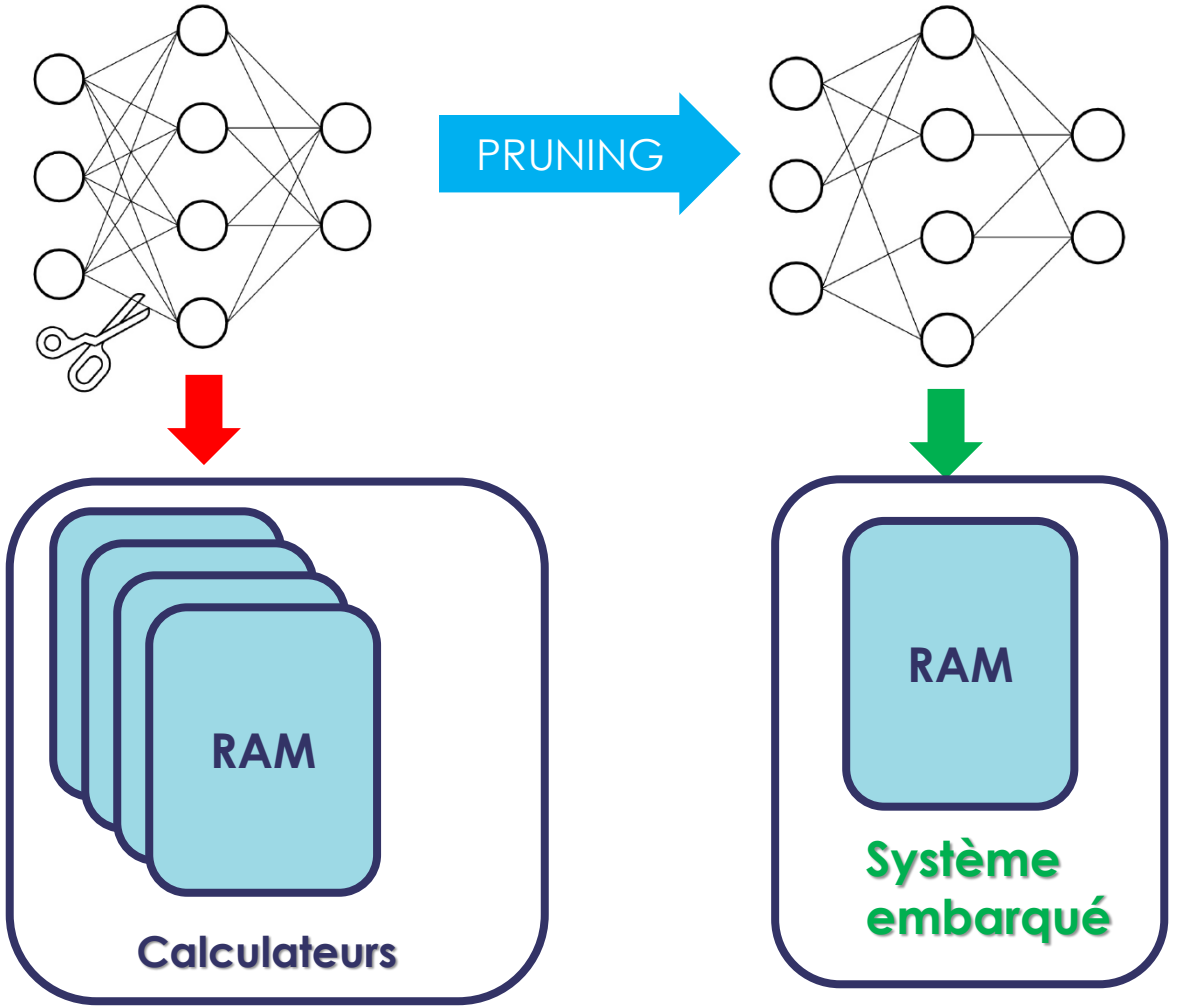
INT8 <3,5>

Perte de précision → Justesse du modèle

Travail sur la précision du modèle

PRUNING

On conserve les poids au dessus d'une certaine valeur :
Elagage



Les Frameworks d'inférence des modèle d'IA

AI Edge computing

- **Ax8i (max78000)**
- **ST Microelectrics**
  - CubeIDE + Cube-AI
- **VITIS AI (AMD)**
- **OpenVino (Intel)**
- **N2D2 – DeepGreen**
  Eclipse Aidge

