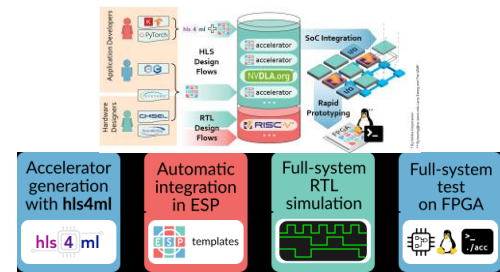
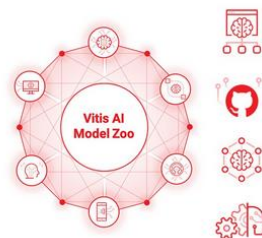
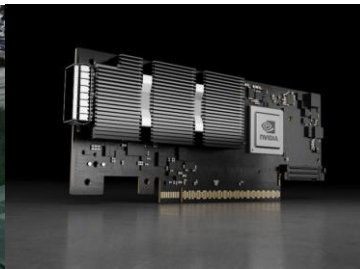
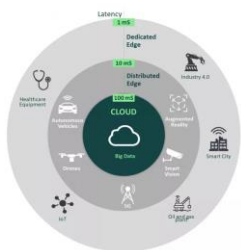


Principes et vocabulaire de l'IA moderne



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. »

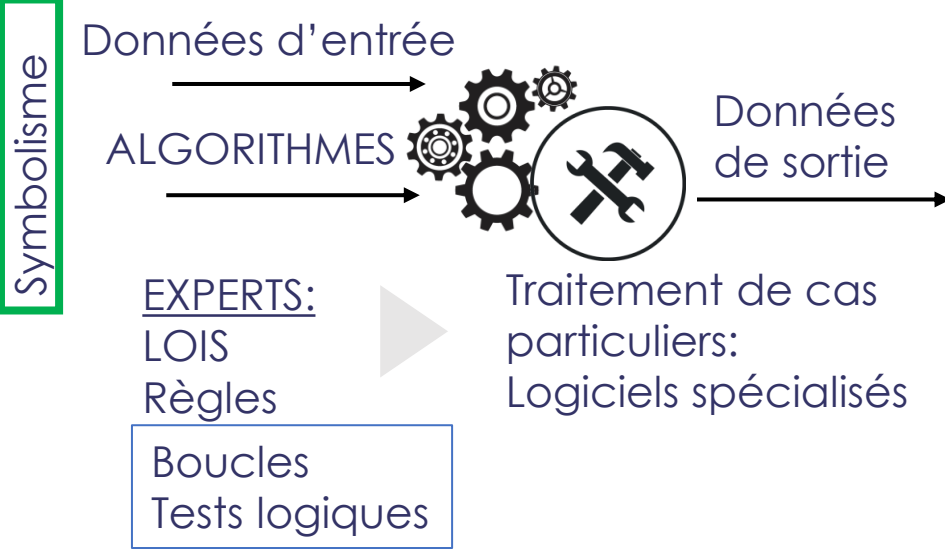
Pour y parvenir, trois composants sont nécessaires :

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

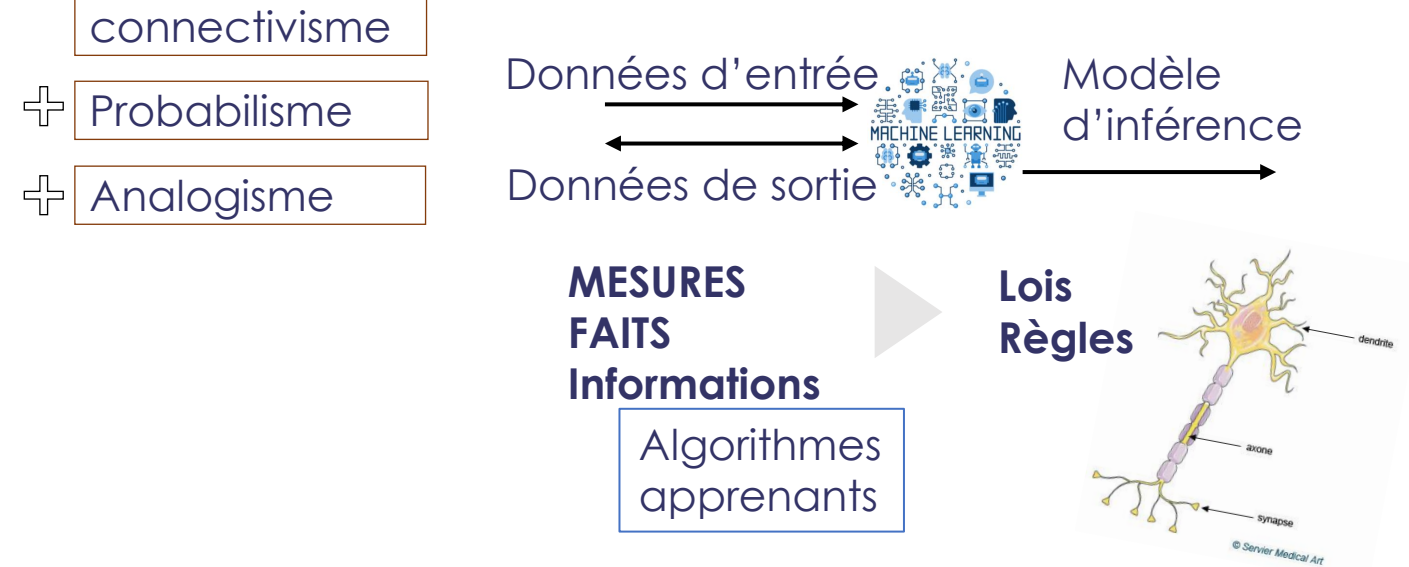
Pour se rapprocher le plus possible du comportement humain, l'intelligence artificielle a besoin d'une quantité de données et d'une capacité de traitement élevées. »

 **Système embarqué**

Approche déductive

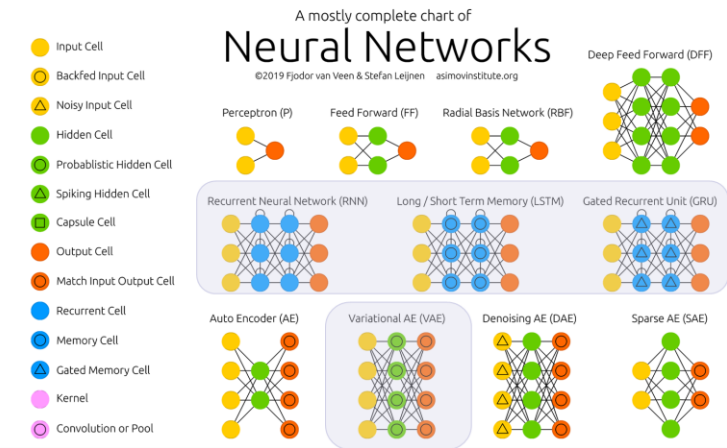
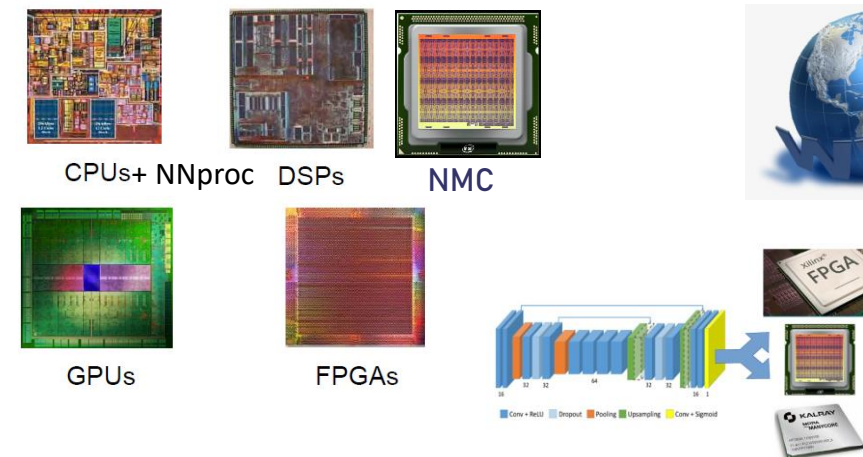


Approche inférentielle (Inductive)

