ECOLE IN2P3 : Techniques de base des acquisitions multi-détecteurs

l'électronique du détecteur à la mesure
III – IA (embarqué) pour la physique

LP2i Laboratoire de Physique des 2 infinis Bordeaux

Frédéric Druillole – LP2IB -

Sommaire

- Le contexte de l'IA
- Les grands principes
  - Definitions
  - Méthodes
  - Rôles
- Les Techniques
  - Solutions
  - Algorithmes, principes
  - Zoologie de l'IA
- L'IA dans nos activités
  - IA embarqués
  - Methodologie
  - Premier resultats
- Le Futur …

Définition

« *L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique.*

Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains. »

Pour y parvenir, trois composants sont nécessaires :

- ***Des systèmes informatiques (Matériels)***
- ***Des données avec des systèmes de gestion (Simulation/instrumentation)***
- ***Des algorithmes d'IA avancés (code)***

Pour se rapprocher le plus possible du comportement humain, l'intelligence artificielle a besoin d'une quantité de données et d'une capacité de traitement élevées. »

Système embarqué

(https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/)

La bataille de l'informatique

SCIENCES EXPERIMENTALES | SCIENCES THEORIQUES | SCIENCES CALCULATOIRES

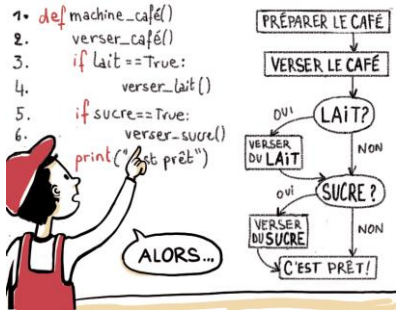
1600 1950

Informatique : traitement automatisé de l'information

Représentation de l'information

Algorithmes

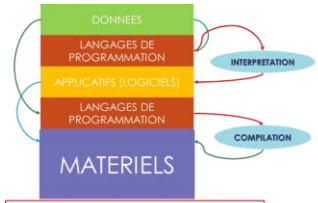
- Symbolisme
- connectivisme
- Probabilisme
- Analogisme



➔ Force brute
➔ Prog. Dynamique
…

➔ Diviser et régner
➔ Glouton
…

Matériels (Microprocesseurs)
Architecture de Von Neumann

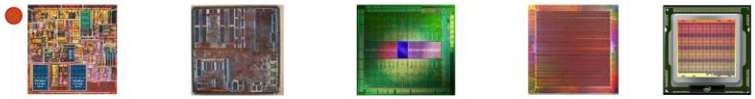


Ordinateur actuel
Central Processing Unit

Ordinateur quantique
Langage spécifique

Calcul haute performance

Langage C, C++ et outils spécifiques de compilation



CPUs DSPs GPUs FPGAs NMC

STRUCTURES DE DONNEES

Fichiers
Base de données

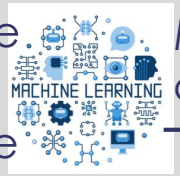
- ☺ Table de hachage
- ☺ Listes
- ☺ Listes chainées
- ☺ Vecteurs
- ☺ Pile / Tas
- ☺ Tableau N dimension (Tenseur)

La bataille de l'informatique

SCIENCES EXPERIMENTALES → SCIENCES THEORIQUES → SCIENCES CALCULATOIRES → **SCIENCES PILOTEES PAR LES DONNEES**

1600 — 1950 — 2000

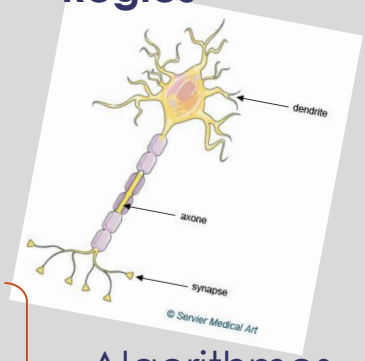
Approche inférentielle (Inductive)

Données d'entrée
Données de sortie


Modèle d'inférence

MESURES
FAITS
Informations

Lois
Règles



connectivisme
+ Probabilisme
+ Analogisme

Algorithmes apprenants

vs

Approche déductive

Données d'entrée
ALGORITHMES

Données de sortie

EXPERTS:
LOIS
Règles

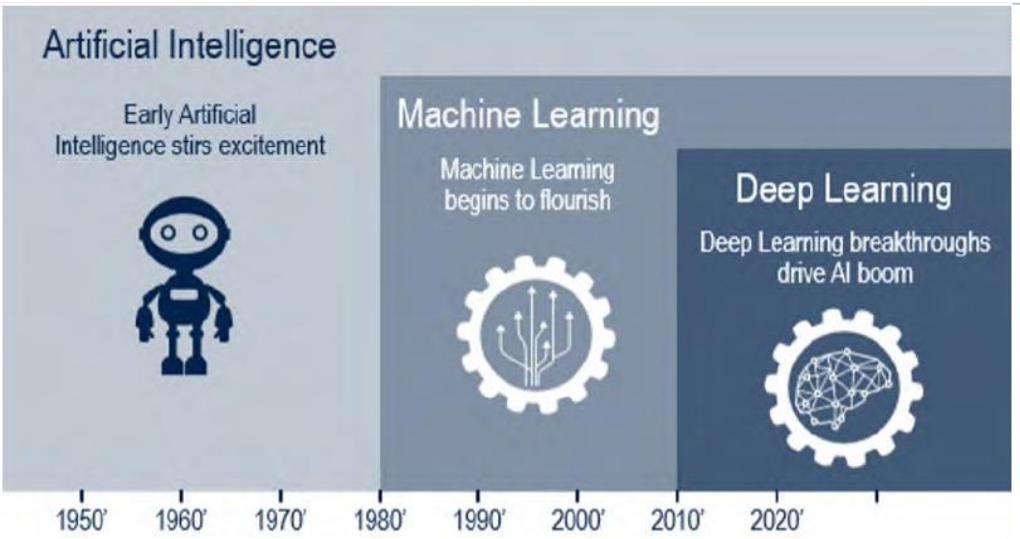
Traitement de cas particuliers:
Logiciels spécialisés

Symbolisme

Boucles
Tests logiques

```
 if (strncmp(cmd, "ADS:SPI:REGLIST", 15) == 0) {
if (Verbose > 0) {
xil_printf("**************************************\r\n");
xil_printf("* User R2D2 OWEN SPI LIST REGISTER   *\r\n");
xil_printf("**************************************\r\n");
}
if (sscanf(cmd, "ADS:SPI:REGLIST %s", str_array) != 1) {
sprintf(resp, " ERROR: ADS:SPI:REGLIST command failed !! \r\n");
er = -6;
return XST_FAILURE;
} else {
spiwordvalue = (u16_t*)realloc(spiwordvalue, 16*sizeof(u16_t));
strcpy(cp_array, str_array);
nb_parameter = sscanTabInt(cp_array, spiwordvalue);
xil_printf("spidata: nombre de parametres : %d\n", nb_parameter);
xil_printf("spi reg =[ ");
            for (int i= 0; i < nb_parameter; i++) {
            xil_printf("%x, ", spiwordvalue[i]);
            }
            xil_printf(" ]\r\n");
            sprintf(resp, "spi data list: : %s\r\n", str_array);
}
```

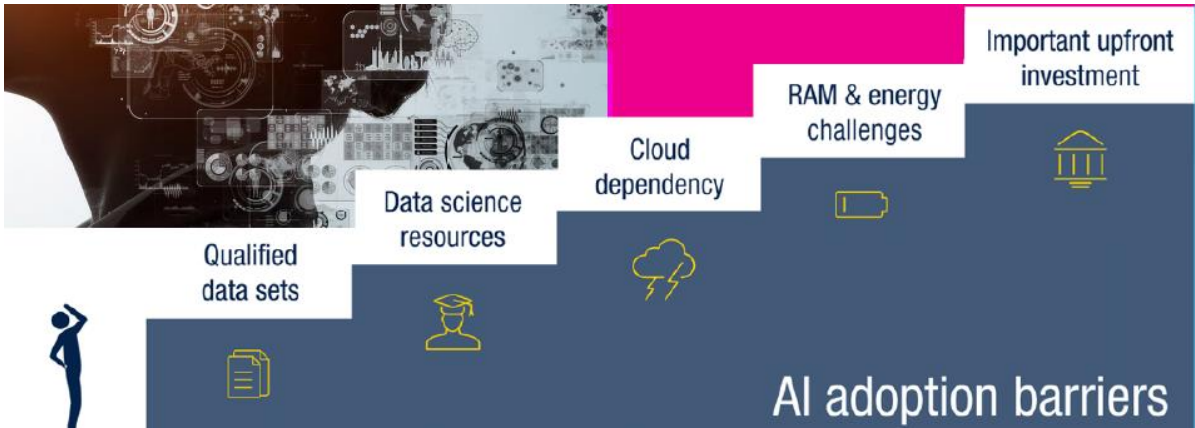
Historique



➔ **Evolution et progression de l'IA**

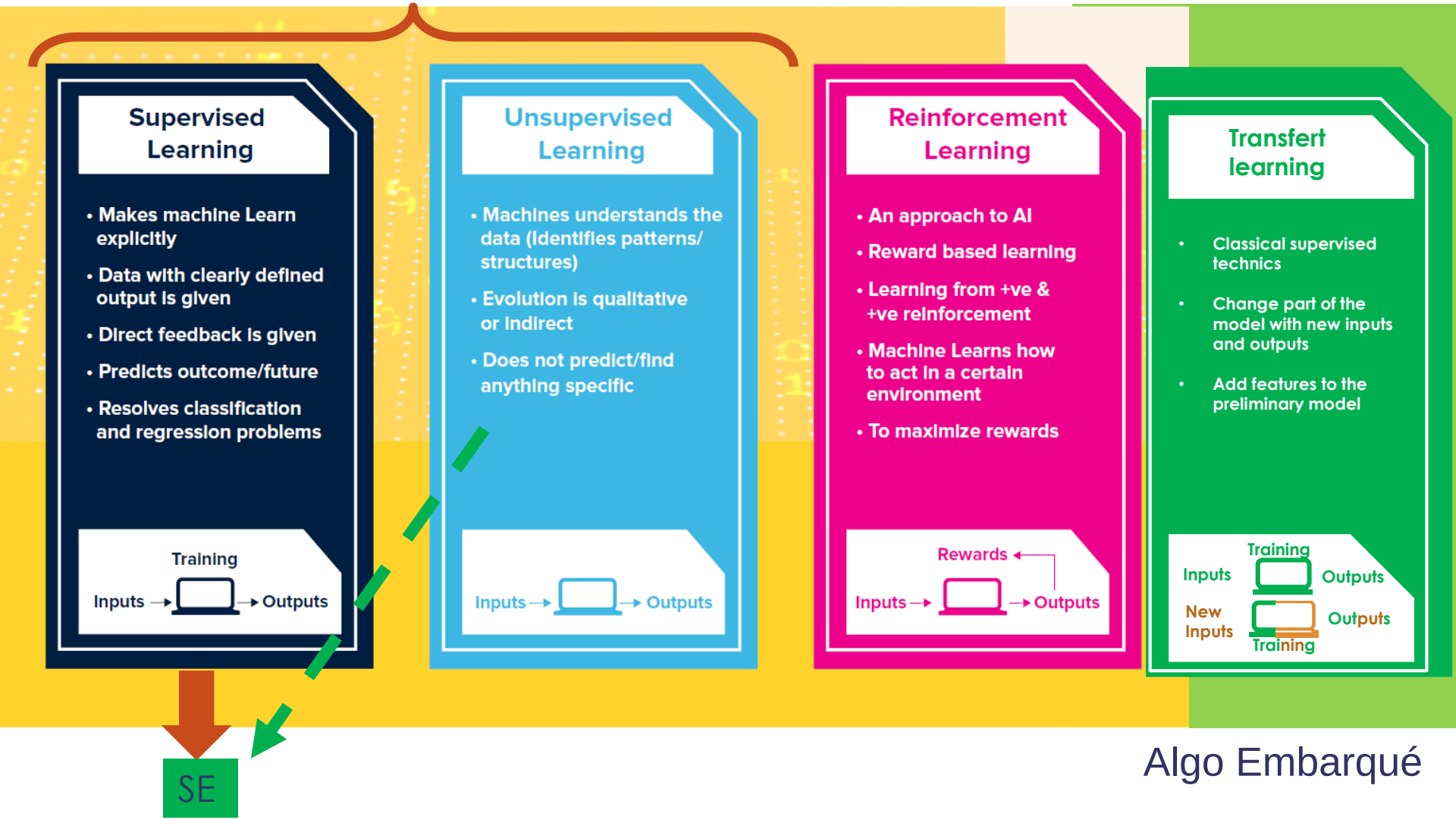
➔**Les barrières pour adopter l'IA dans les produits pour les sociétés.**
-investissement en RH, matériels, données
-Grandes compagnies

➔ **ML dans SE : solution économique**



Les quatre catégories d'apprentissage profond

D:\02-Projets\Cours_IA_Embarqué\10-Formation_code\deep-reinforcement-learning-python-main\RL_code\snake-ai-pytorch-main