  Solution Europeenne Open
- **Edge Impulse (société)**

Spatial Accelerators

- **VIVADO HLS**
  - HLS4ML
- FiNN (Xilinx R&D)
- ChaiDNN

  Consortium ?
- **HADDOC2**
- **DNNWeaver**
- **…**

Framework servant à réaliser l'accélération des calculs des modèles IA pour l'inférence.
Utilise des modèles éprouvés
Difficile de créer des inférences sur des composants à faibles ressources (multi-voies, proche des capteurs)

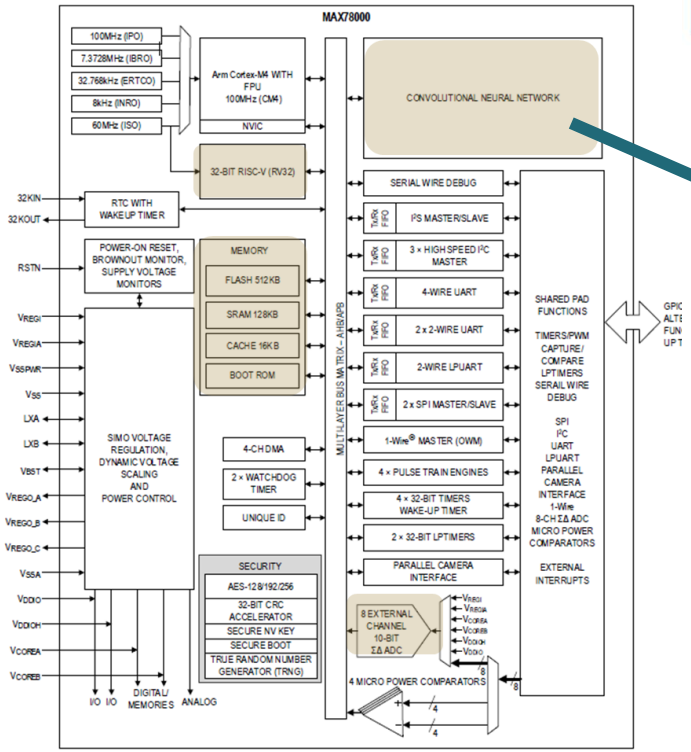
Création d'un framework dédié à Xilinx et Intel en VHDL

Critères de comparaison des matériels

Critères	CPU	CPU+AI-CoPro	GPU	FPGA	ASIC
Adaptabilité (à une variété de modèles)	**Elevé**	**Elevé**	Moyen	**Faible**	**Aucune**
Puissance de calcul	**Faible**	Moyen-Elevé	**Elevé**	**Elevé**	Moyen
Latence	Moyen (ms)	Moyen (ms)	**Faible** (µs)	**Faible**(10 ns)	**Très faible**
Flux d'entrée	**Faible**	Moyen	**Elevé**	**Elevé**	**Elevé**
Parrallelisme	**Faible**	Moyen	**Elevé**	**Elevé**	**Elevé**
Efficacité de la consommation électrique	Moyen	Moyen	Moyen	Moyen	**Elevé**
Mise en Oeuvre	**Facile**	Moyen	Moyen	**Complexe**	**Très Complexe**