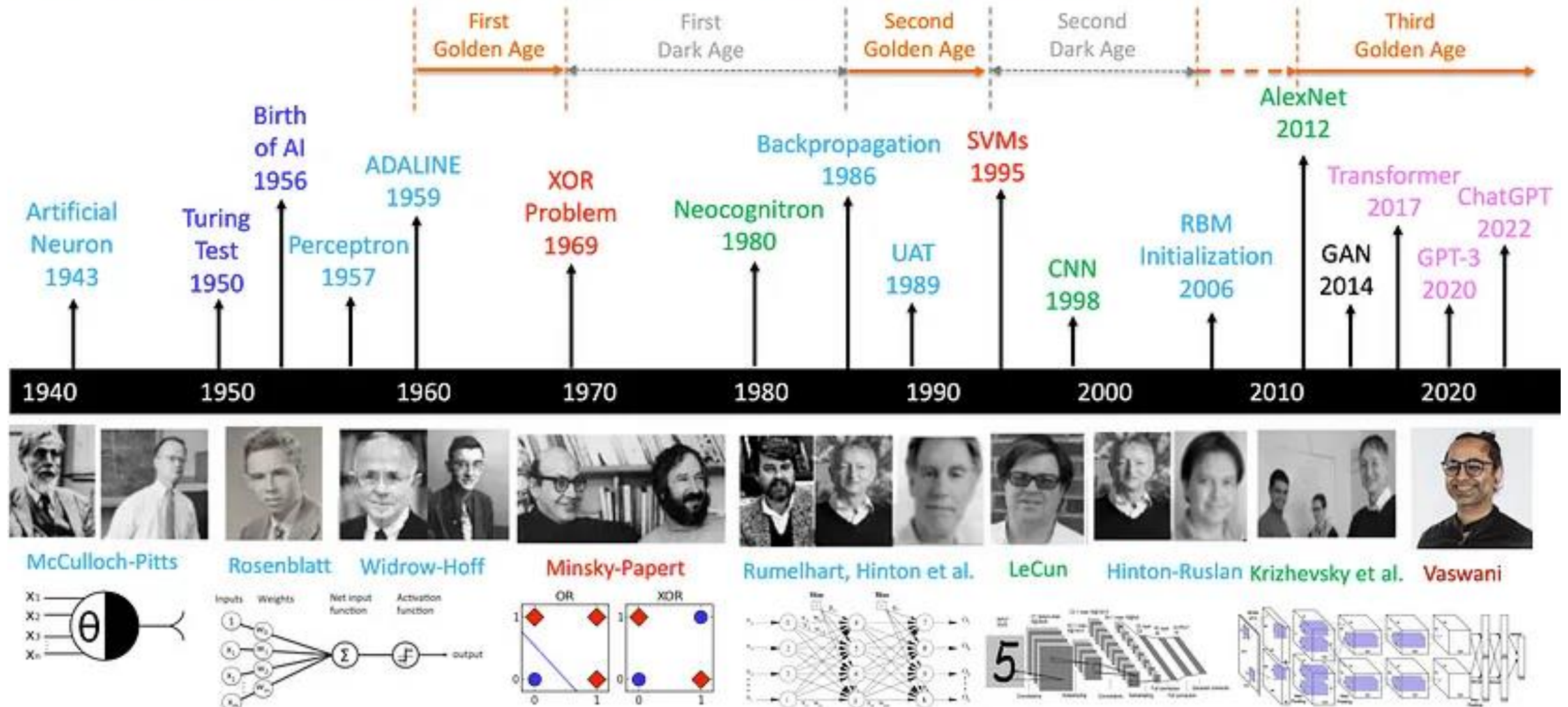


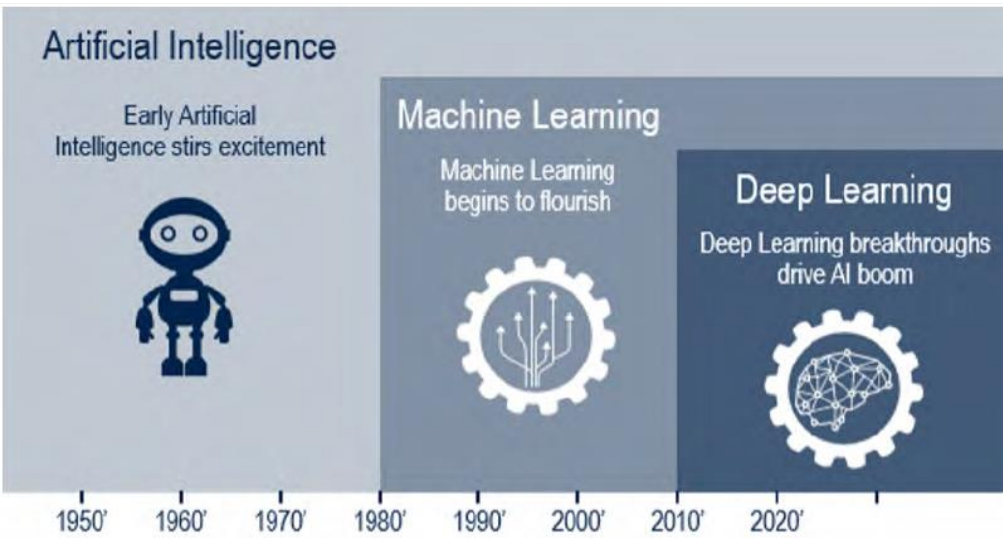
L'intelligence Artificielle dans les systèmes embarqués

Nouvelles architectures matérielles Outils de générations de données Réseaux de neurones et descente de gradient



A Brief History of AI with Deep Learning





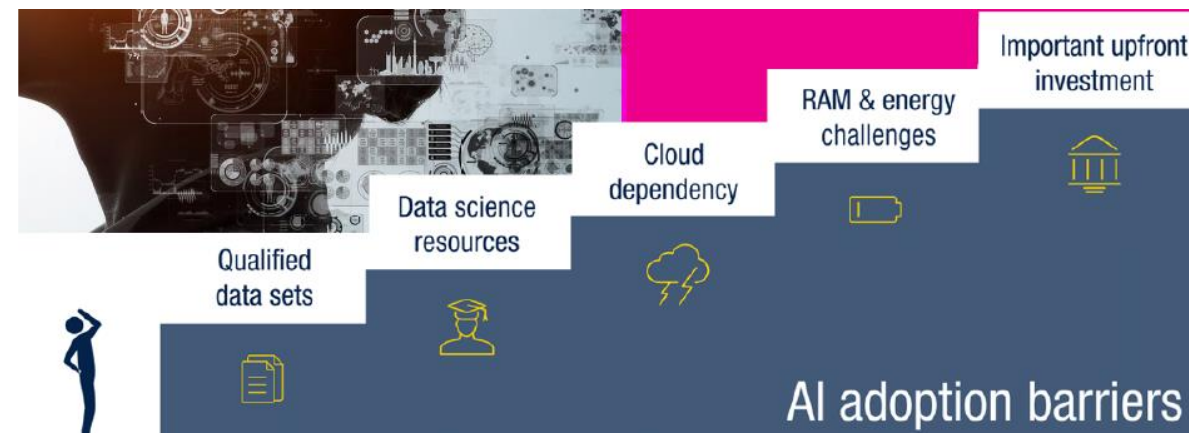
→ Evolution et progression de l'IA

→ Les barrières pour adopter l'IA dans les produits pour les sociétés.

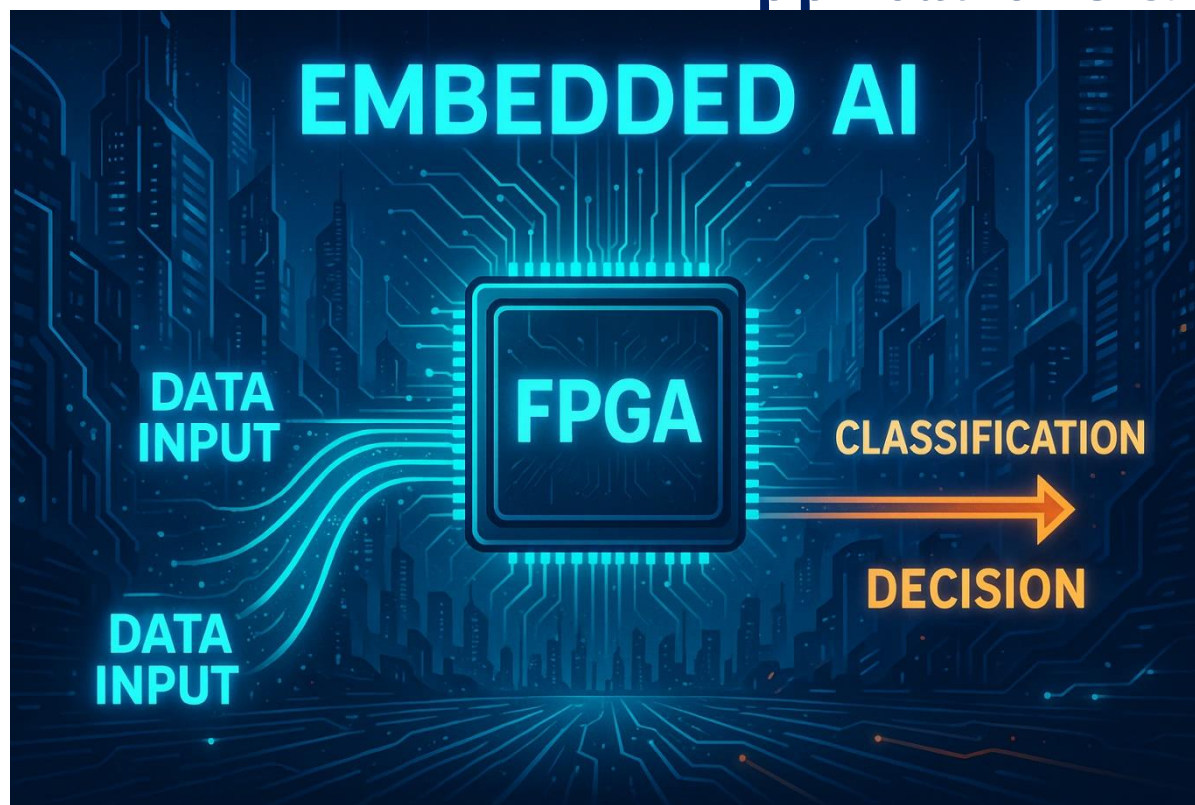
- investissement en RH, matériels, données
- Grandes compagnies

→ ML dans SE : solution économique

Explosion des projets et des solutions



– Principles – Material Aspects – Software Aspects –
– Applications & performances –