Algo Embarqué

Machine apprenante locale ➔ Responsive AI on the Edge

IA cloud centralisé

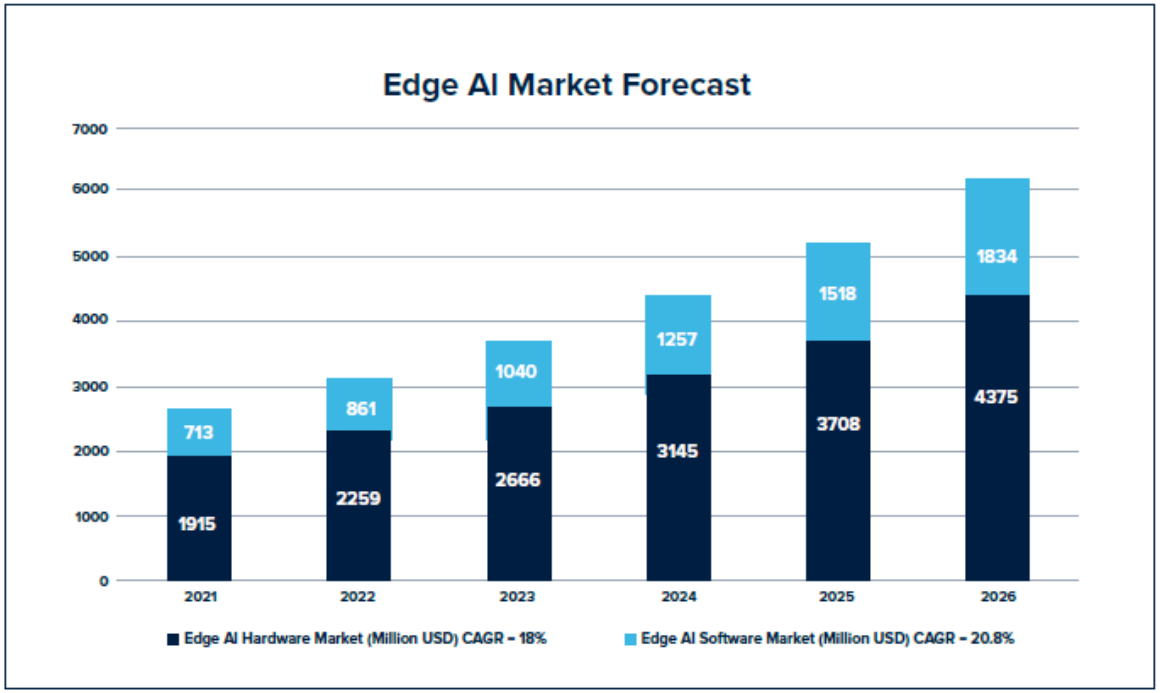
Grandes compagnies

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

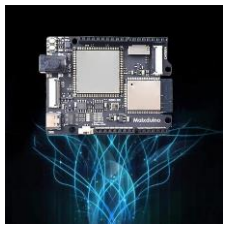
Migration

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

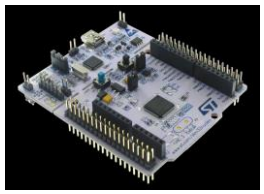
STMicroelectronics 2022



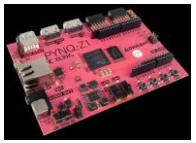
Kendryte K210



STM32 Cube AI

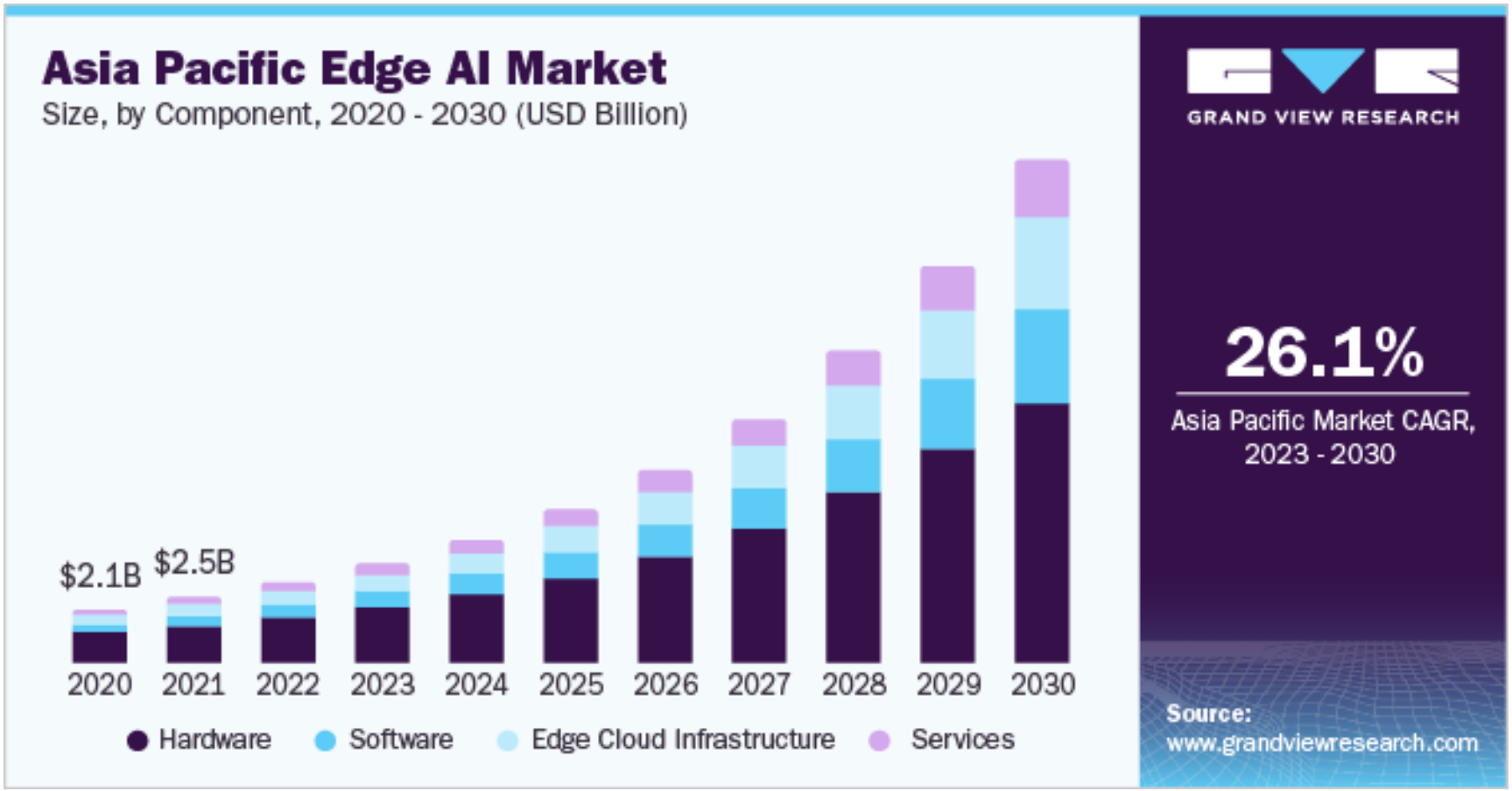


Digilent



AMD-Xilinx

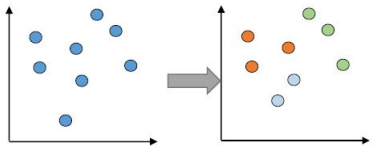
Intel

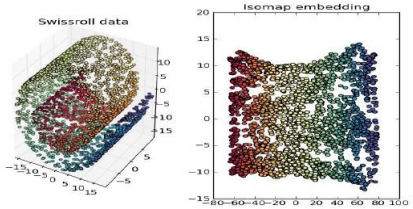

Asia Pacific Edge AI Market
Size, by Component, 2020 - 2030 (USD Billion)
$2.1B
$2.5B
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
Hardware
Software
Edge Cloud Infrastructure
Services
GRAND VIEW RESEARCH
26.1%
Asia Pacific Market CAGR, 2023 - 2030
Source:
www.grandviewresearch.com

Grands Principes de l'IA

- Méthode:
  - Définir un modèle suivant l'objectif (Classification / Régression/generation)
  - **Entraine** le modèle avec des données connues [y=f(x)]
  - Utilise le modèle avec des données de **test** [y=f(x)]
  - **Valide** le modèle sur des données inconnues [G(x) =/!= ? y

- **Apprentissage non supervisé**
  - On a des données d'entrée mais pas de données de sortie
  - L'objectif est de généraliser une structure, des caractéristiques qui permettront de répondre à la question posée. [Est-ce f(x) = a vrai ?]



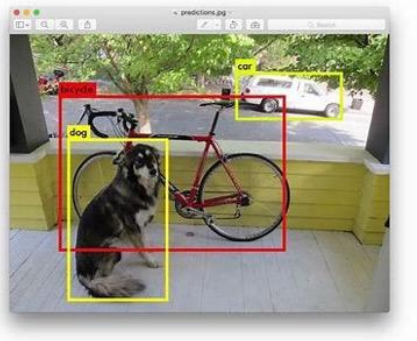


- **Apprentissage supervisé : Deep Learning**
  - On a un jeux de données avec:
    - Les données d'entrée
    - La réponse (travail d'expertise) associée aux données et à la question posée ➔ [y=f(x)]

Grands Principes de l'IA

- ☺ Machine Learning = Apprentissage automatique
- ☺ Experts versus Trouver un modèle prédictif

Exemple :



L'Expert sait à travers ses connaissances ce qu'est un chien et un vélo. Il peut mettre une étiquette sur les objets de cette image.

Le Machine Learning essaye de déterminer des paramètres (caractéristiques) de l'image et **infère** une règle qu'il appliquera à une entrée.

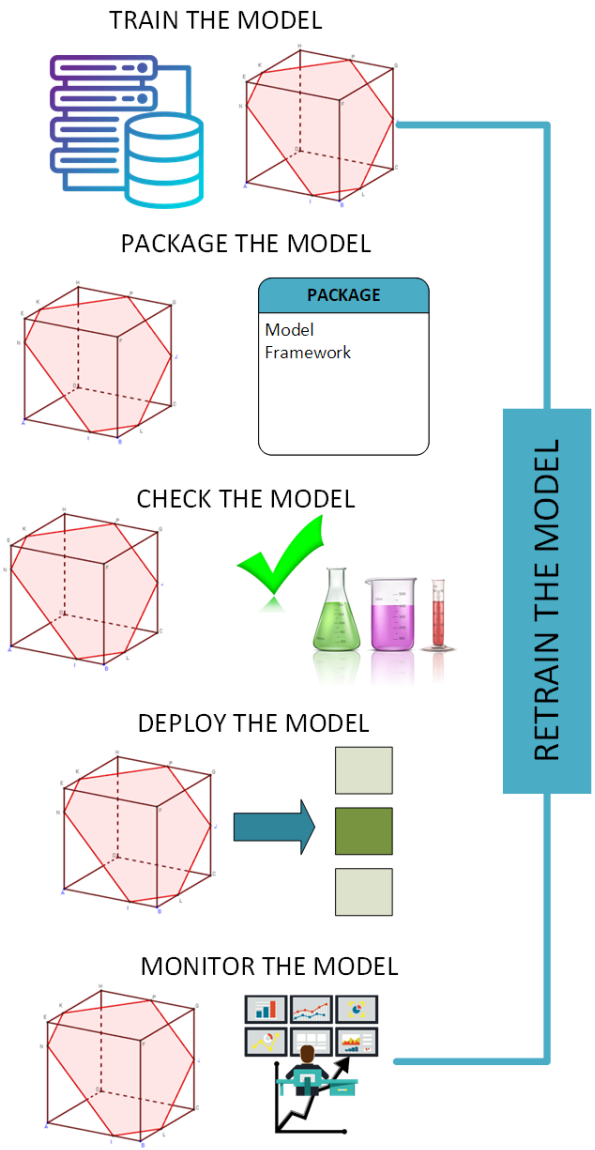
Comment fait-on cela ?

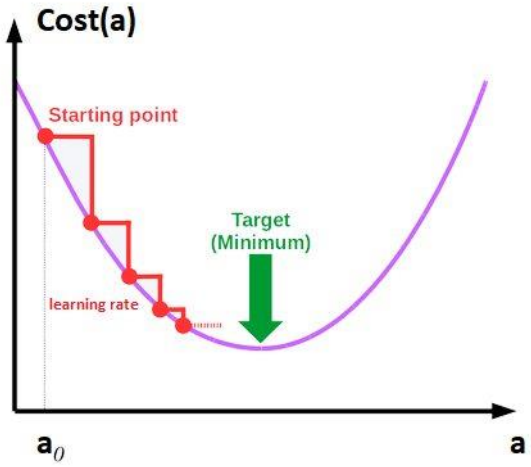
A partir de données d'entrée, il mémorise les caractéristiques des données. Quand il a en entrée une nouvelle donnée, il prédit une solution par rapport à la question qu'on lui a posé/entrainé à répondre.
➔ Il fera des erreurs mais l'objectif est de réduire son taux d'erreur en jouant sur l'**architecture** et les **hyperparamètres** du machine learning.

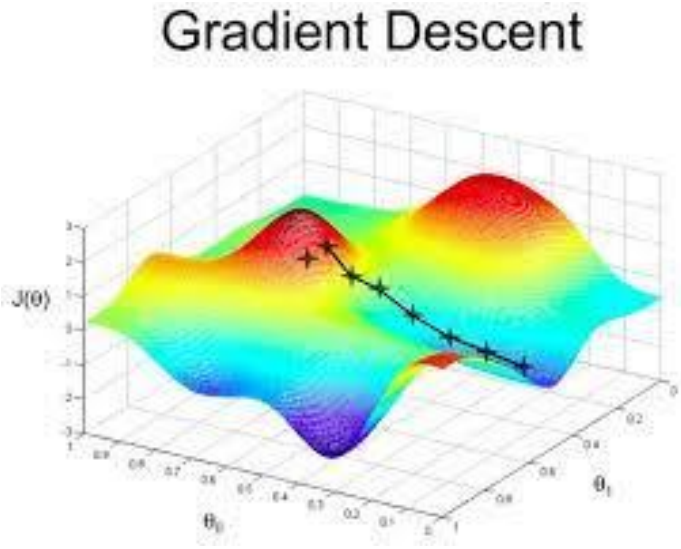
Le Machine Learning consiste à extraire à partir de données des **règles générales** qui permettent de répondre à un problème suivant de nouvelles entrées. **Généralisation**