Savoir-Faire

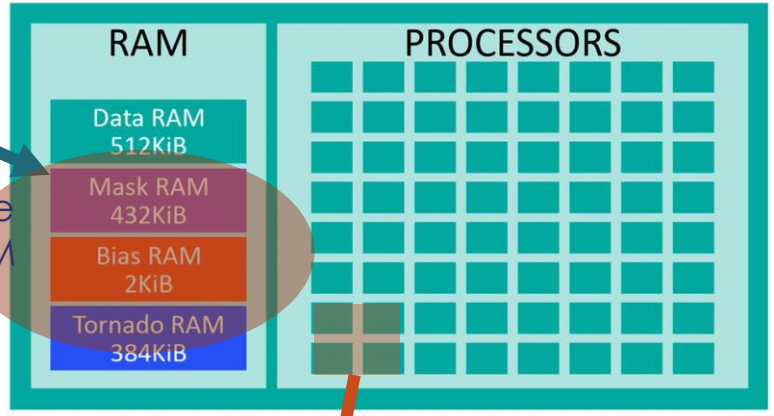
Architecture Max78000



ANALOG DEVICES

MAX78000 CNN Accelerator

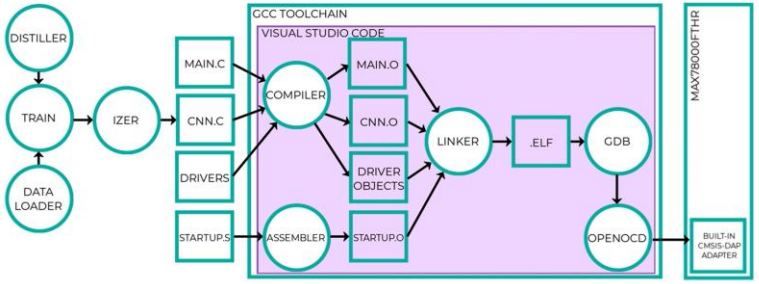
Modèle en RAM



Génération du code C++ cnn.h et main.c
➔ Chargement en mémoire du modèle, machine d'état pour réaliser l'inférence

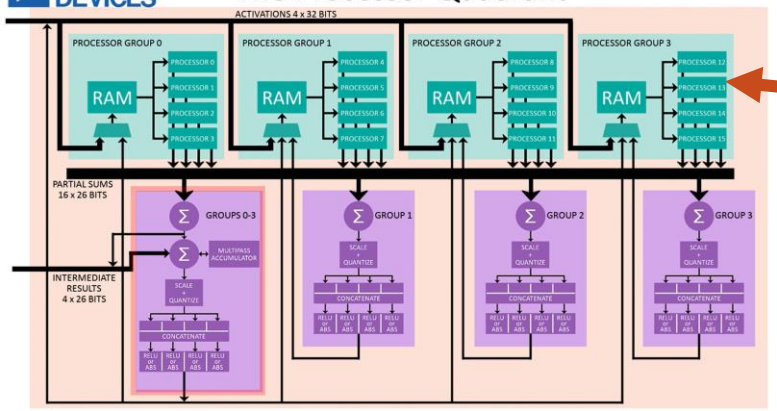
ANALOG DEVICES

A Choice of Wrappers

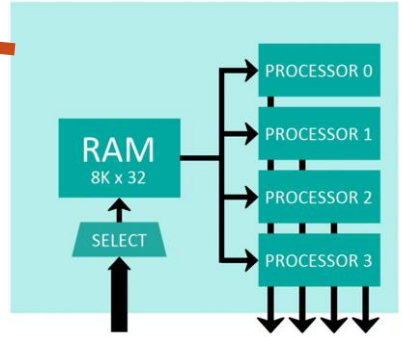


ANALOG DEVICES

The Processor Quadrant

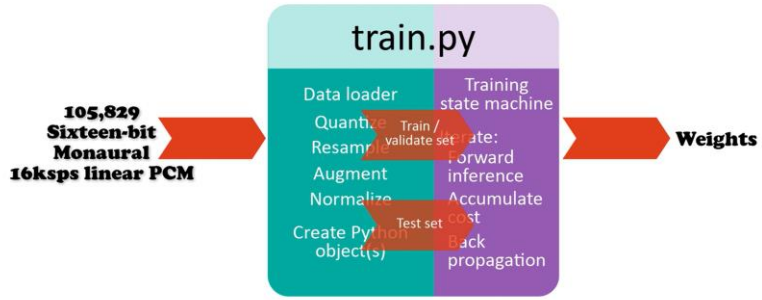


The Processor Group



ANALOG DEVICES

Training KWS20v3



Des exemples matériels et applicatifs: MAX78000

Données Y et X **Modèle PyTorch** **Apprentissage classique** → *Architecture* → *Poids du réseau*