Test of Hardware Inferring Neural network

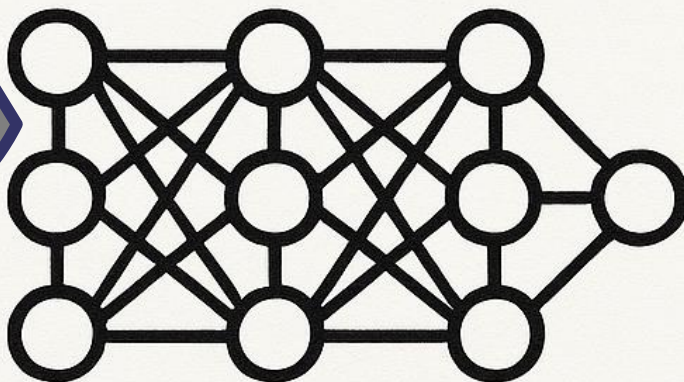


Reseau DAQ – Hardware-Firmware-Software-

Phase 2

DEEP LEARNING

Base de
connaissances

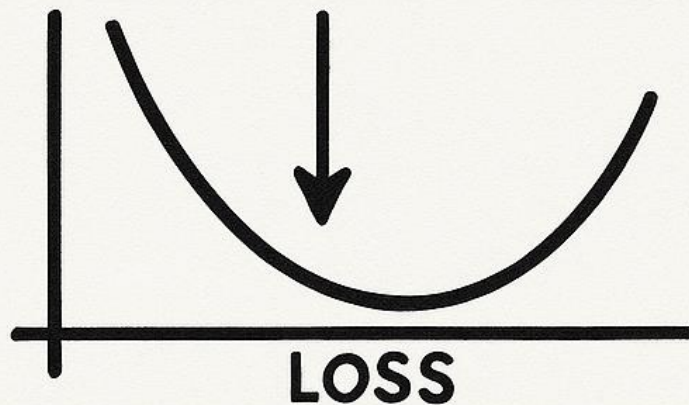


MODELS BASED ON
NEURAL NETWORKS

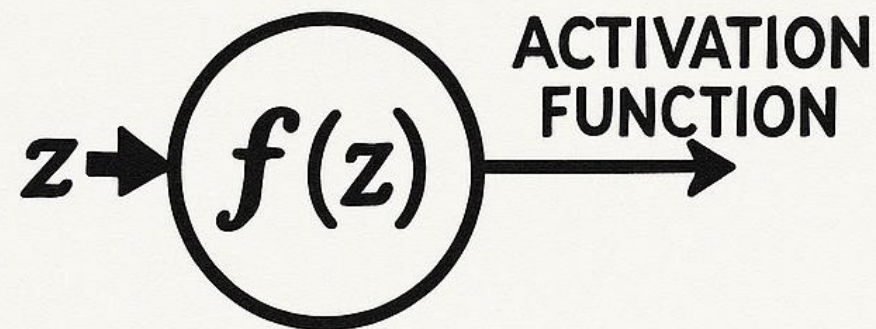
GRADIENT DESCENT ALGORITHM

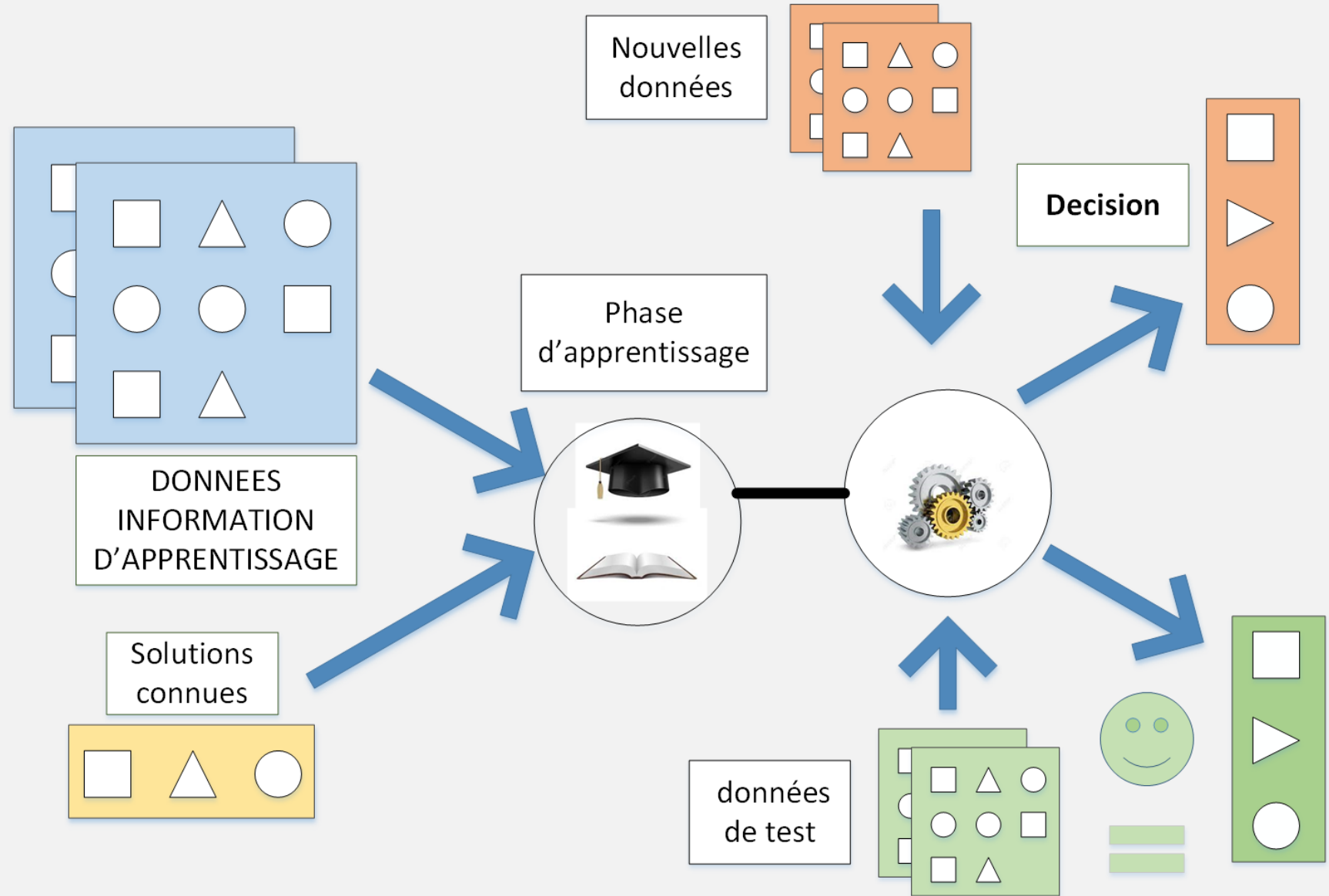


LOSS FUNCTION



NEURON





Les quatre catégories d'apprentissage profond

LLM

Fine-tuning

Supervised Learning

- Makes machine Learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Resolves classification and regression problems



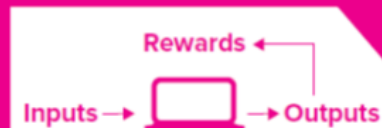
Unsupervised Learning

- Machines understand the data (Identifies patterns/structures)
- Evolution is qualitative or indirect
- Does not predict/find anything specific



Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning from +ve & -ve reinforcement
- Machine learns how to act in a certain environment
- To maximize rewards



Transfer learning

- Classical supervised techniques
- Change part of the model with new inputs and outputs
- Add features to the preliminary model



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*

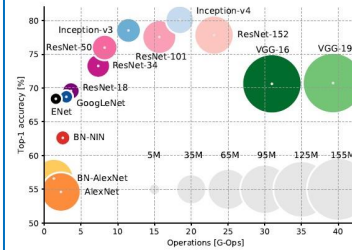


Tools

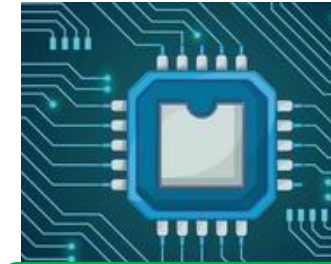
- **ML Tools:**
 - TF-KERAS, PyTorch ...
- **HLS4ML (Xilinx...)**
- **HLS**
- **Brevitas & FiNN(Xilinx)**
- **CONIFER (LLR)**
- **N2D2 (CEA)**
- **VHDL**
- ...



Artefacts & ML zoology



- **Model**
- **Code source...**



Digital hardware technologies

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



Deployment & Operational AI

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

- 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

- Séries temporelles (LLM, ChatGPT)
 - Prédiction des données dans une série 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

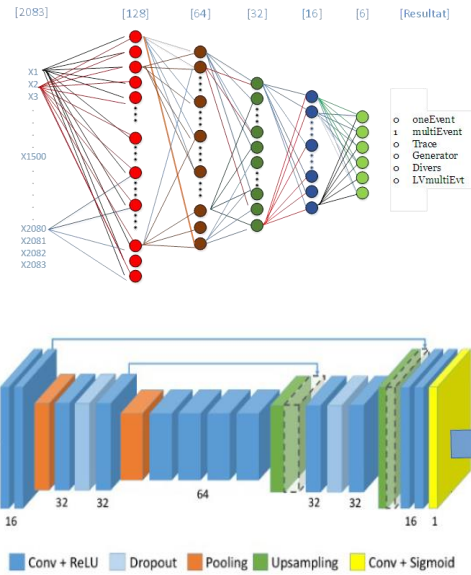


Imaginer comment se servir de ces techniques dans vos futurs domaines

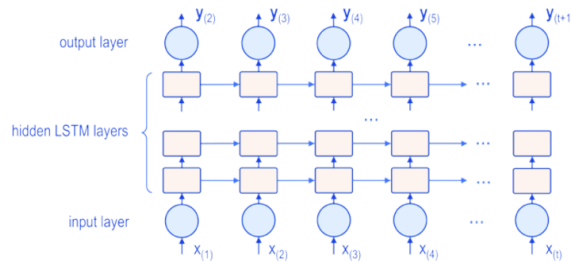


En Instrumentation, en IoT, avec détecteurs/capteurs

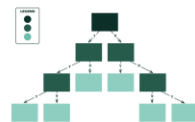
CONVOLUTIONAL DEEP NEURAL NETWORK



RECURRENT NEURAL NETWORK



DECISION TREE



RANDOM FOREST



Off-line

- Signal generation
- Design Optimisation

On-line

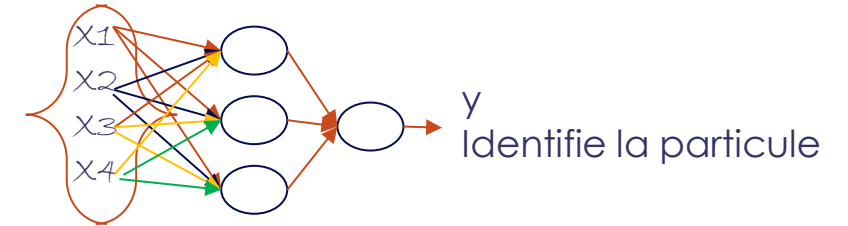
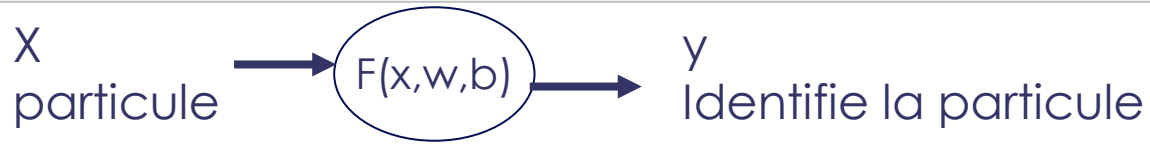
- Instrument Optimization
 - Signal recognition
 - Pile-Up recovery
 - Signal deconvolution
- Selection/ Classification/ Decision
 - Data selection
 - Data parameters prediction
 - Denoizing
- Data Compression
 - Reduced data format