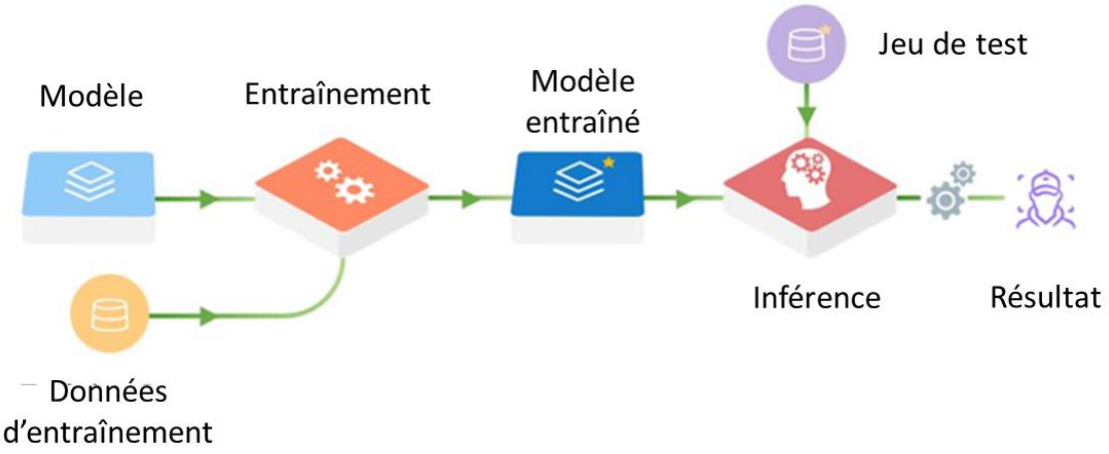
Méthodes

MLOps

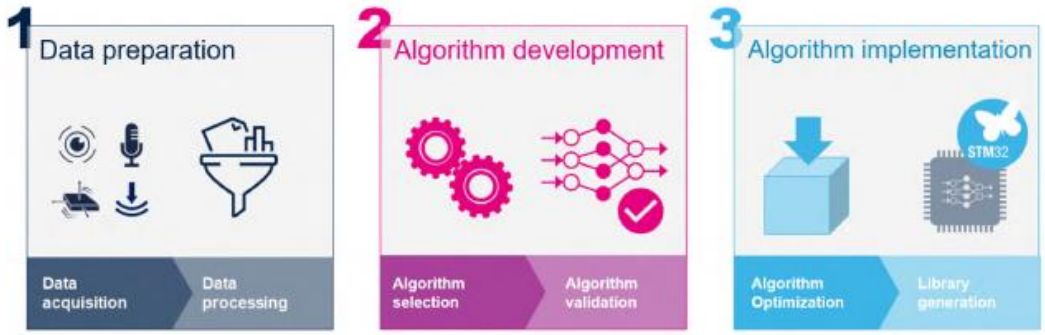






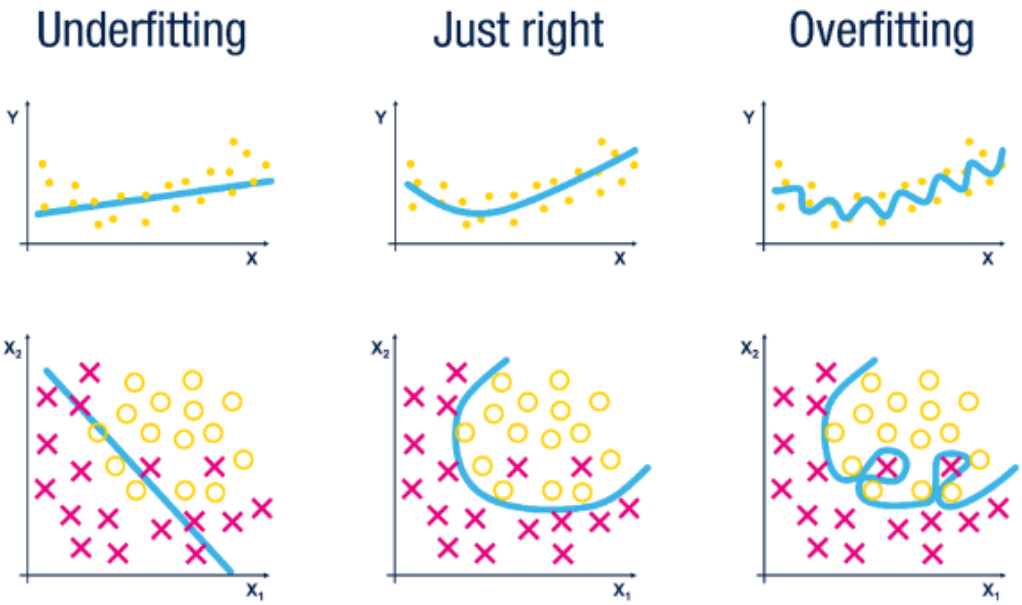


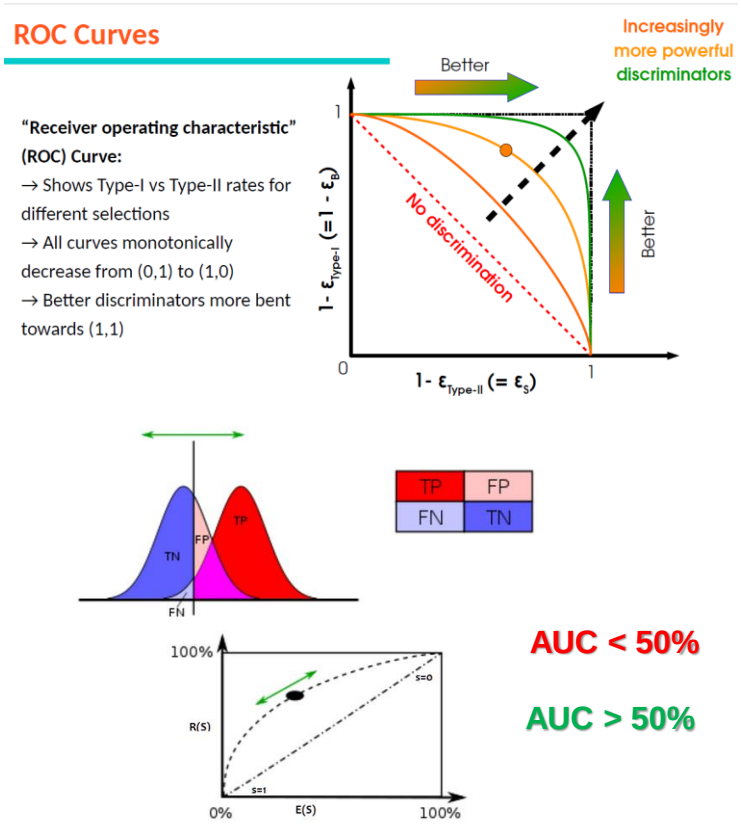
Analyse des performances des modèle IA



Courbe ROC et AUC (Test d'hypotheses)

Hyp/resultat	H0 faux	H0 Vrai
H0 faux	OK	KO (Type-II error)
H0 Vrai	KO (Type-I error)	OK





les Défis de l'IA embarqué: Une culture à avoir.

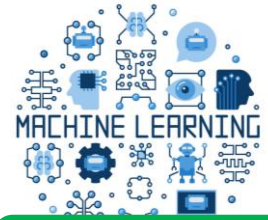
Roles & competencies

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

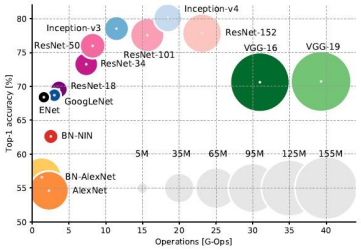
Tools

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

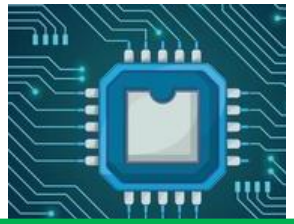
…



Artefacs & ML zoology



- *Model*
- *Code source…*



Digital hardware technologies

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

Deployement & Operational AI

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

Les techniques de l'IA

- Traitement des images et des vidéos (computer Vision)
  - Identification d'objets
  - Capture d'image
  - Etc…
- Traitement du langage naturel
  - Classification de texte
  - Lecture automatique
  - Reconnaissance de mot
  - Traduction
- Structuration des données
  - Classification des informations
  - Proposition (recommandation)
  - Maintenance
  - Détection de fraude

Les techniques de l'IA

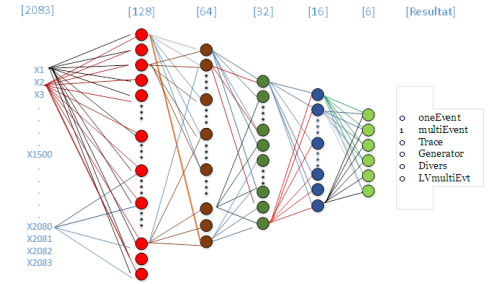
- Séries temporelles (LLM, ChatGPT)
  - Prévision des données dans une serie temporelle de données (météo)
  - Classification des données en temps réel
- Données Audio
  - Reconnaissance de la parole
  - Reconnaissance des personnes suivant leur conversation
- Générateur d'information en apprentissage profond
  - Génération d'images
  - Génération d'information (texte, nombre …)
- Apprentissage par renforcement
  - Robotique
  - Simulation suivant meilleur choix
- Traitement des données avec graphes (chaînes)
  - Classification de nœuds
  - Arbre de décision
  - Algorithme génétique

Imaginer comment se servir de ces techniques dans vos futurs domaines

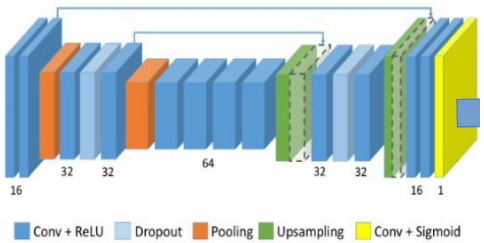
En Instrumentation, en IoT, avec détecteurs/capteurs

WHAT WE COULD DO WITH ML

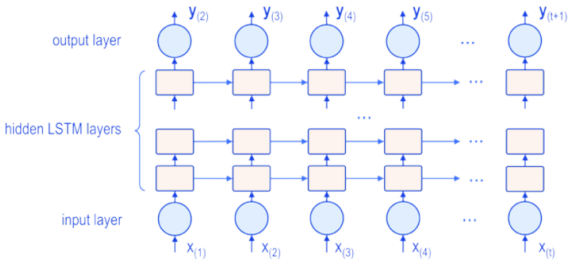
DEEP NEURAL NETWORK



CONVOLUTIONAL NEURAL NETWORK



RECURRENT NEURAL NETWORK



DECISION TREE

RANDOM FOREST

Off-line

- Signal generation
- Design Optimisation

On-line

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

L1

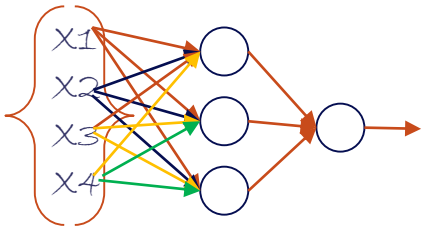
L2

L3

Apprentissage supervisé : Réseau de Neurones

X
particule

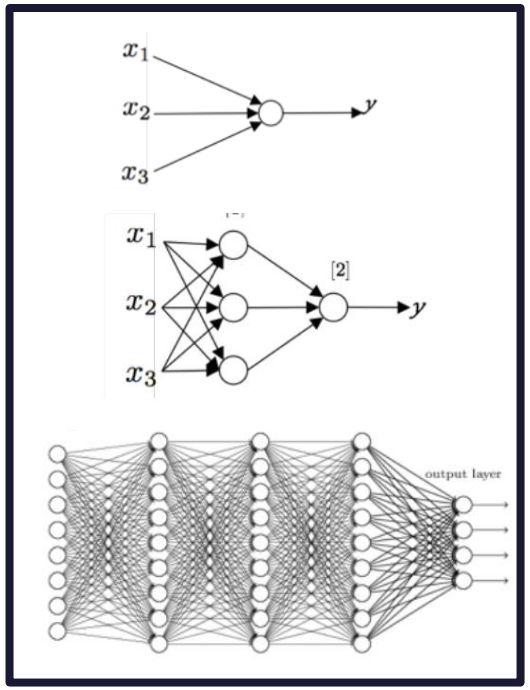
y
Identifie la particule



Big Data : grand nombre de données
Gros modèles de taille importante de neurones
un grand nombre d'exemples d'entrainement

Puissance de calcul : CPU/GPU/FPGA….

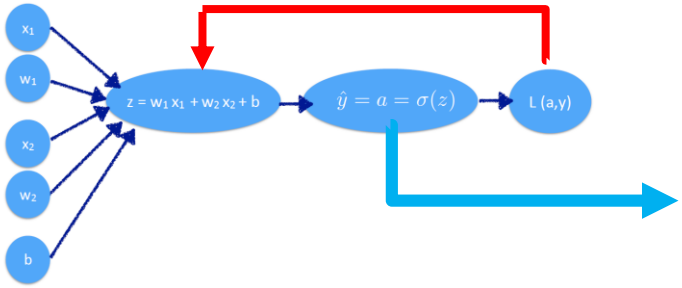
Innovation dans les algorithmes (la descente du gradient)



$z = w^T x + b$
$\hat{y} = a = \sigma(z)$

Loss function:

$L(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$

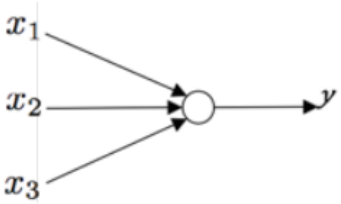


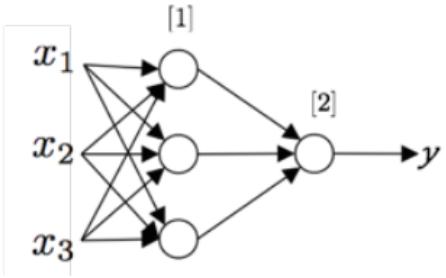
ACTIVATION FUNCTIONS

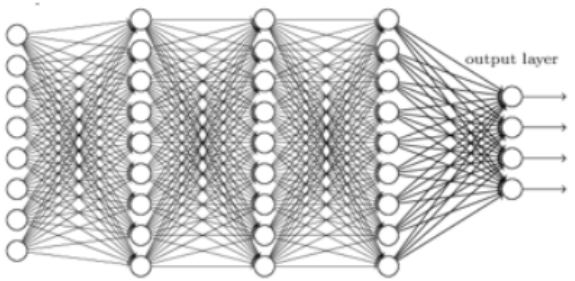
Name	Plot	Equation
Sigmoid		$g(z) = \frac{1}{1+e^{-z}}$
Hyperbolic tangent		$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$
Rectified linear unit (ReLU)		$g(z) = \max(0, z)$
Leaky ReLU		$g(z) = \max(\alpha z, z)$

Using a linear activation function is useless un more than one layer networks. It's rarely used

DEEP NEURAL NETWORKS







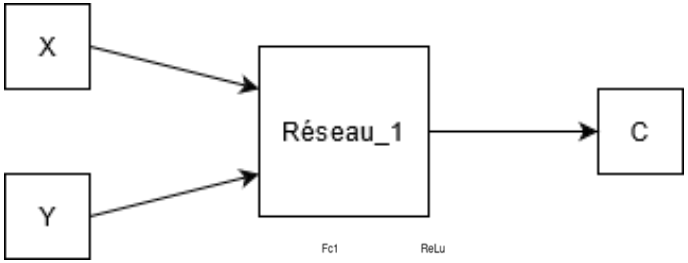
Shallow

Deep Neural Network

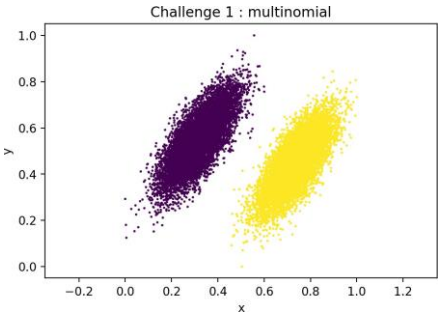
Need to define:

- L : number of layers,
- $n^{[l]}$: number of nodes (units) in layer l,
- $a^{[l]}$: *activations* in layer l,
- $a^{[L]}$: Output/ prediction

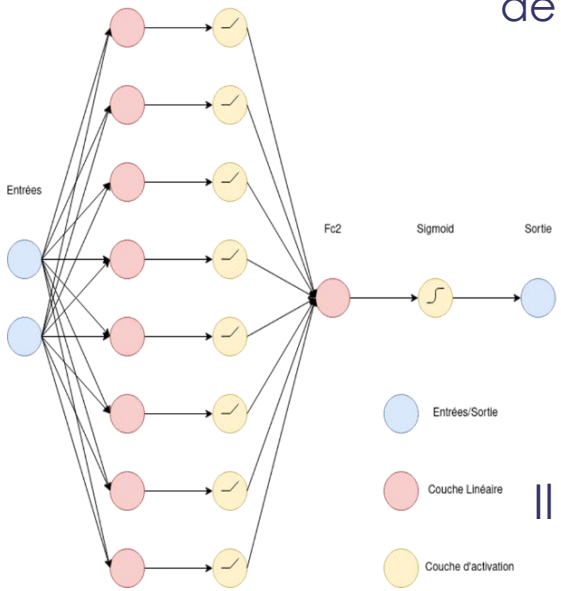
Un exemple de calcul: réseau de neurone peu profond

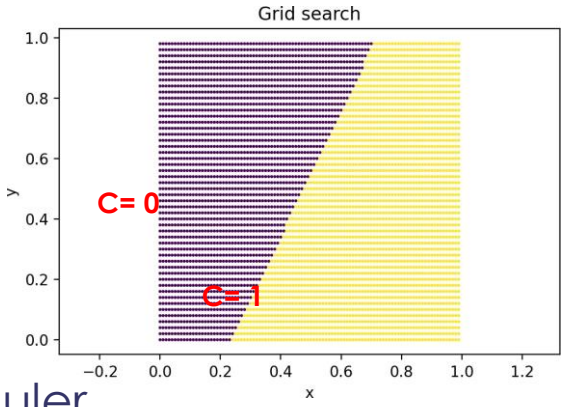


Principe: on fournit en entrée des coordonnées (x,y). Le réseau nous classifie les coordonnées selon la région d'appartenance. Ici, c'est une simple régression logistique avec une séparation de deux région linéaire.



2x8x1





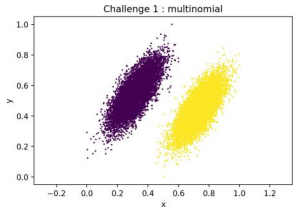
Il faut calculer

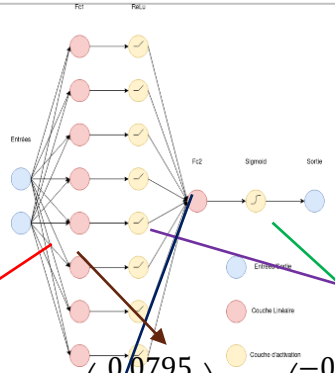
42 multiplications

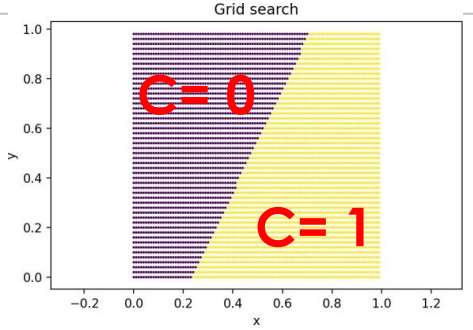
$$W = \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{pmatrix} \text{ et } B = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

$$s = \sigma(\mathbf{W}x + \mathbf{B})$$

Un exemple de calcul: réseau de neurone peu profond







X=0,2
Y=0,7

$$\begin{pmatrix} 1.7856 & -1.2345 \\ 2.4834 & -0.5797 \\ -0.3130 & 0.2658 \\ 0.3110 & -0.4116 \\ -0.4458 & 0.8010 \\ 3.4282 & -1.7166 \\ -0.6808 & 0.1631 \\ -0.5633 & 0.5694 \end{pmatrix} \text{X} \begin{pmatrix} 0.2 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.0795 \\ -0.5202 \\ 0.4716 \\ -0.4349 \\ 1.3222 \\ -0.2305 \\ -0.3073 \\ 0.9498 \end{pmatrix} = \begin{pmatrix} -0.4275 \\ -0.4294 \\ 0.5951 \\ -0.6608 \\ 1.7938 \\ -0.7465 \\ -0.3293 \\ 1.2358 \end{pmatrix}$$

$$\begin{pmatrix} -0.4275 \\ -0.4294 \\ 0.5951 \\ -0.6608 \\ 1.7938 \\ -0.7465 \\ -0.3293 \\ 1.2358 \end{pmatrix} \xrightarrow{ReLu} \begin{pmatrix} 0 \\ 0 \\ 0.5951 \\ 0 \\ 1.7938 \\ 0 \\ 0 \\ 1.2358 \end{pmatrix}$$

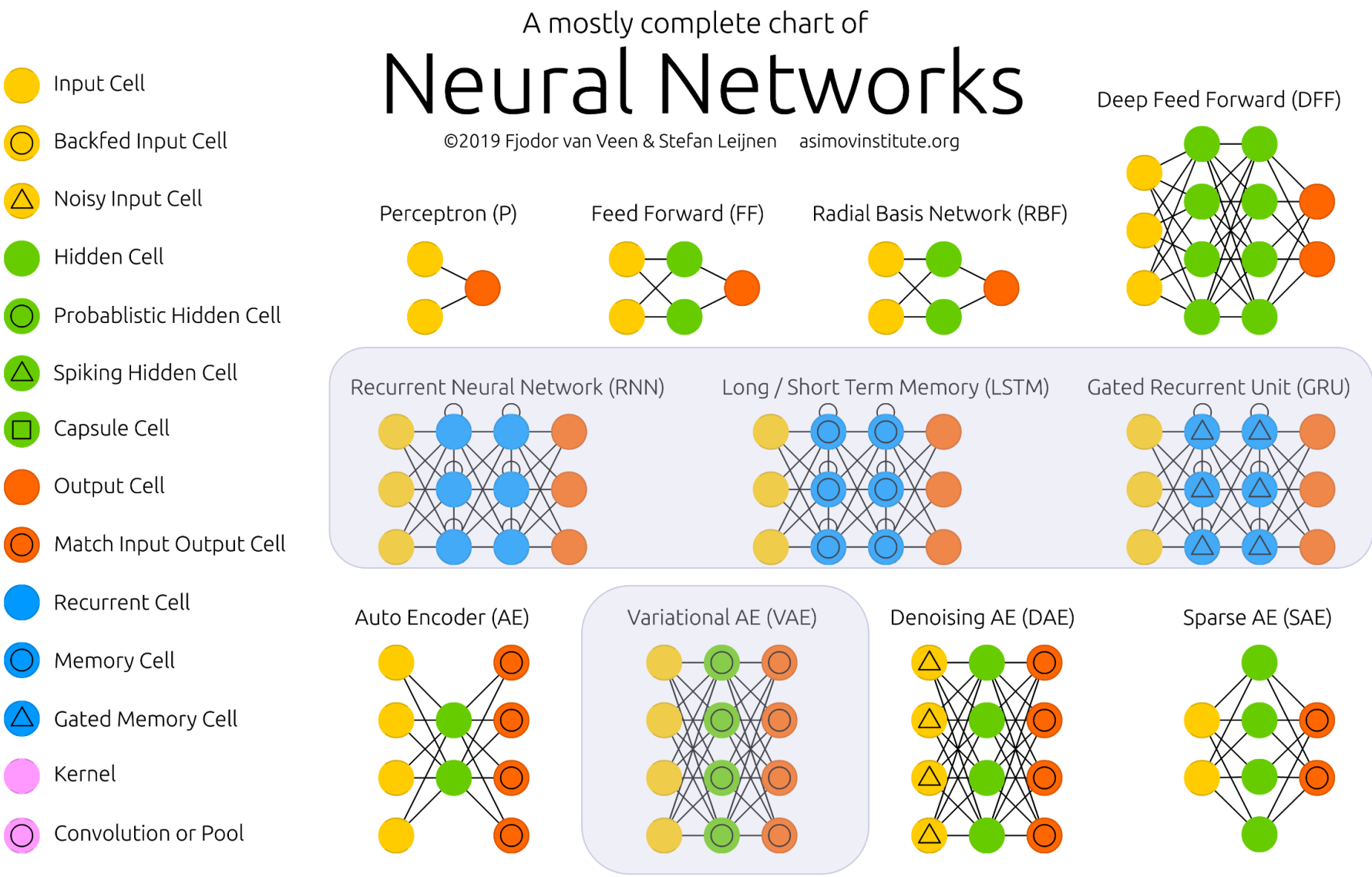
$$(2.0441\ 2.5193 - 0.6255\ 0.0535\ - 1.5080\ 3.7406\ 0.1052\ - 1.2330)\ \text{X} \begin{pmatrix} 0 \\ 0 \\ 0.5951 \\ 0 \\ 1.7938 \\ 0 \\ 0 \\ 1.2358 \end{pmatrix} + (-1.4529) = (-6.0539)$$