On réecrit le modèle avec la bibliothéques ai8x

Outil sous Linux

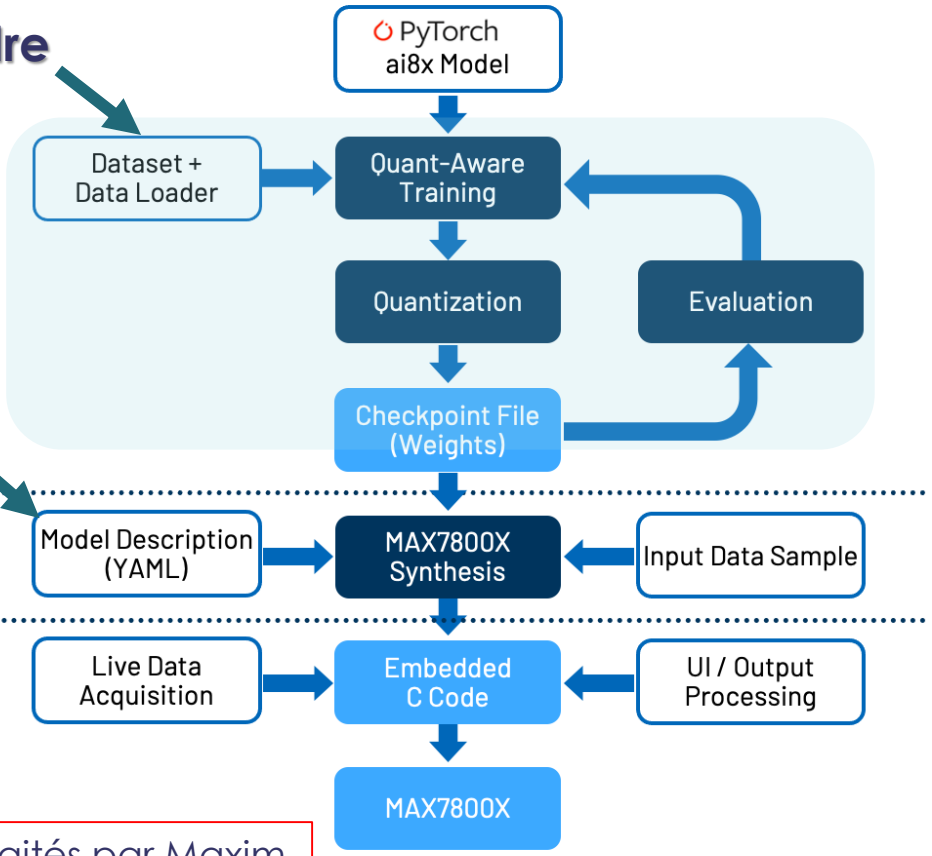
Fonctions à apprendre

① Training
ai8x-training

Syntaxe à apprendre

② Synthesis
ai8x-synthesis

③ Deployment
Embedded SDK



Difficile d'adapter les outils à des modèles non traités par Maxim
Reconstruire les scripts Python
Retro-engineering à réaliser
Validation de la chaine à refaire

hls4ml pipeline

Supported DL frameworks:

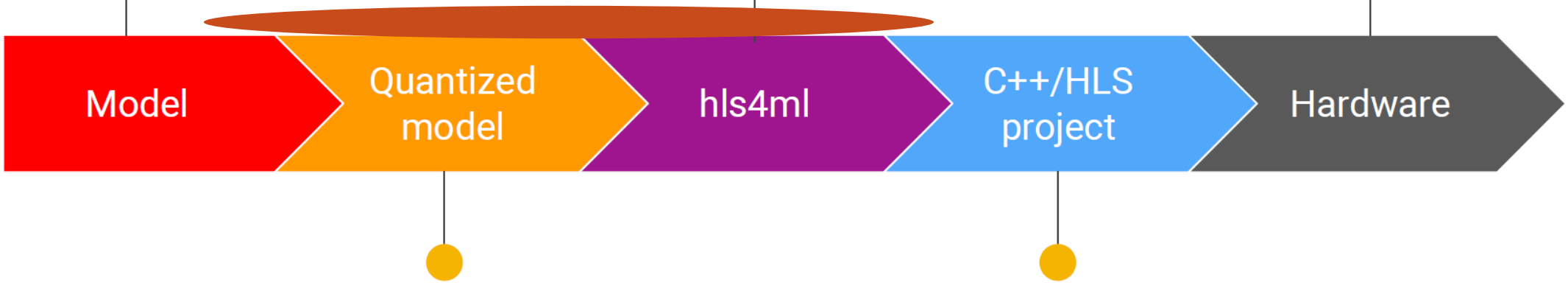
Keras
PyTorch
ONNX

Model conversion, optimization, profiling & tuning

hls4ml

Xilinx FPGAs, Intel/Altera FPGAs, Intel x86 CPUs

XILINX
intel



Quantization and pruning techniques:

- QKeras + AutoQ (Keras)
- Brevitas (PyTorch)

VIVADO
Mentor A Siemens Business
oneAPI
Quartus Prime

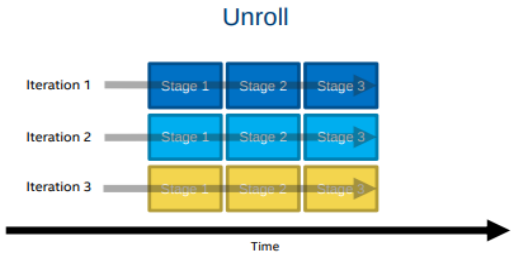
- **Zynq + HLS4ML**: io_parallel / io_stream /Reusefactor / Resource / Latency