L1

L2

L3

Apprentissage supervisé : Réseau de Neurones



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

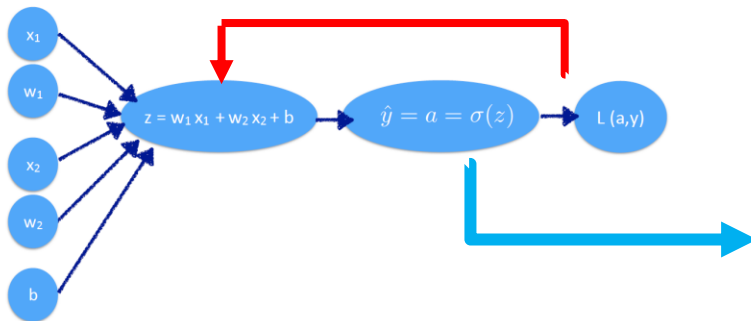
Puissance de calcul : CPU/GPU/FPGA....

Innovation dans les algorithmes (la descente du gradient)

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

Loss function:

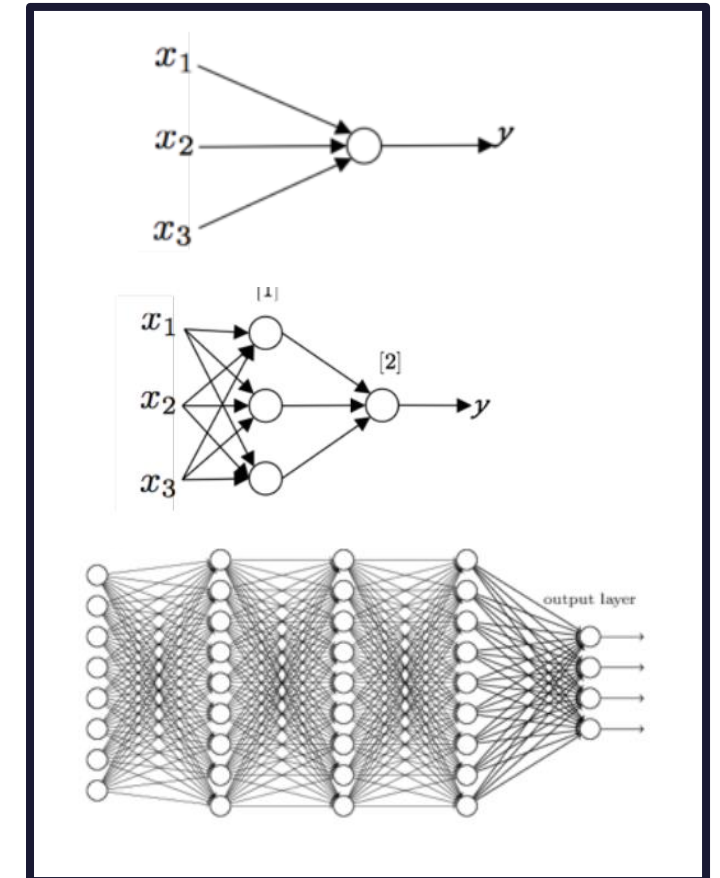
$$L(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$$



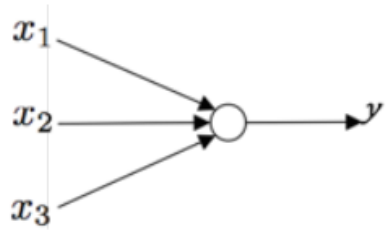
ACTIVATION FUNCTIONS

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		$g(z) = \max(\alpha z, z)$

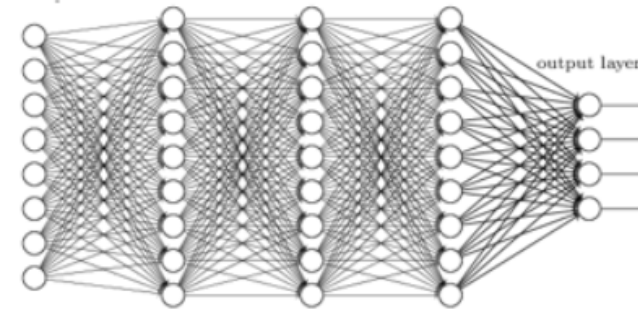
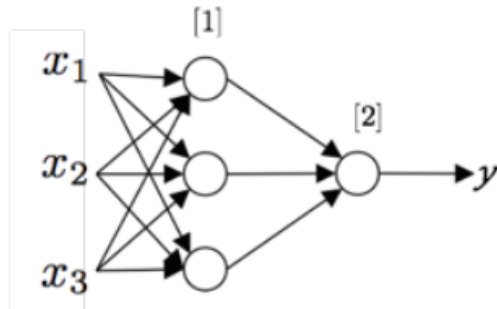
Using a linear activation function is useless un more than one layer networks.
It's rarely used



DEEP NEURAL NETWORKS



Shallow

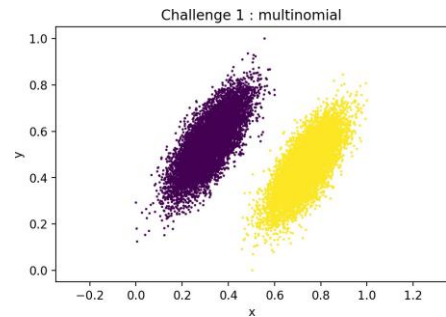
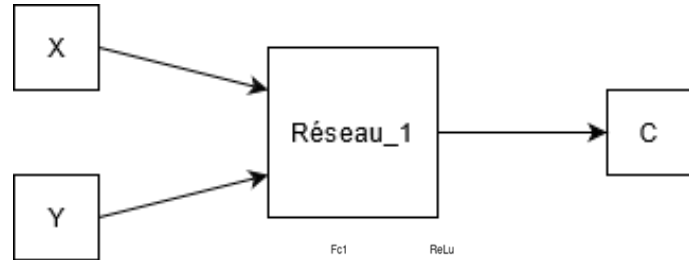


Deep Neural Network

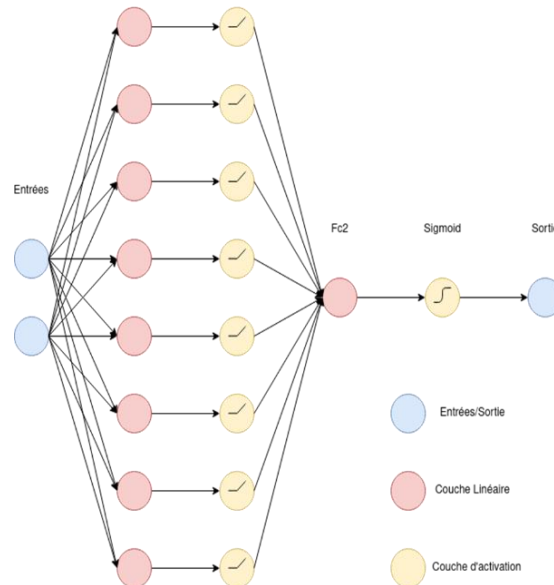
Need to define:

- 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



2x8x1

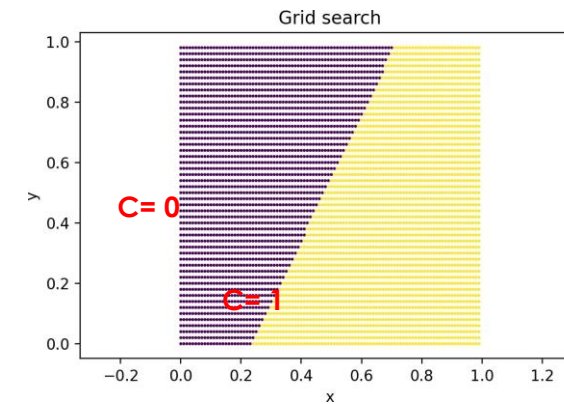


42 multiplications

Il faut calculer

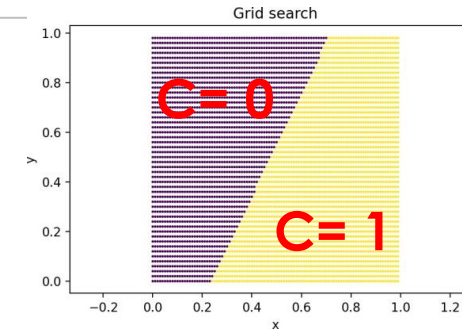
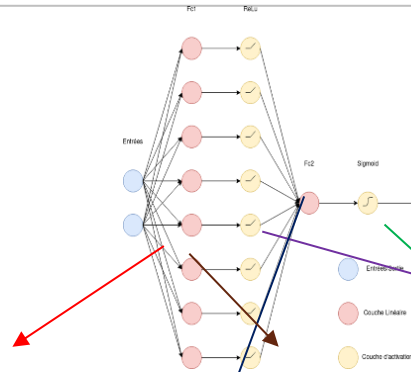
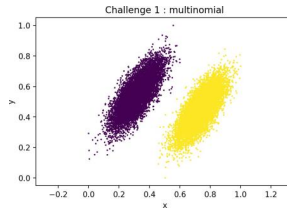