$$(-6.0539) \xrightarrow{sigmoïde} (0.0023) \sim 0$$

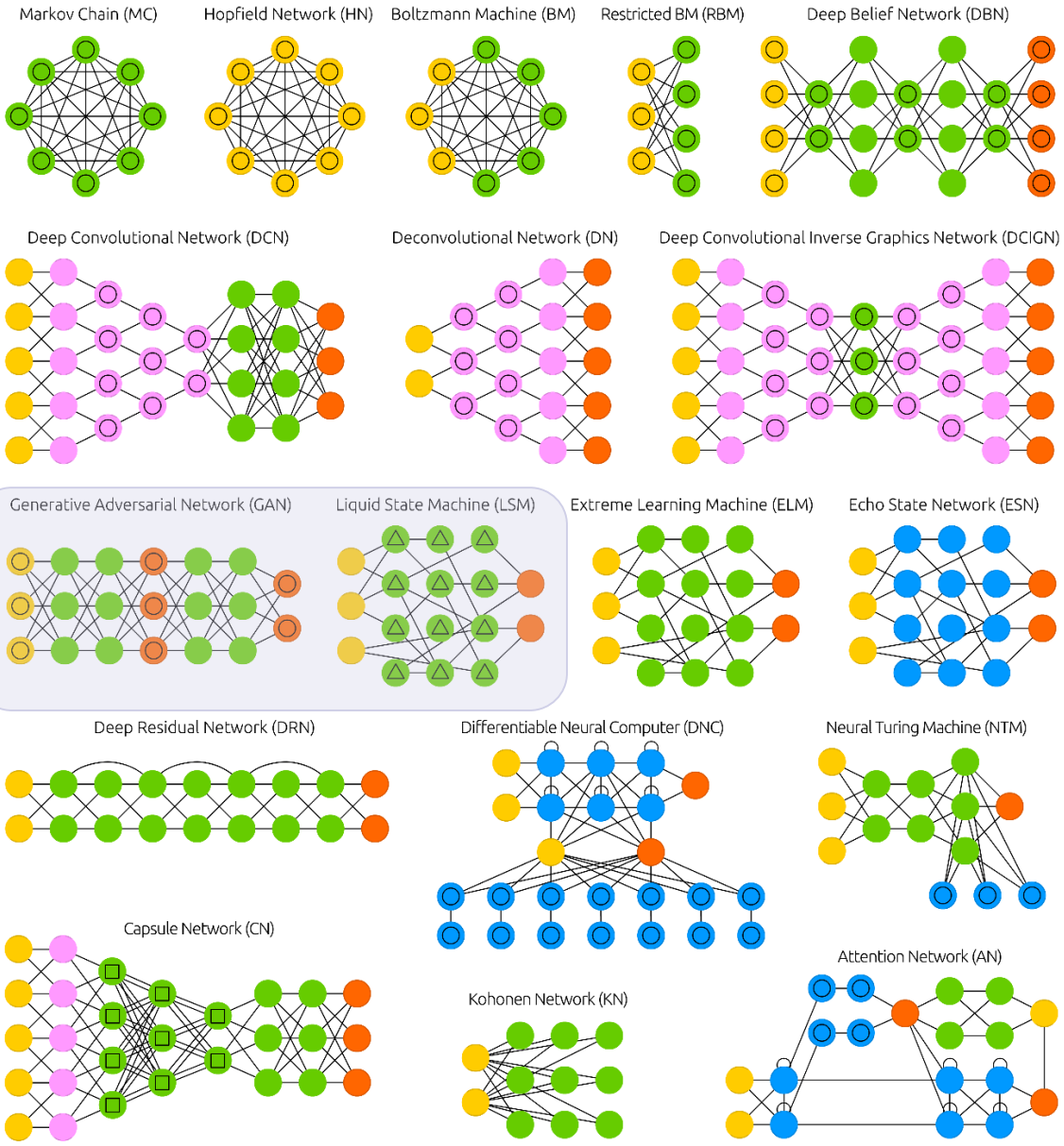
X=0,2
Y=0,7

Appartient à la zone **C= 0**

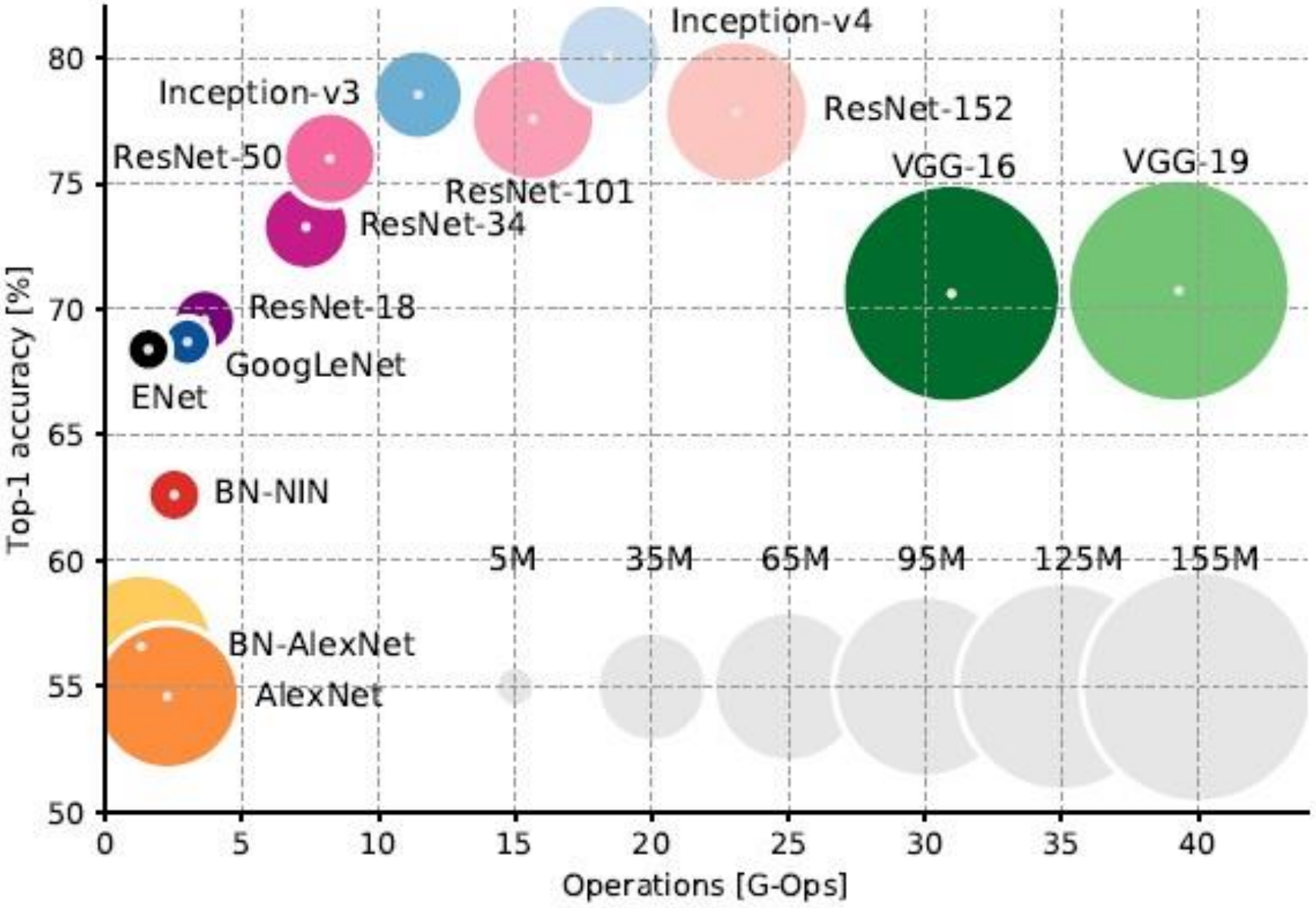
Zoo des NNs

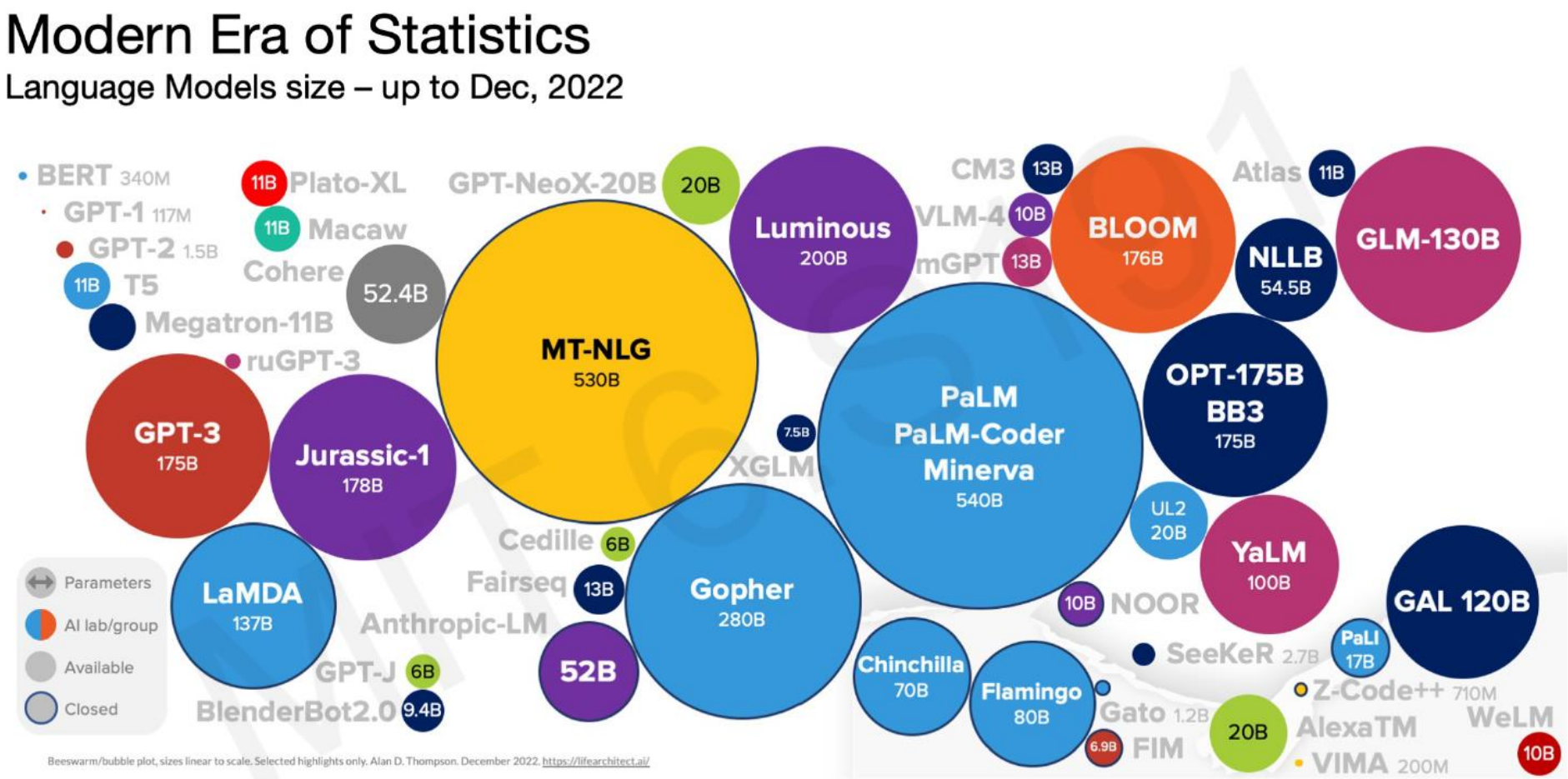

Input Cell
Backfed Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Capsule Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Gated Memory Cell
Kernel
Convolution or Pool
A mostly complete chart of
Neural Networks
©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)

Zoo des NNs


Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Differentiable Neural Computer (DNC)
Neural Turing Machine (NTM)
Capsule Network (CN)
Kohonen Network (KN)
Attention Network (AN)

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-J 6B
BlenderBot2.0 9.4B
MT-NLG 530B
GPT-NeoX-20B 20B
Luminous 200B
Cedille 6B
Fairseq 13B
Anthropic-LM
52B
Gopher 280B
CM3 13B
VLM-4 10B
mGPT 13B
7.5B XGLM
PaLM PaLM-Coder Minerva 540B
Chinchilla 70B
Flamingo 80B
Gato 1.2B
6.9B FIM
BLOOM 176B
Atlas 11B
NLLB 54.5B
GLM-130B
OPT-175B BB3 175B
UL2 20B
YaLM 100B
10B NOOR
SeeKeR 2.7B
PaLI 17B
GAL 120B
Z-Code++ 710M
20B
AlexaTM
WeLM
VIMA 200M
10B
Parameters
AI lab/group
Available
Closed
Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. Alan D. Thompson. December 2022. https://lifearchitect.ai/

L'IA en physiques des particules, astrophysique et physiques nucléaires

Non exhaustifs

Boosted decision Tree :
Discrimination signals /bruit de fond
applications: CMS

Autoencodeurs et classifieurs:
Détection d'anomalies dans l'analyse en physique
Application: Atlas

Reduction de dimensions:
simples paramètres de representation pour l'analyse statistqiue classique
Discrimination signals/bruit de fond
applications: CMS

CNN, ANN:
Monitorage de la qualité des données (taux de trigger, latence…)
Applications: CMS

GAN, Variational autoencoder:
Simulation et modélisation d'interaction
Modèles génératifs

Implémentation de NN embarqués:
Triggers et réduction des données lors de l'acquisition
LHC, Dune

Image classification
Reconstruction des données des détecteurs
Analyse des jets

Apprentissage non supervisé (k-nn, SVM, U-Net):
Regroupement des données (clustering) pour la reconstruction de trace
Application: Dune

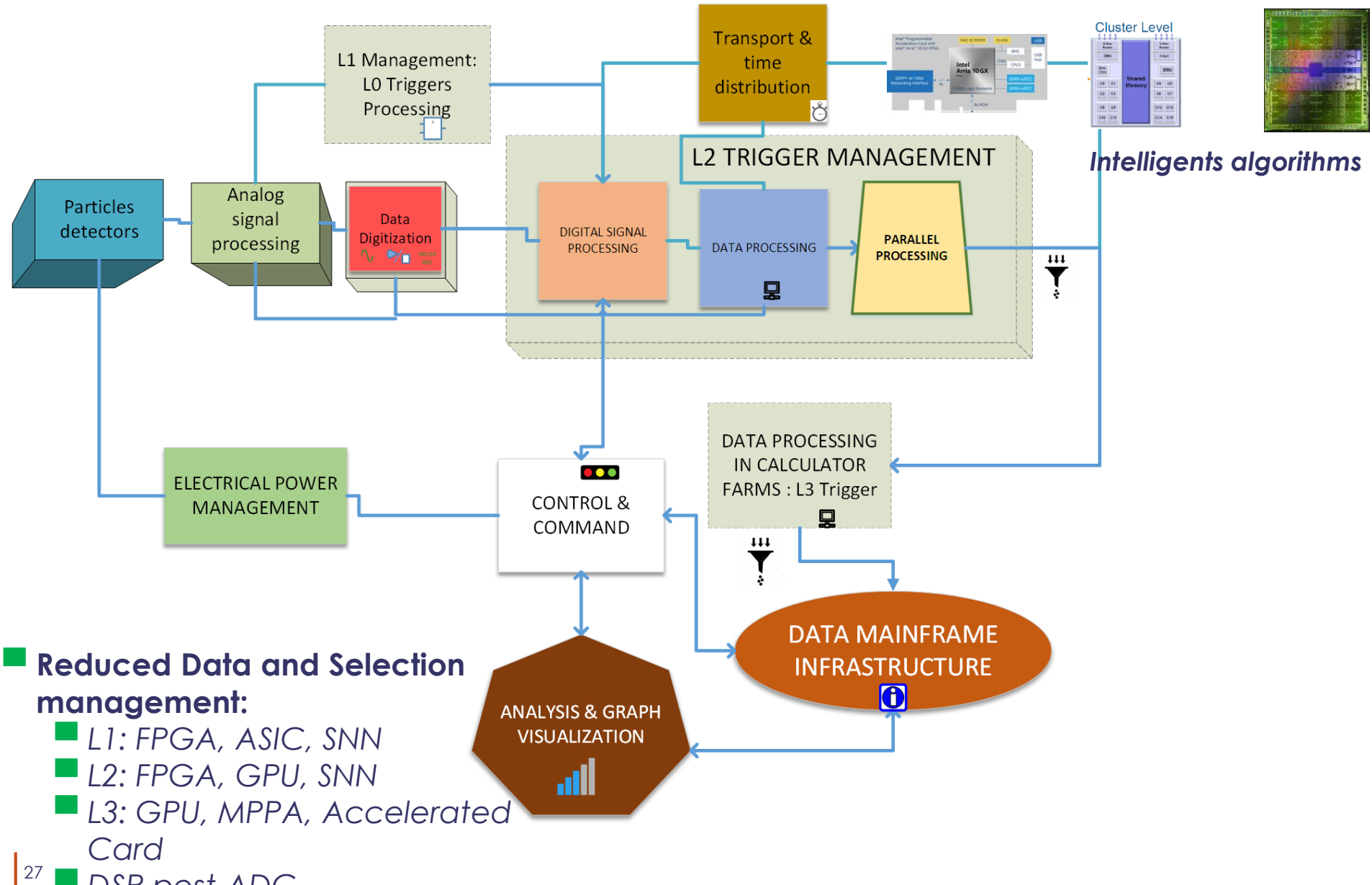
Graph-NN, Point-cloud NN:
Regroupement des données (clustering) pour la reconstruction de trace
Applications: JUNO, LHC, BELLE-II…

Intelligence Artificielle appliquée à la spectro-identification de radioéléments en environnement radiologique complexe.
Génération de données spectrales
Identification par CNN

Deconvolution, probleme mal-posé:
CNN, Autoencodeurs
Melange des galaxies ➔ de-blending

Accelerateurs: constructions
Aimants, monitorage d'anomalies,
Contrôle & commande

My research : Dominant Design in Instruments for research in fundamental physics



- **Reduced Data and Selection management:**
  - *L1: FPGA, ASIC, SNN*
  - *L2: FPGA, GPU, SNN*
  - *L3: GPU, MPPA, Accelerated Card*
  - *DSP post-ADC*

Challenges in the field

LHCb – 2032 ~2000 Exabytes/year
ATLAS+CMS 2027 ~ 260 Exabytes/year
Square Kilometers Array – 2030 ~ 30000 EB/year

2021 global Ethernet Dataflow ~2800 EB/year

DataStream before storage
LHCb – 2032 ~500TB/s
ATLAS+CMS 2027 ~ 20-40 TB/s

Forecast cost of storing data to disk (Annual)
LHCb – 2032 ~2,5 Billions of €
ATLAS+CMS 2027 ~ 325 Millions of €

Challenge: Real-time data reduction to avoid disk storage (very expensive):

➔ Embedded algorithms in decision nodes

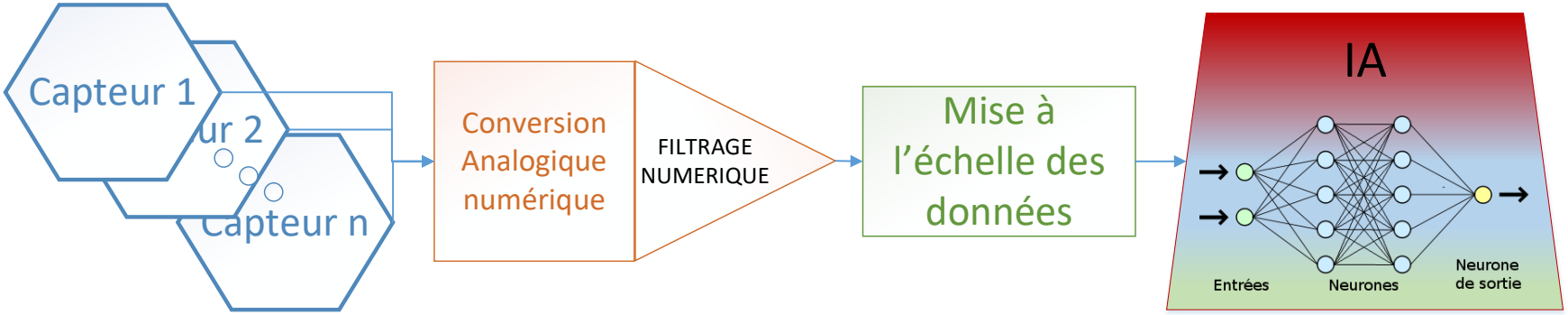
➔ Optimize processing (classifications, Prediction, Selection)