	ARTY-Z7 CH1	ARTY Z7 CH3	ARTY Z7 CH3 Optimisé	ZCU102 CH3	CH4	ZCU102 CH4 Qbit<16,2>	ZCU102 CH5	ZCU102 CH7 Optim Ressource	ZCU102 CH7 Optim Latence	ARTY Z7 CH7 Optim HLS Stream
Nombre de cellules	**16**	**25801**	**25801**	**25801**	**658951**	**658951**	**156951**	**156951**	**156951**	**156951**
Horloge de reference	4,166ns	9,408ns	9,408ns	4,396ns		4,369ns	4,369ns	4,369ns	4,028ns	9,410ns
Temps de latence	70ns	19,804us	19,804us	2,510us		5,510us	5,510us	10,010us	10,015us	141us
BRAM	0%	41%	8%	6%		36%	18%	9%	15%	72%
DSP48E	21%	115%	11%	10%		59%	59%	12%	24%	6%
FF	2%	85%	28%	10%		28%	22%	32%	53%	167%
LUT	1%	280%	68%	46%		103%	113%	81%	104%	423%
URAM	0%	0%	0%	0%		0%	0%	0%	0%	0%

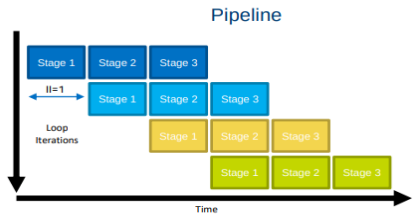
ARTY Z7
ZCU104

Tensorflow/Keras
+ HLS4ML
VITIS HLS

depending on VITIS-HLS #pragma
The way HLS handles vector/matrix before DSP



Question of HLS optimisation



What if directly written in VHDL ?

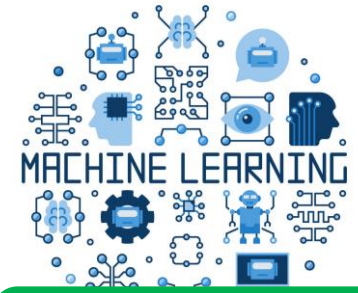
Conclusion : challenges du ML en Système embarqué

Roles & competencies

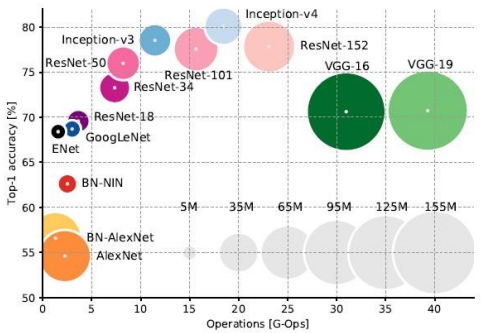
- *Data Physicist*
- *System Engineering team*
- *ML Engineer*
- *Software Engineer*
- *Hardware Engineer*
- *Infra & Security teams*

Tools

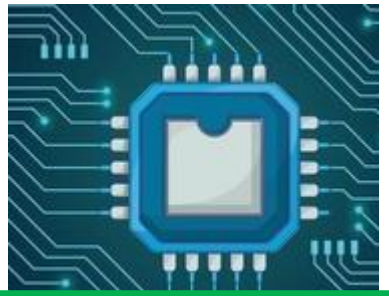
- *ML Tools:*
  - *TF-KERAS, PyTorch …*
- *HLS4ML (Xilinx…)*
- *HLS*
- *Brevitas & FiNN(Xilinx)*
- *CONIFER (LLR)*
- *N2D2 (CEA)*
- *VHDL*
- *C/C++*
- *SDK dédié*



Artefacs & ML zoology



- *Model*
- *Code source…*



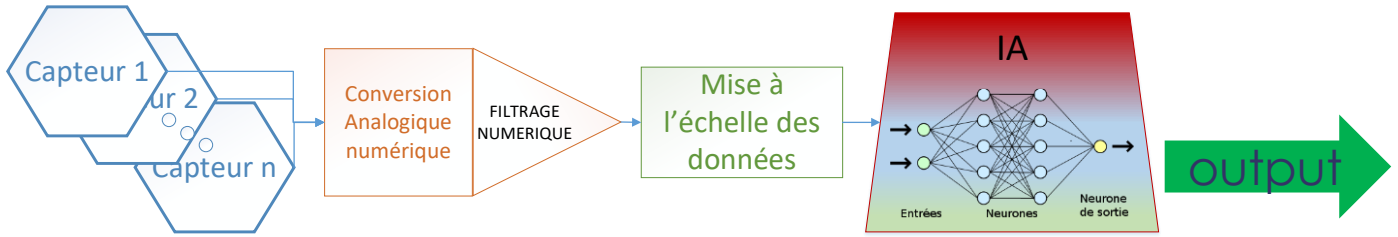
Digital hardware technologies

- *CPU + CoPU*
- *FPGA SOM*
- *SNN*
- *MPPA*
- *GPU*
- *…*

Deployement & Operational AI

- *GitLab/Git*
- *Training Service skew*
- *Model Monitoring*
- *Responsible AI*
- *…*