Principe: on fournit en entrée des coordonnées (x,y). Le réseau nous classe les coordonnées selon la région d'appartenance. Ici, c'est une simple régression logistique avec une séparation de deux région linéaire.



$$W = \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{pmatrix} \text{ et } B = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$
$$s = \sigma(\mathbf{W}X + \mathbf{B})$$

Un exemple de calcul: réseau de neurone peu profond



$X=0,2$
 $Y=0,7$

$$\begin{pmatrix} 1.7856 & -1.2345 \\ 2.4834 & -0.5797 \\ -0.3130 & 0.2658 \\ 0.3110 & -0.4116 \\ -0.4458 & 0.8010 \\ 3.4282 & -1.7166 \\ -0.6808 & 0.1631 \\ -0.5633 & 0.5694 \end{pmatrix} \times \begin{pmatrix} 0.2 \\ 0.7 \end{pmatrix} +$$

$$\begin{pmatrix} 0.0795 \\ -0.5202 \\ 0.4716 \\ -0.4349 \\ 1.3222 \\ -0.2305 \\ -0.3073 \\ 0.9498 \end{pmatrix} = \begin{pmatrix} -0.4275 \\ -0.4294 \\ 0.5951 \\ -0.6608 \\ 1.7938 \\ -0.7465 \\ -0.3293 \\ 1.2358 \end{pmatrix}$$

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

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

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

$X=0,2$
 $Y=0,7$

Appartient à la zone

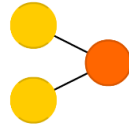
C=0

A mostly complete chart of Neural Networks

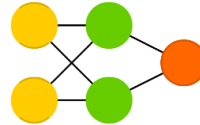
©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

-  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

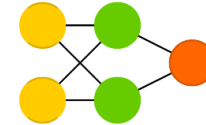
Perceptron (P)



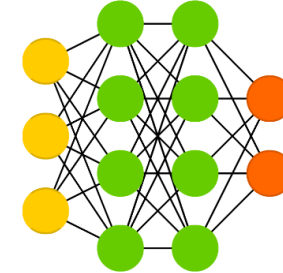
Feed Forward (FF)



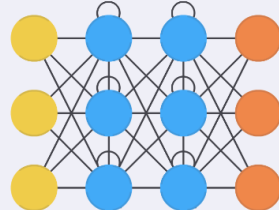
Radial Basis Network (RBF)



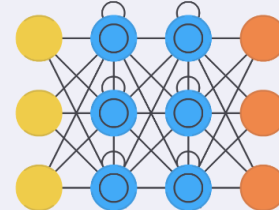
Deep Feed Forward (DFF)



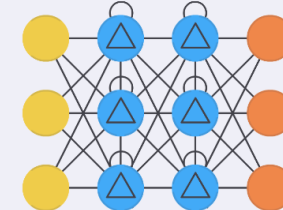
Recurrent Neural Network (RNN)



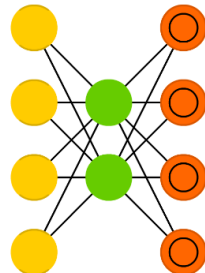
Long / Short Term Memory (LSTM)



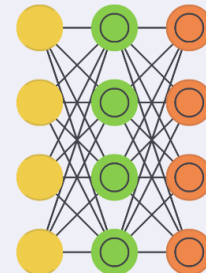
Gated Recurrent Unit (GRU)



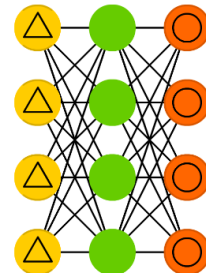
Auto Encoder (AE)



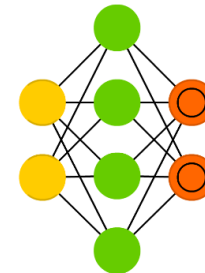
Variational AE (VAE)