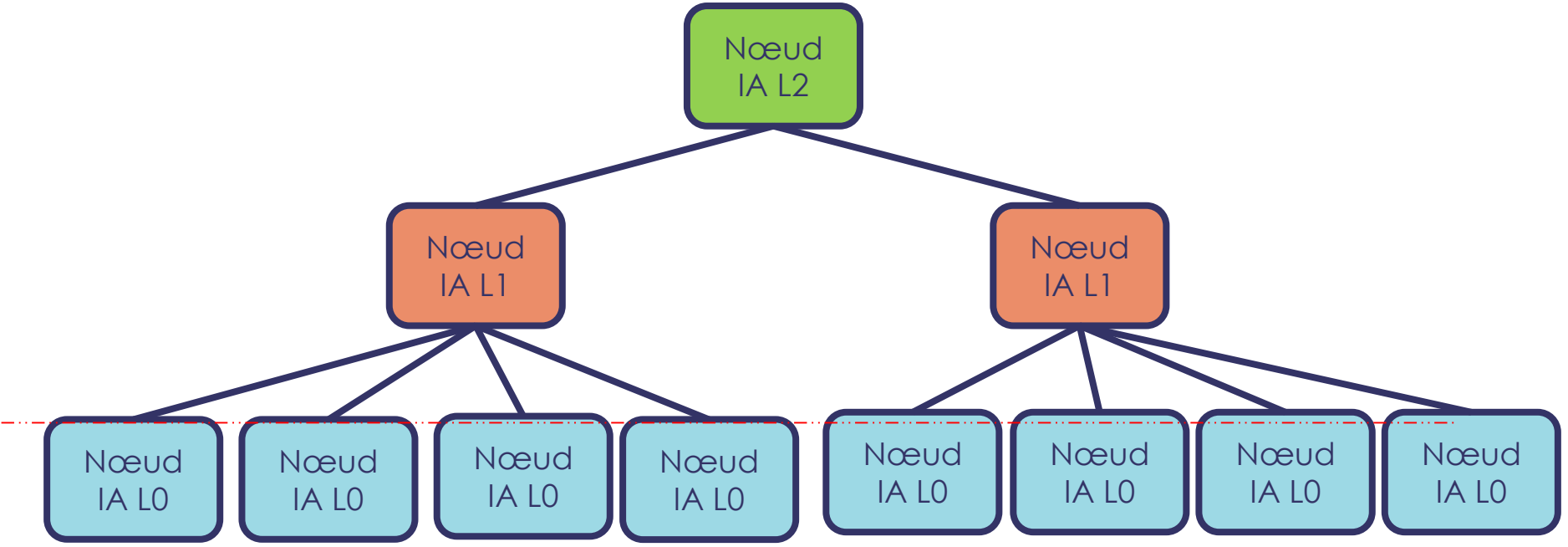
➔ Use a mixed GPU, MPPA, FPGA

- **Use powerfull hardware component to compute ML Model**
- **And deploy them in ours instruments**

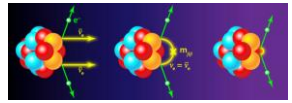
Mon objectif

Nœud IA Level 0

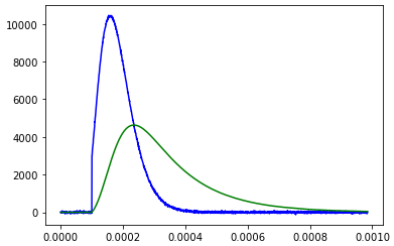




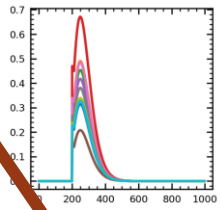
Générer des données réalistes rapidement



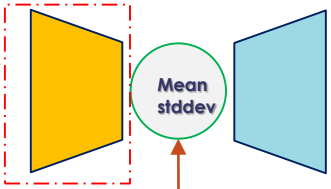
Issus des mesures : Modèle H(p)

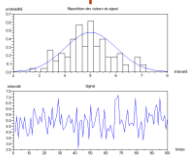


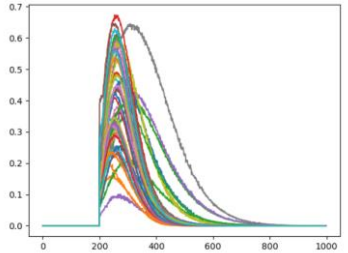
Cas limité du modèle

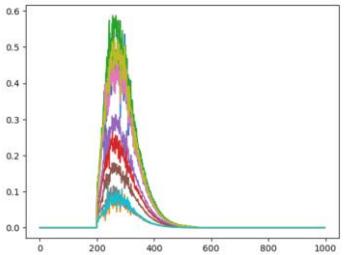


Variational **A**uto**E**ncoder



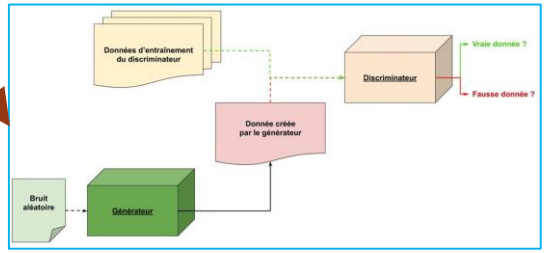






Génération de **cas génériques** avec ces deux modèles

Generative **A**dversal **N**etwork

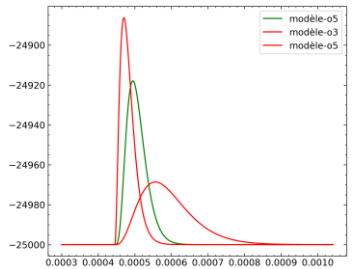


Millions de modèles de signaux

```
import sympy as sp
A,b = sp.symbols("A b", real=True)
G = A/(b*s+1)**3
h_t = inverse_laplace_transform(G, s, t)
print (h_t)
dh_t = sp.diff(h_t,t)
print(dh_t)
```

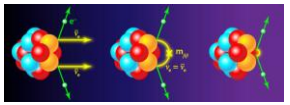
A*t**2*exp(-t/b)*Heaviside(t)/(2*b**3)

A*t**2*exp(-t/b)*DiracDelta(t)/(2*b**3) + A*t*exp(-t/b)*Heaviside(t)/b**3 - A*t**2*exp(-t/b)*Heaviside(t)/(2*b**4)

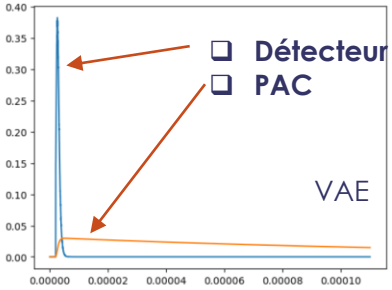


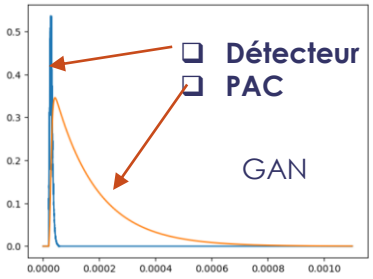
Débruitage et déconvolution (en cours)

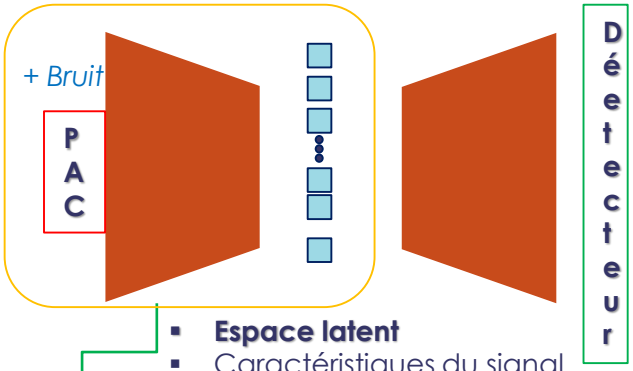
R2D2: Rare Decays with Radial Detector



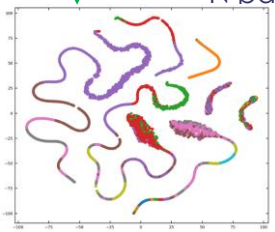
Preampli de charge







- **Espace latent**
- Caractéristiques du signal
- N paramètres

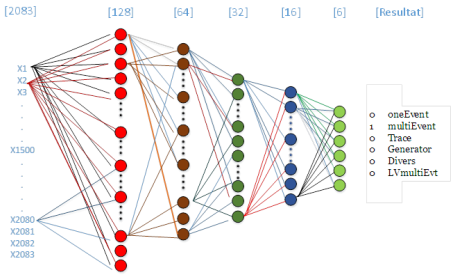


Exemple avec 6 paramètres

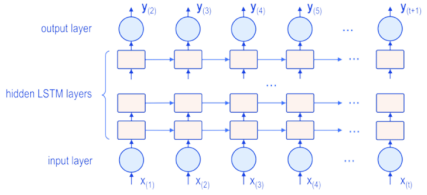
Déconvolution on line

Mise en œuvre dans FPGA

- Régression pour déterminer l'énergie
- Régression pour la mesure d etemps



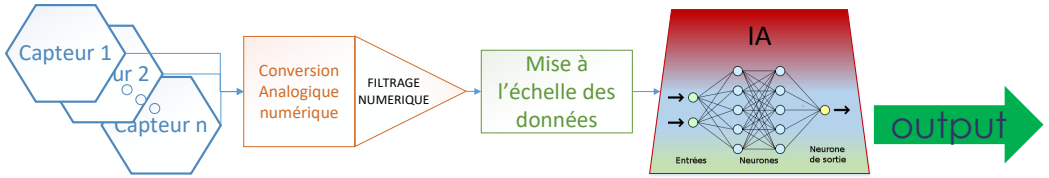
Classification:
Evenement rare
Evenement connu



Régression
Erreur balistique
Erreur d'empilement
compression

Remplacement du TNS classique (plus besoin de shaper)