OSCAR: L'architecture numérique générale / idéale



RISC-V®

AMD | XILINX | altera An Intel Company

D_i

Extraction des caractéristiques

N entrées

Apprentissage profond: ANN, RNN, CNN…

P sorties

Formatage humain : Vecteur, matrice, images

IHM Analyses humaine

µPs

FreeRTOS

Linux…

Compression & stockage

Mémoires

OSCAR:
ASIC multivoies pour lire tous types de détecteurs

Conclusion

1 Mparam

Edge AI

ESP32

max78000

stm32

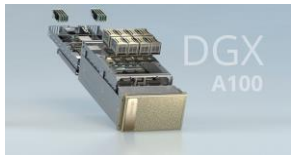
Embedded system

FPGA/ SoC

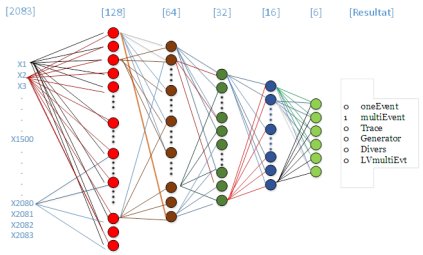
Puce neuromorphique

1 Gparam

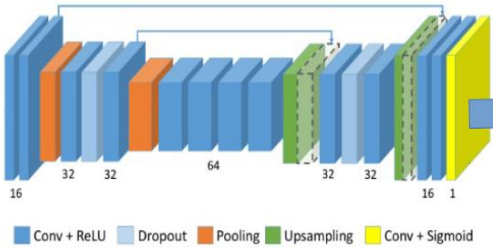
GPU

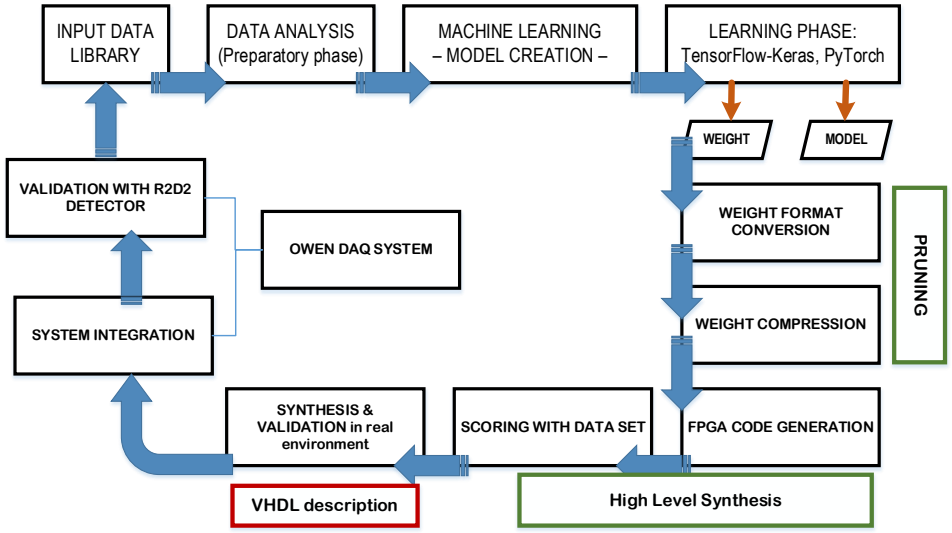


DEEP NEURAL NETWORK



CONVOLUTIONAL NEURAL NETWORK





Quelques conseils

- Ne travaillez qu'avec les modèles dont vous comprenez réellement le fonctionnement mathématique.
- Privilégiez les modèles les plus rapides et les plus simples (DNN et CNN)
- Donnez-vous une estimation de la latence du modèle.
- Tenez compte des transformations des données pour optimiser le modèle (opération supplémentaire)
- Créer des modèles autour de 100k paramètres maximum.