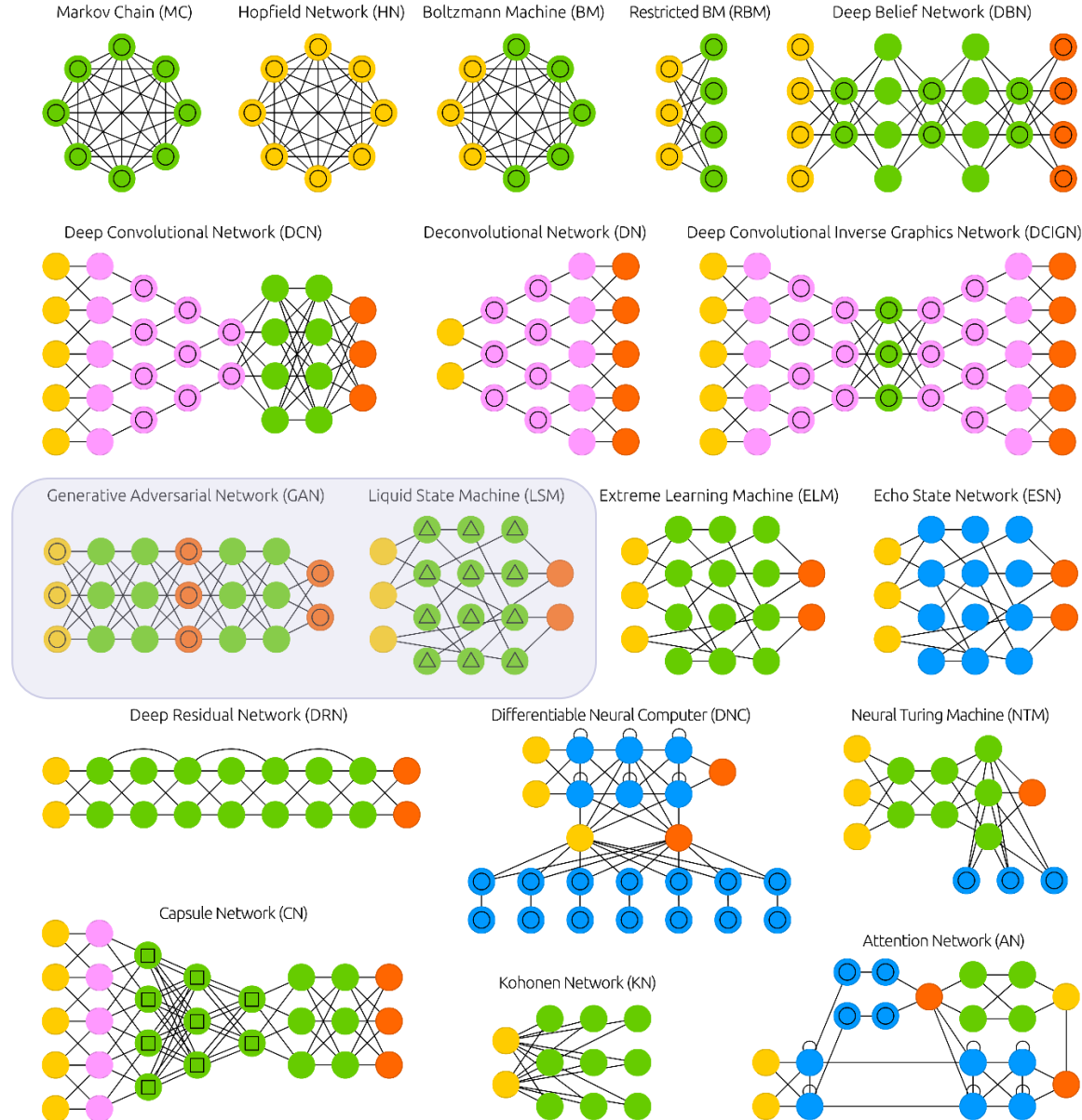


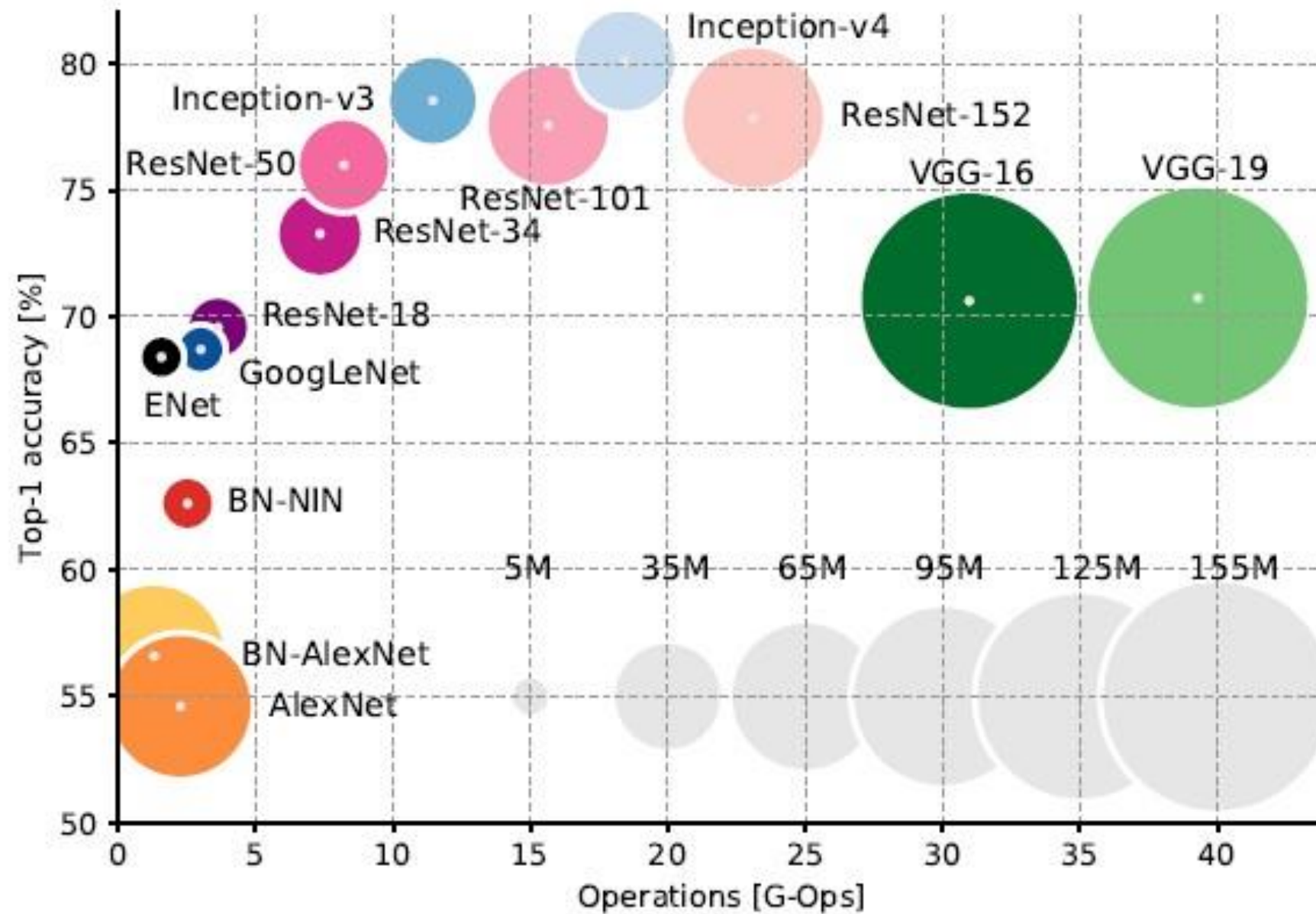
Denoising AE (DAE)



Sparse AE (SAE)

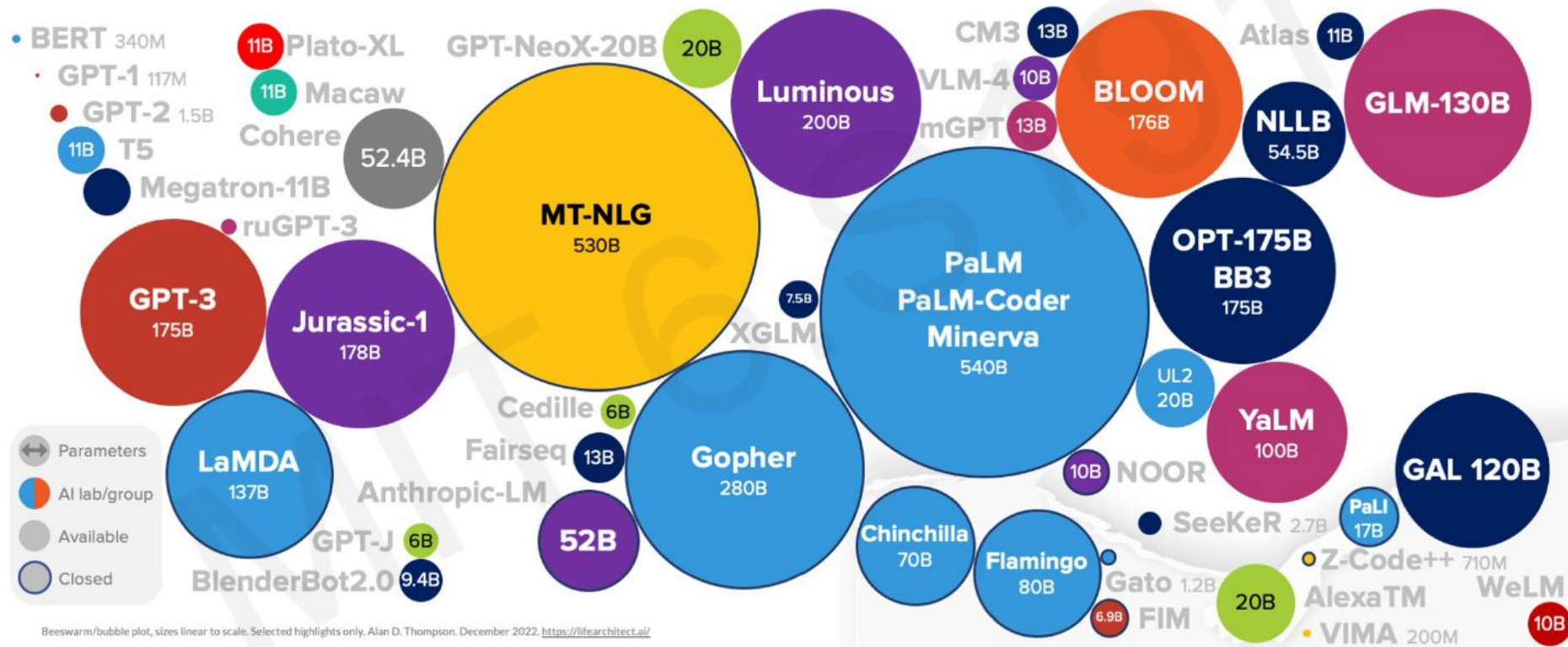






Modern Era of Statistics

Language Models size – up to Dec, 2022



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 représentation pour l'analyse statistique classique
Discrimination signals/bruit de fond
applications: CMS

CNN, ANN:

Monitoring 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

Accélérateurs: constructions

Aimants, monitoring d'anomalies,
Contrôle & commande



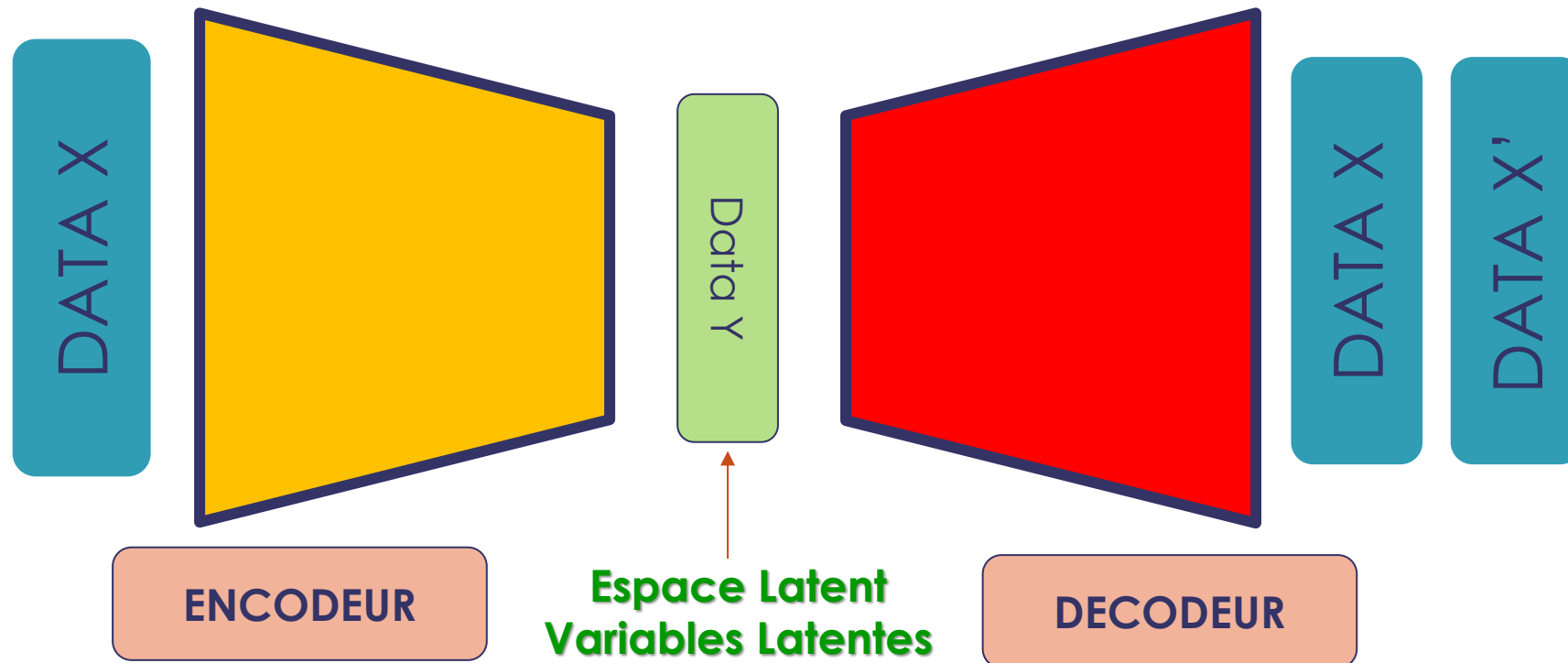
Et si l'IA avait de l'imagination

- Réaliser la labellisation des données est ennuyeux, couteux
- Comment faire pour automatiser cette phase de labellisation ?
- ➔ laisser le modèle découvrir les clusters !