FPGA: L'architecture générale / ideale



D_i

Extraction des caractéristiques

N entrées

Apprentissage profond: ANN, RNN, CNN…

P sorties

Formatage humain : Vecteur, matrice, images

IHM Analyses humaine

LOI 2024

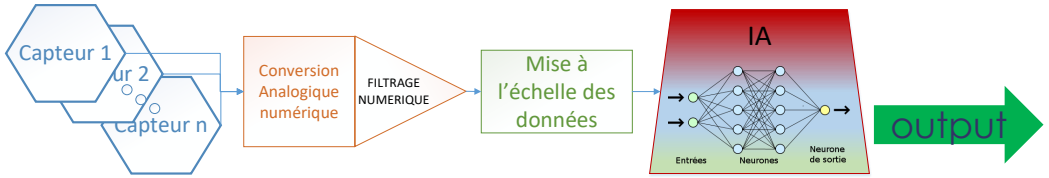
News: IA pour SE fiabilité

Explicabilité du reseau

Compression & stockage

Mémoires

FPGA: L'architecture generale / ideale



D_i

Extraction des caractéristiques

N entrées

Apprentissage profond: ANN, RNN, CNN…

P sorties

Formatage humain : Vecteur, matrice, images

IHM Analyses humaine

Explicabilité du reseau

µPs

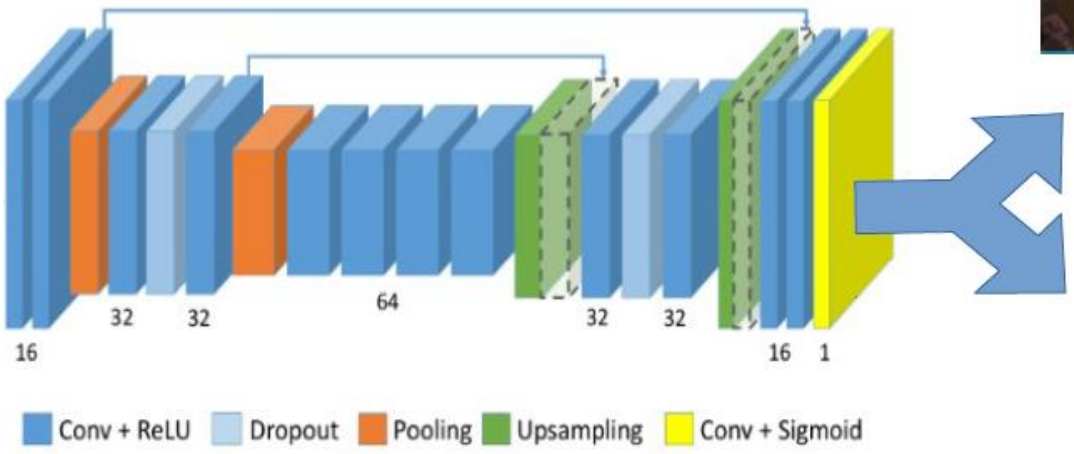
FreeRTOS Linux…

Compression & stockage

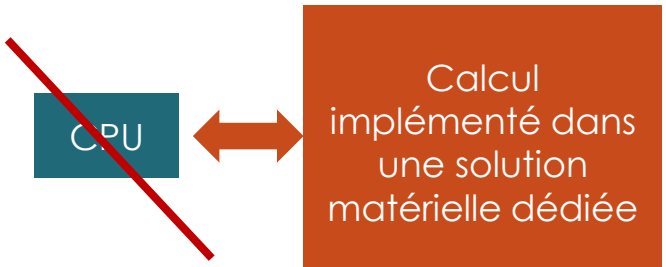
Mémoires

Notre objectif

Implémenter les différents types de RN (DNN, CNN, RNN) dans des SE



IA embarqué : 2 solutions

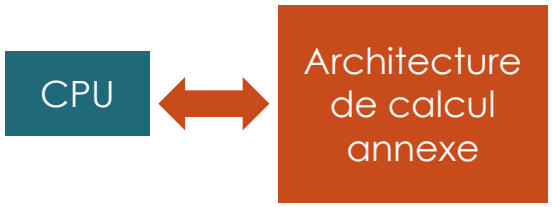


Spatial Accelerators

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

ASIC
Circuit neuromorphique
FPGA

2 En cours de développement



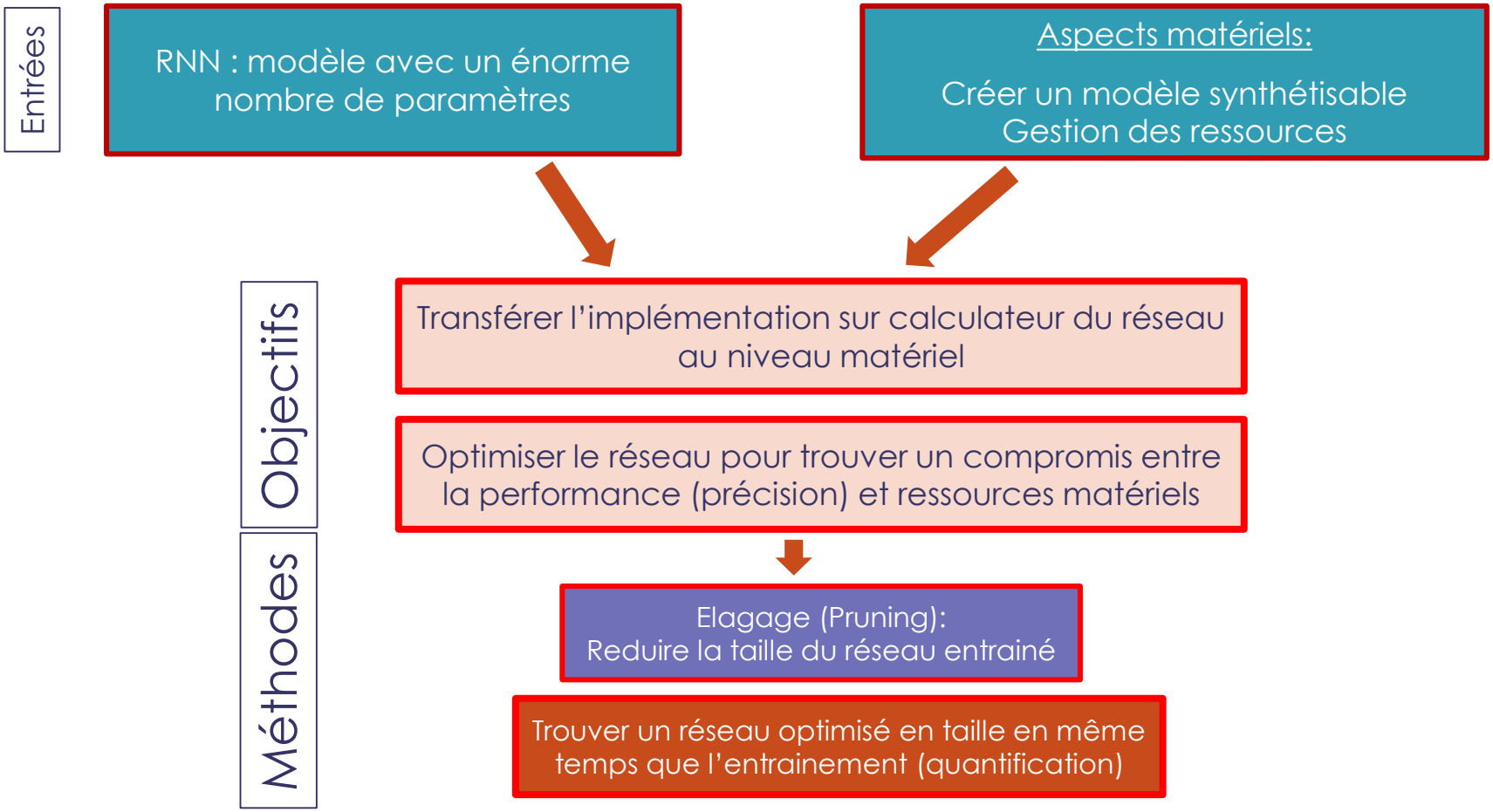
Edge computing

-- Reste basé sur de la programmation logicielle

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

1 Pilote les développements actuels

L'approche embarquée : Une question d'optimisation



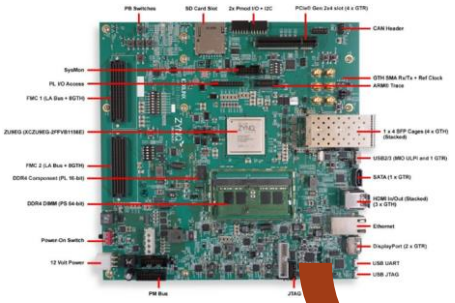
Outils sous Linux

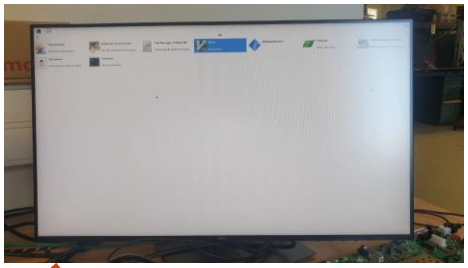
Télécharger le bloc Docker associé aux outils Tensorflow/Keras/Vitis-AI:

- **https://hub.docker.com/r/xilinx/vitis-ai**
  - **docker pull xilinx/vitis-ai**

- Extraire les données (qq 10 milliers)
  - Données d'entrainements
  - Données de validation
  - Données de test
- Choix et création de l'architecture du RN
- Apprentissage du réseau (entrainement)
- Optimisation du réseau
- Conversion et gel du réseau
- Quantification des coefficients
- Compilation du RN pour le DPU
- Execution du réseau sur le DPU (ZCU102/ZCU104/Versal)

Principe de Vitis AI sur ZCU102





PetaLinux

- TEST DE RESNET50 SUR DES IMAGES
- TEST DE YOLO IMAGES ET VIDEO

Démonstration avec le “model Zoo” : Resnet50

Avion 91%

top[0] prob = 0.912573 name = **airliner**
top[1] prob = 0.074909 name = wing
top[2] prob = 0.010138 name = warplane,military plane
top[3] prob = 0.002262 name = space shuttle
top[4] prob = 0.000053 name = airship, dirigible

Corail cerveau 98%

top[0] prob = 0.980779 name = **brain coral**
top[1] prob = 0.008485 name = coral reef
top[2] prob = 0.008485 name = jackfruit, jak, jack
top[3] prob = 0.000542 name = sea urchin
top[4] prob = 0.000422 name = puffer, pufferfish, blowfish

oseille 39% / vache 31%

top[0] prob = 0.398545 name = **sorrel**
top[1] prob = 0.310387 name = **ox**
top[2] prob = 0.114185 name = redbone
top[3] prob = 0.042006 name = vizsla, Hungarian pointer
top[4] prob = 0.032715 name = worm fence, snake fence

Démonstration avec le “model Zoo” : Resnet50

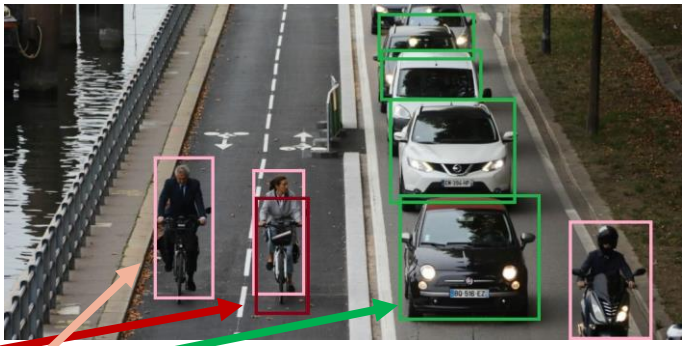
top[0] prob = 0.951893 name = **teddy, teddy bear** top[1] prob = 0.010575 name = Chihuahua top[2] prob = 0.006414 name = Airedale top[3] prob = 0.003890 name = corboy hat, ten-gallon hat top[4] prob = 0.002360 name = toy poodle	

➔ **main.cc** de **Resnet50** :

- **ListImages()** : fonction qui charge les images que l’on veut traiter.
- **LoadWords()** : fonction qui charge les différentes catégories que connait Resnet.
- **CPUCalcSoftmax()** : fonction qui calcule la fonction mathématique Softmax.
- **TopK()** : fonction qui renvoie le top K (dans notre exemple K=5) en terme de probabilité pour une image.
- **runResnet50()** : fonction qui fait le lien entre toutes les fonctions précédentes. Elle commence par charger

```
auto job_id = runner->execute_async(inputsPtr, outputsPtr);
runner->wait(job_id.first, -1);
for (unsigned int i = 0; i < runSize; i++) {
cout << "\nImage : " << images[n + i] << endl;
/* Calculate softmax on CPU and display TOP-5 classification
results */ CPUCalcSoftmax(&FCResult[i * outSize], outSize, softmax);
TopK(softmax, outSize, 5, kinds);
/* Display the impage */
cv::imshow("Classification of ResNet50", imageList[i]);
cv::waitKey(10000);
```

YOLO : traitement multicible sur images

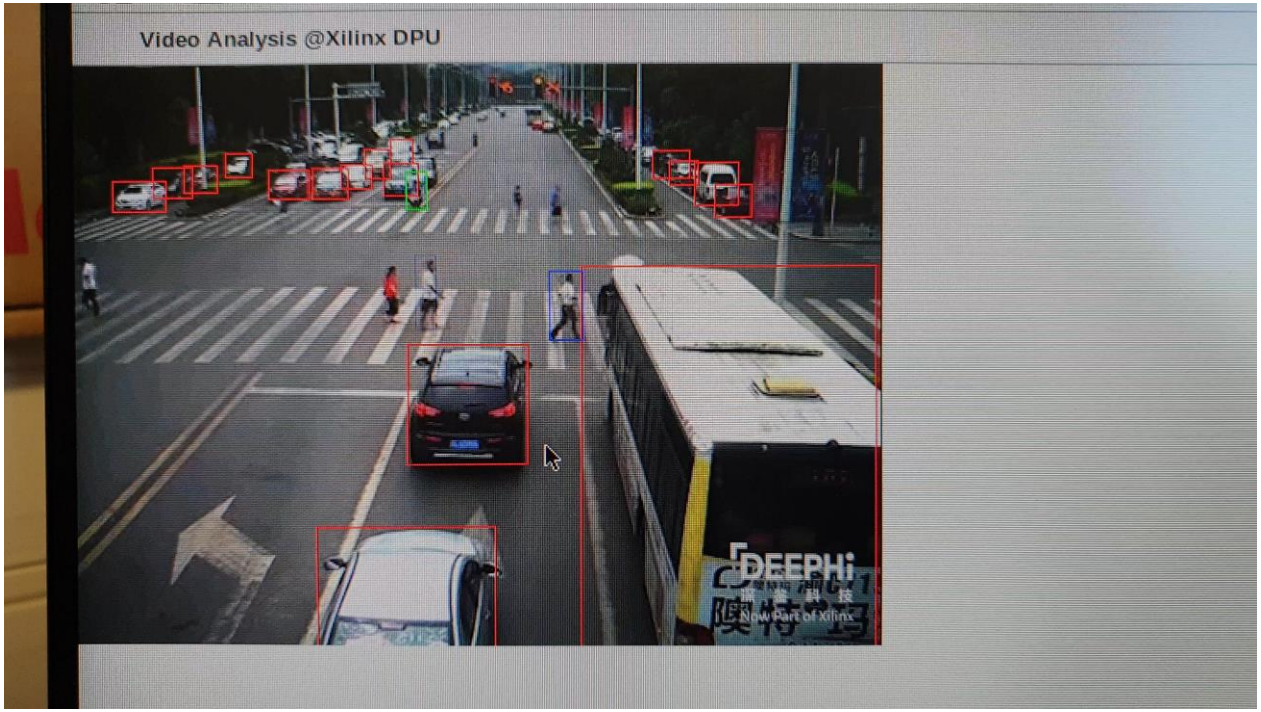


RESULT : 1	395.669 [illegible] 471.263 483.167	0.779274
RESULT : 6	[illegible] 734.577 84.4693	0.979318
RESULT : 6	604.624 151.448 797.66 310.705	0.968632
RESULT : 6	618.412 299.663 837.151 491.764	0.928414
RESULT : 6	589.709 72.6563 747.69 149.985	0.607446
RESULT : 14	236.443 240.04 327.627 457.719	0.998934
RESULT : 14	390.115 260.296 470.584 452.397	0.99409
RESULT : 14	885.904 341.999 1010.54 522.461	0.989938

YOLO : traitement multicible sur video

Traitement vidéo en ligne sur ZCU102

YOLO : traitement multicible sur video

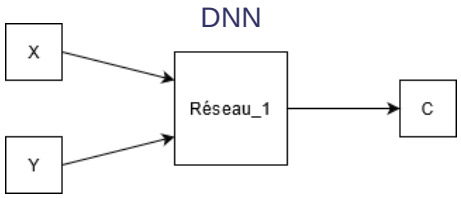


Traitement vidéo en ligne sur ZCU102

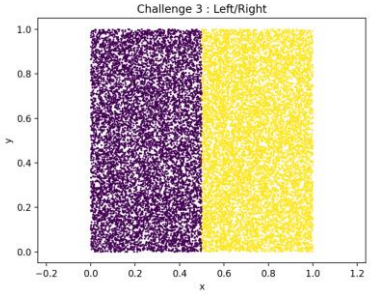
Zynq XcZ7020 & DNN: TF, Keras, QKeras, HLS4ML

XcZ7020: **70€ - 150€**

- Dual-core ARM Cortex-9
- Maximum frequency 667MHz-867MHz
- 2x AXI Master & 2 AXI Slave 32 bits
- 53,2 k LUT
- 220 DSP
- 106,4 k Flip-flop



2x200x100x50x1



Raw implementation <32,11>

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 9.883 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+----------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline |
    |   min   |   max   |    min   |    max   | min | max |   Type   |
    +---------+---------+----------+----------+-----+-----+----------+
    |      382|      386| 3.775 us | 3.815 us |  100|  100| dataflow |
    +---------+---------+----------+----------+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+-------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT  | URAM|
+-----------------+---------+-------+--------+-------+-----+
|DSP              |        -|      -|       -|      -|    -|
|Expression       |        -|      -|       0|   2858|    -|
|FIFO             |        0|      -|    3515|  30732|    -|
|Instance         |      229|   1006|  133648|  95258|    -|
|Memory           |        -|      -|       -|      -|    -|
|Multiplexer      |        -|      -|       -|   6336|    -|
|Register         |        -|      -|     706|      -|    -|
+-----------------+---------+-------+--------+-------+-----+
|Total            |      229|   1006|  137869| 135184|    0|
+-----------------+---------+-------+--------+-------+-----+
|Available        |      280|    220|  106400|  53200|    0|
+-----------------+---------+-------+--------+-------+-----+
|Utilization (%)  |       81|    457|     129|    254|    0|
+-----------------+---------+-------+--------+-------+-----+
```

Model 1

Code HLS generé avec HLS4ML

Model: Keras,
No Pruning:
tf.keras.optimizers,
Quantification:
HLS4ML

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 9.143 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+----------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline |
    |   min   |   max   |    min   |    max   | min | max |   Type   |
    +---------+---------+----------+----------+-----+-----+----------+
    |      374|      378| 3.419 us | 3.456 us |  100|  100| dataflow |
    +---------+---------+----------+----------+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+-------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT  | URAM|
+-----------------+---------+-------+--------+-------+-----+
|DSP              |        -|      -|       -|      -|    -|
|Expression       |        -|      -|       0|   2858|    -|
|FIFO             |        0|      -|    3515|  18114|    -|
|Instance         |      101|    255|   36726|  82229|    -|
|Memory           |        -|      -|       -|      -|    -|
|Multiplexer      |        -|      -|       -|   6336|    -|
|Register         |        -|      -|     706|      -|    -|
+-----------------+---------+-------+--------+-------+-----+
|Total            |      101|    255|   40947| 109537|    0|
+-----------------+---------+-------+--------+-------+-----+
|Available        |      280|    220|  106400|  53200|    0|
+-----------------+---------+-------+--------+-------+-----+
|Utilization (%)  |       36|    115|      38|    205|    0|
+-----------------+---------+-------+--------+-------+-----+
```

Model:
Quantification:
QKeras,
No Pruning:
tf.keras.optimizers,

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 9.143 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+----------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline |
    |   min   |   max   |    min   |    max   | min | max |   Type   |
    +---------+---------+----------+----------+-----+-----+----------+
    |      373|      377| 3.410 us | 3.447 us |  100|  100| dataflow |
    +---------+---------+----------+----------+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+-------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT  | URAM|
+-----------------+---------+-------+--------+-------+-----+
|DSP              |        -|      -|       -|      -|    -|
|Expression       |        -|      -|       0|   2850|    -|
|FIFO             |        0|      -|    3505|  18054|    -|
|Instance         |      100|    253|   36030|  81756|    -|
|Memory           |        -|      -|       -|      -|    -|
|Multiplexer      |        -|      -|       -|   6318|    -|
|Register         |        -|      -|     704|      -|    -|
+-----------------+---------+-------+--------+-------+-----+
|Total            |      100|    253|   40239| 108978|    0|
+-----------------+---------+-------+--------+-------+-----+
|Available        |      280|    220|  106400|  53200|    0|
+-----------------+---------+-------+--------+-------+-----+
|Utilization (%)  |       35|    115|      37|    204|    0|
+-----------------+---------+-------+--------+-------+-----+
```

Model 2

Model: Keras,
Pruning: tf.keras.optimizers,
Quantification: HLS4ML
Optim for resources

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 9.143 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+-----------+-----------+------+------+----------+
    |  Latency (cycles) |   Latency (absolute)  |   Interval  | Pipeline |
    |   min   |   max   |    min    |    max    |  min |  max |   Type   |
    +---------+---------+-----------+-----------+------+------+----------+
    |     2476|     2480| 22.638 us | 22.675 us |  1000|  1000| dataflow |
    +---------+---------+-----------+-----------+------+------+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+-------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT  | URAM|
+-----------------+---------+-------+--------+-------+-----+
|DSP              |        -|      -|       -|      -|    -|
|Expression       |        -|      -|       0|   2858|    -|
|FIFO             |        0|      -|    3515|  18314|    -|
|Instance         |       25|     27|   29254|  48722|    -|
|Memory           |        -|      -|       -|      -|    -|
|Multiplexer      |        -|      -|       -|   6336|    -|
|Register         |        -|      -|     706|      -|    -|
+-----------------+---------+-------+--------+-------+-----+
|Total            |       25|     27|   33475|  76230|    0|
+-----------------+---------+-------+--------+-------+-----+
|Available        |      280|    220|  106400|  53200|    0|
+-----------------+---------+-------+--------+-------+-----+
|Utilization (%)  |        8|     12|      31|    143|    0|
+-----------------+---------+-------+--------+-------+-----+
```

→ Code HLS generé avec HLS4ML

Model 3 - #PRAGMA ARRAY_PARTITION dim=3

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 9.062 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+----------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline |
    |   min   |   max   |    min   |    max   | min | max |   Type   |
    +---------+---------+----------+----------+-----+-----+----------+
    |      199|      203| 1.803 us | 1.840 us |   50|   50| dataflow |
    +---------+---------+----------+----------+-----+-----+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+--------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |   LUT  | URAM|
+-----------------+---------+-------+--------+--------+-----+
|DSP              |        -|      -|       -|       -|    -|
|Expression       |        -|      -|       0|    2858|    -|
|FIFO             |        0|      -|    3515|   14157|    -|
|Instance         |      171|     10|   35019|  207910|    -|
|Memory           |        -|      -|       -|       -|    -|
|Multiplexer      |        -|      -|       -|    6336|    -|
|Register         |        -|      -|     706|       -|    -|
+-----------------+---------+-------+--------+--------+-----+
|Total            |      171|     10|   39240|  231261|    0|
+-----------------+---------+-------+--------+--------+-----+
|Available        |      280|    220|  106400|   53200|    0|
+-----------------+---------+-------+--------+--------+-----+
|Utilization (%)  |       61|      4|      36|     434|    0|
+-----------------+---------+-------+--------+--------+-----+
```

Model 4: REMOVE ALL #PRAGMA ARRAY_PARTITION in HLS code

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  | 5.00 ns | 7.144 ns |   0.62 ns  |
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-------+-------+----------+
    |  Latency (cycles) |  Latency (absolute) |    Interval   | Pipeline |
    |   min   |   max   |    min   |    max   |  min  |  max  |   Type   |
    +---------+---------+----------+----------+-------+-------+----------+
    |    91251|    91255| 0.652 ms | 0.652 ms |  50000|  50000| dataflow |
    +---------+---------+----------+----------+-------+-------+----------+
```

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+-------+-----+
|       Name      | BRAM_18K| DSP48E|   FF   |  LUT  | URAM|
+-----------------+---------+-------+--------+-------+-----+
|DSP              |        -|      -|       -|      -|    -|
|Expression       |        -|      -|       0|    238|    -|
|FIFO             |        0|      -|     255|   1326|    -|
|Instance         |       25|     27|   33689|  36552|    -|
|Memory           |        5|      -|       0|      0|    0|
|Multiplexer      |        -|      -|       -|    486|    -|
|Register         |        -|      -|      56|      -|    -|
+-----------------+---------+-------+--------+-------+-----+
|Total            |       30|     27|   34000|  38602|    0|
+-----------------+---------+-------+--------+-------+-----+
|Available        |      280|    220|  106400|  53200|    0|
+-----------------+---------+-------+--------+-------+-----+
|Utilization (%)  |       10|     12|      31|     72|    0|
+-----------------+---------+-------+--------+-------+-----+
```

At one point, we need to optimize HLS code
At one point, we could develop directively in VHDL

Discussion autour des FPGA: Spatial Accelerators

Une question de budget (nombre de voies DAQ)

- **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%

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

€ ⟷ €€€

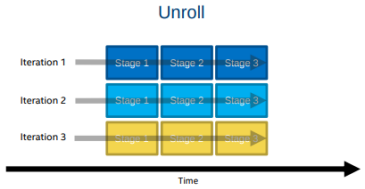
- **Intel Arria 10 + Intel HLS (expert HLS) : €€**

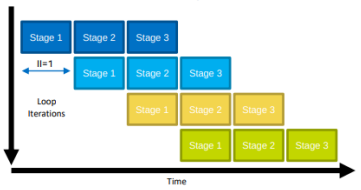
	ALUTs	FFs	RAMs	MLABs	DSPs
Ch1 vanilla	602 (0%)	547 (0%)	4 (0%)	2 (0%)	1.5 (0%)
Ch1 pipeline	610 (0%)	624 (0%)	4 (0%)	5 (0%)	1.5 (0%)
Ch1 unroll	515 (0%)	245 (0%)	4 (0%)	1 (0%)	0 (0%)
Ch1 u+p	515 (0%)	245 (0%)	4 (0%)	1 (0%)	0 (0%)

	ALUTs	FFs	RAMs	MLABs	DSPs
Ch3 vanilla	1 408 (0%)	1 809 (0%)	30 (1%)	12 (0%)	2.5 (0%)
Ch3 pipeline	2 093 (0%)	4 460 (0%)	32 (1%)	48 (0%)	2.5 (0%)
Ch3 unroll	118 041 (14%)	36 737 (2%)	5 (0%)	37 (0%)	0 (0%)
Ch3 u+p	27 524 (3%)	42 546 (2%)	1 855 (68%)	298 (1%)	75 (5%)

	ALUTs	FFs	RAMs	MLABs	DSPs
Ch4 vanilla	4 442 (0%)	6 415 (0%)	355 (1%)	20 (0%)	3.5 (0%)
Ch4 pipeline	5 555 (0%)	10 899 (0%)	362 (1%)	113 (0%)	3.5 (0%)
Ch4 unroll	Problème d'implémentation				
Ch4 u+p					

	ALUTs	FFs	RAMs	MLABs	DSPs
Ch5 vanilla	1 669 (0%)	2 193 (0%)	32 (1%)	16 (0%)	2.5 (0%)
Ch5 pipeline	2 617 (0%)	6 074 (0%)	35 (1%)	76 (0%)	2.5 (0%)
Ch5 unroll	569 259 (67%)	196 051 (11%)	6 (0%)	40 (0%)	0 (0%)
Ch5 u+p	17 560 (2%)	23 244 (1%)	507 (19%)	237 (1%)	50.5 (3%)

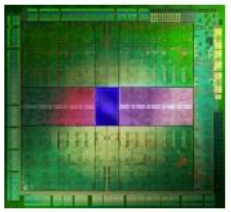




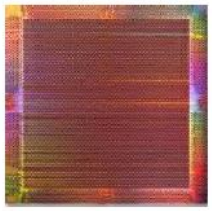
Question of HLS optimisation

Exemples Technologiques

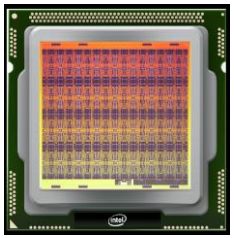
CPUs
MPPA



GPUs



FPGAs

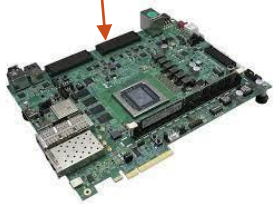


NMC

nVidia

DE10-Agil

ZCU102,
ZCU104



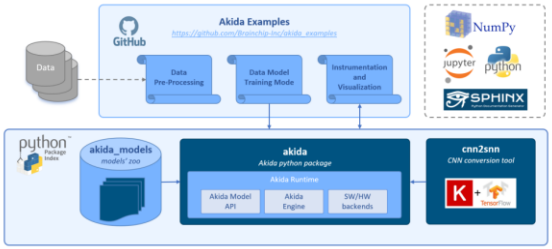
VCK190

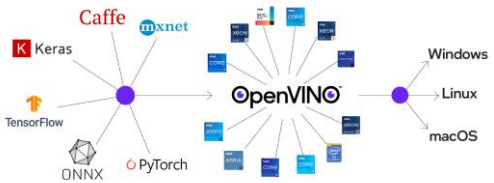
Video Sources

INPUT DATA
- Camera
- Images
- Lidar

Output /Display

RESULTS
- Classification
- Segmentation





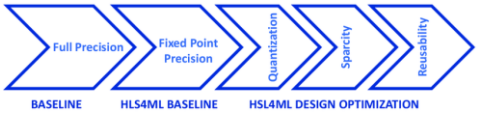
C, C++, SystemC,
OpenCL API C

Vivado™ HLS

VHDL or Verilog

System IP Integration

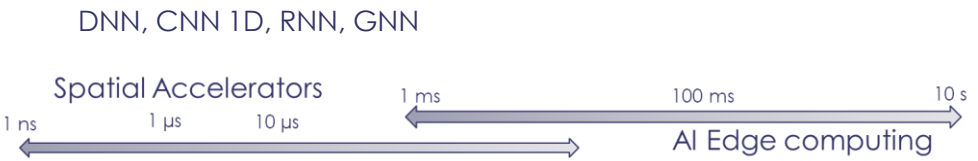
hls 4 ml

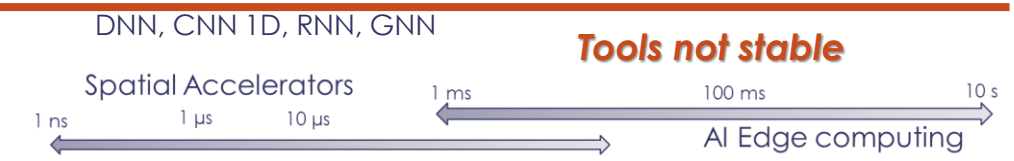


Features with others components

Microcontrollers
AI edge computing
RISC-V Architecture

Type	Company	Components	Dev Boards	Pictures	Soft Tools
AI Edge computing	AMD Xilinx	Zynq	ARTY Z7 PYNQ ZCU104 SCU102		VITIS-AI
Spatial acceleration				 	Tensorflow/Keras + HLS4ML PyTorch + FiNN+Brevitas
AI Edge computing	AMD Xilinx	VERSAL	VK290		HLS+VITIS-AI
AI Edge computing	nVidia	Jetson	Nano TX2 Xavier NX AGX Xavier Orin	 	Jetpack SDK + DeepStream SDK + TensorRT
Spatial acceleration	Intel	AGILEX ARRIA	DE10-AGILEX ARRIA 10 GX		Intel HLS
AI Edge computing					FPGA AI Suite OpenVino
Spatial acceleration	Brainchip	AKD1000	Akida PCIe		Akida MetaTF ML framework
AI Edge computing	SiPEED	Kendryte210 RISC-V	MAIXDUiNO		SiPEED SDK + micropython
AI Edge computing	ST Microelectronic	STM32	Nucleo64F411		CubeMX + Cube.AI
AI Edge computing	Analog Devices	MAX78000	MAX78000EVKIT		Maxim Micros SDK





Chaque composant a son logiciel d'IA !