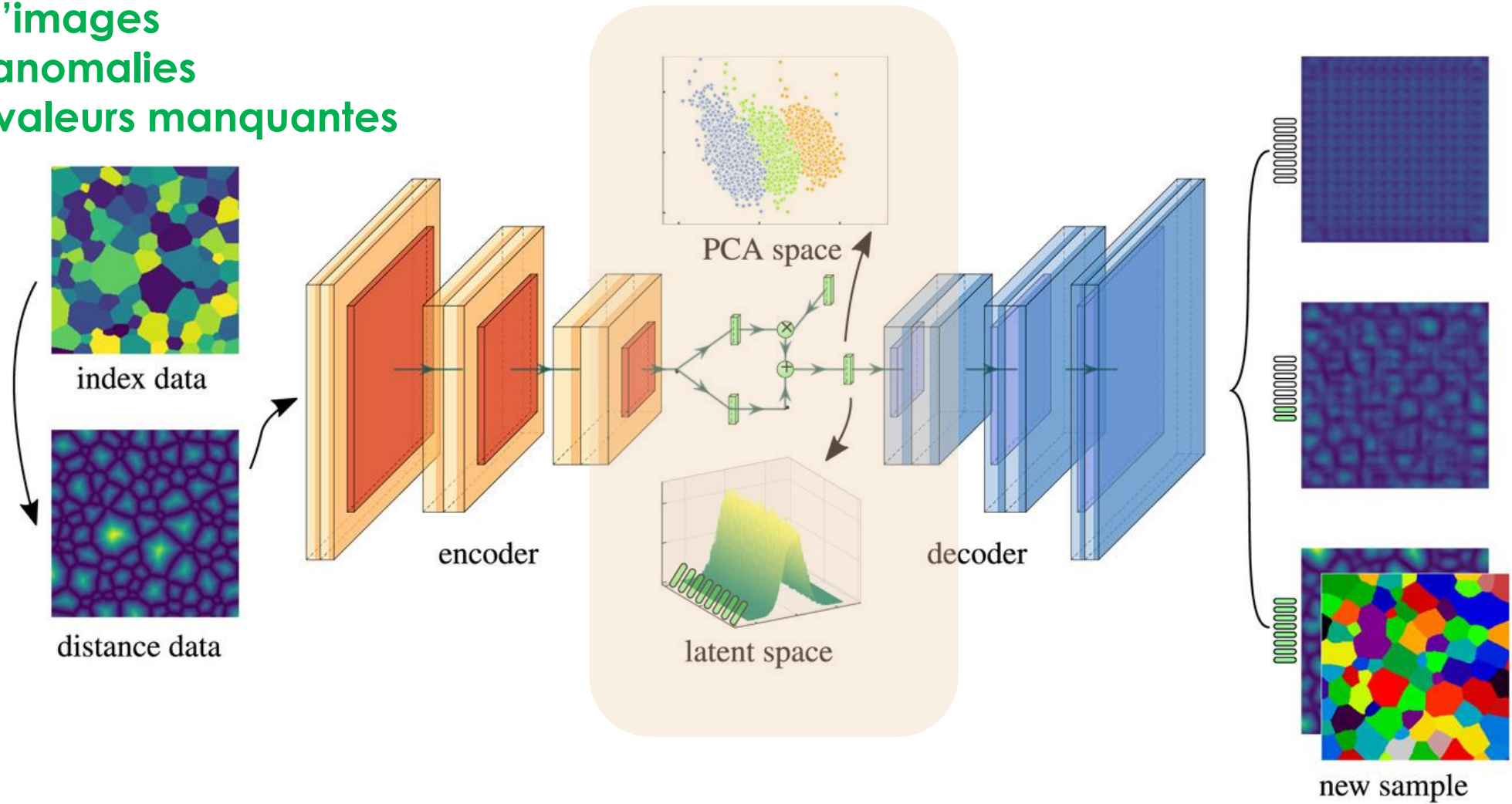
On ne connaît que les X

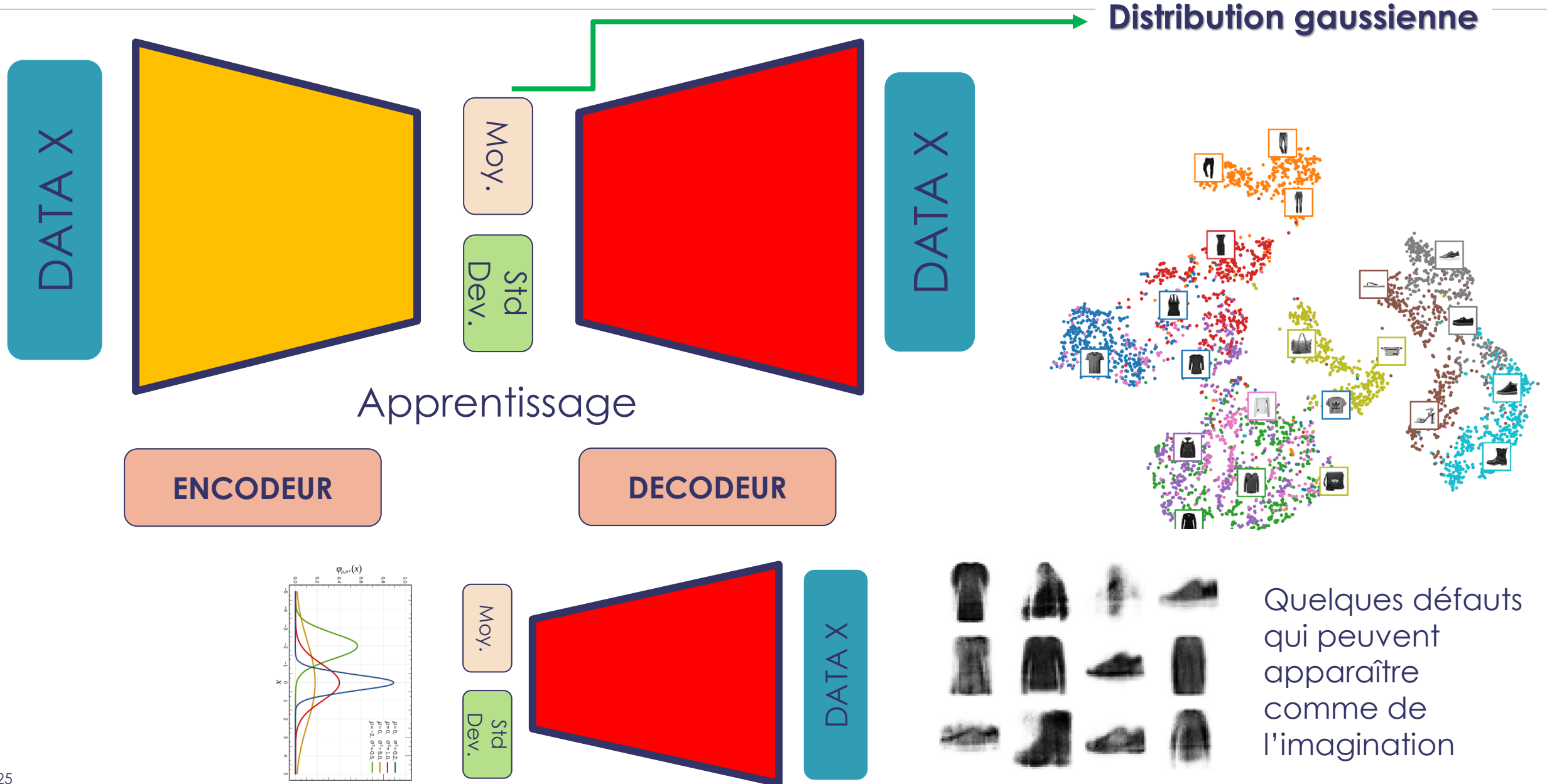


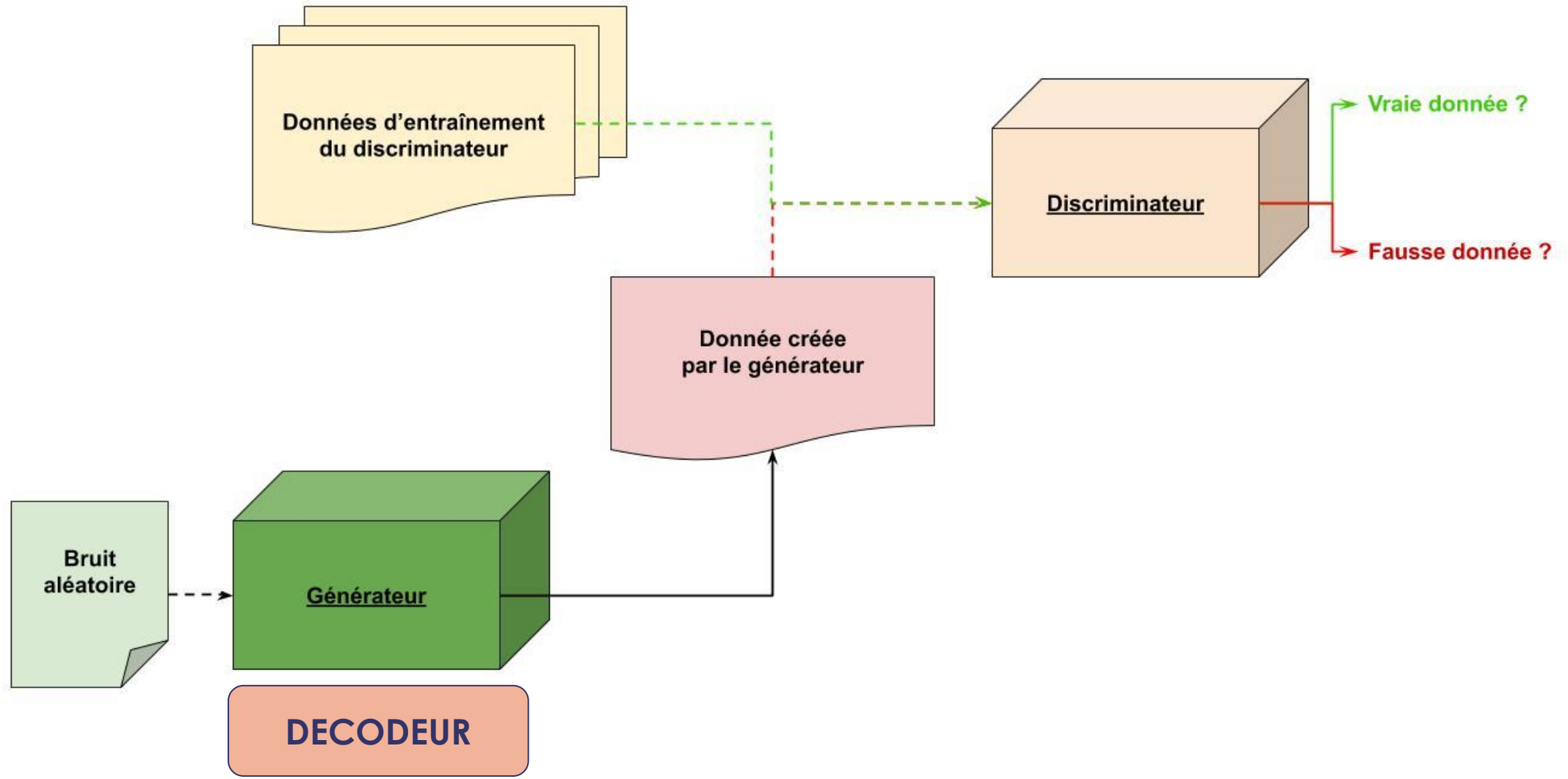
$$Y = X$$



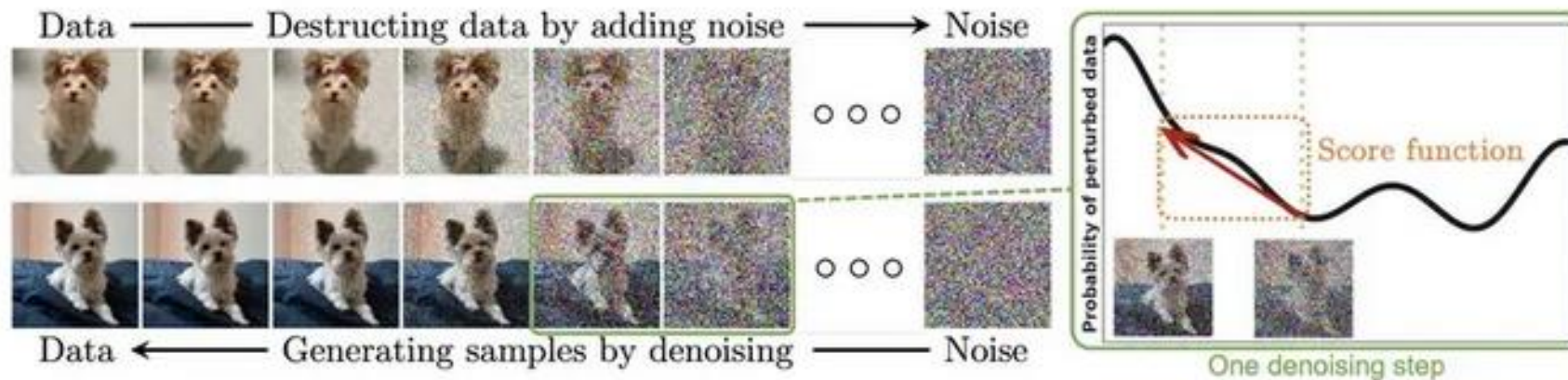
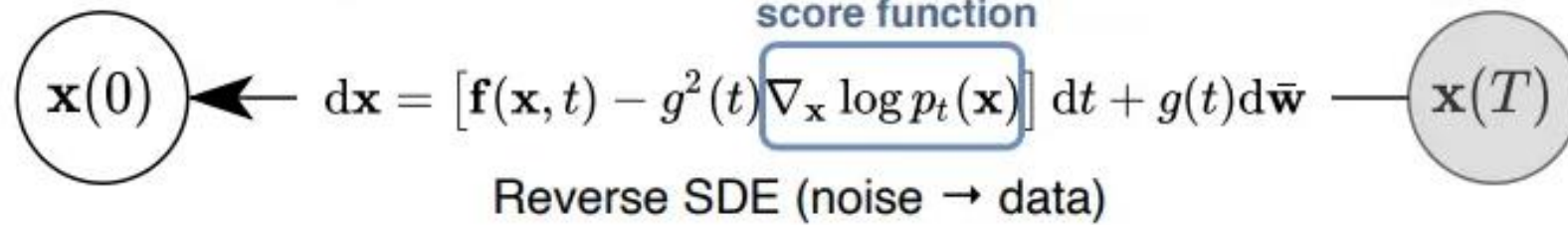
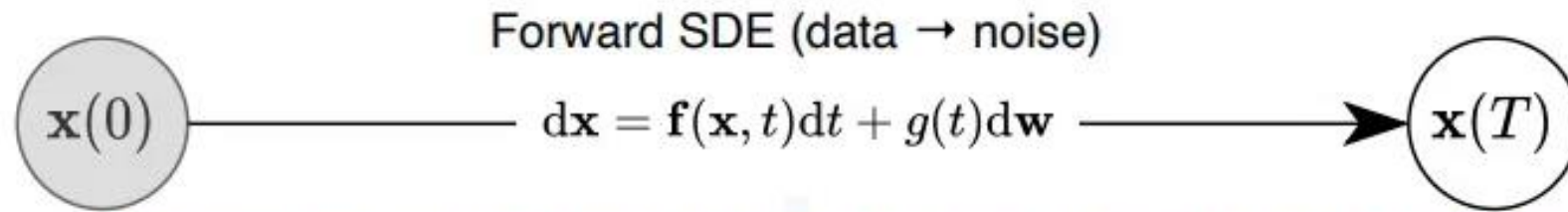
1. Reduction de dimensionalité
2. Extraction de caractéristiques
3. Débruitage d'images
4. Recherche d'images
5. Détection d'anomalies
6. Création de valeurs manquantes







➔ Amélioration du générateur de données grâce à un discriminateur de « fake »



- Texte vers Image



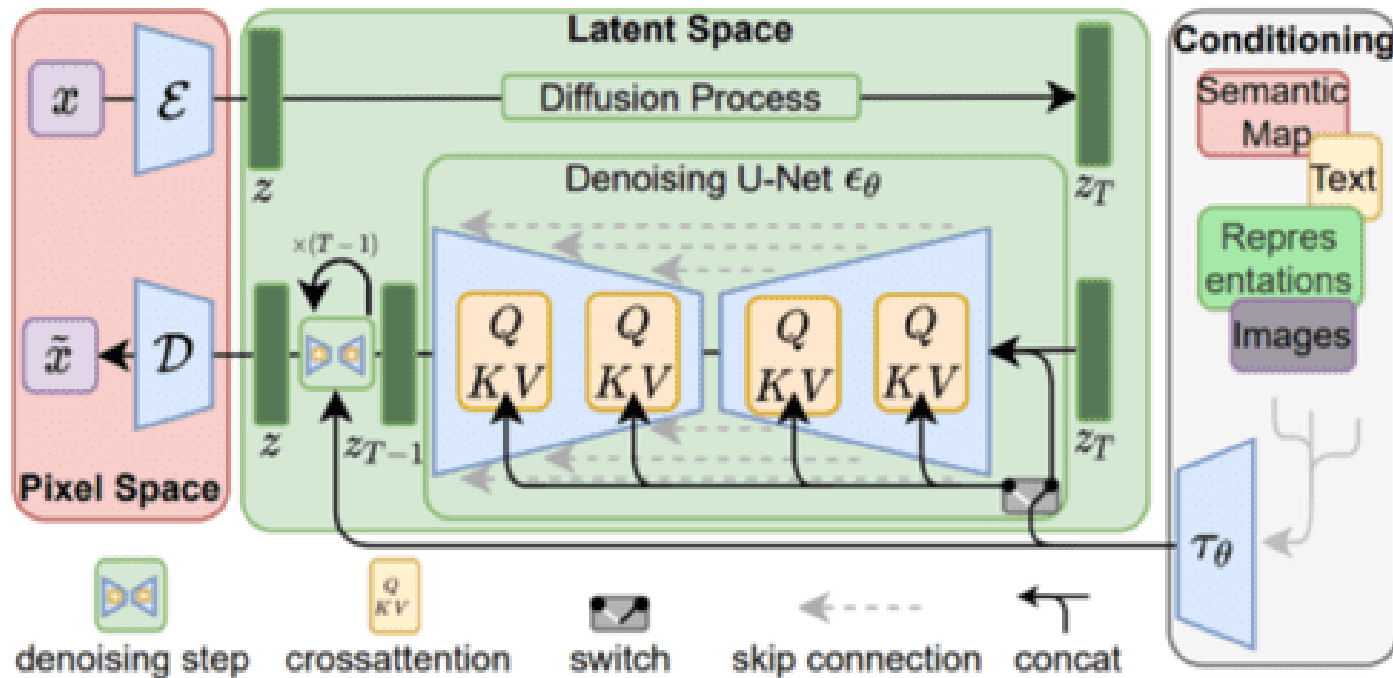
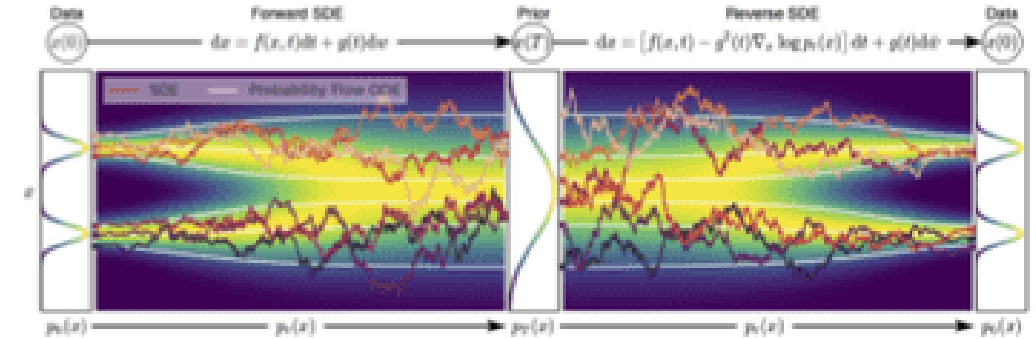
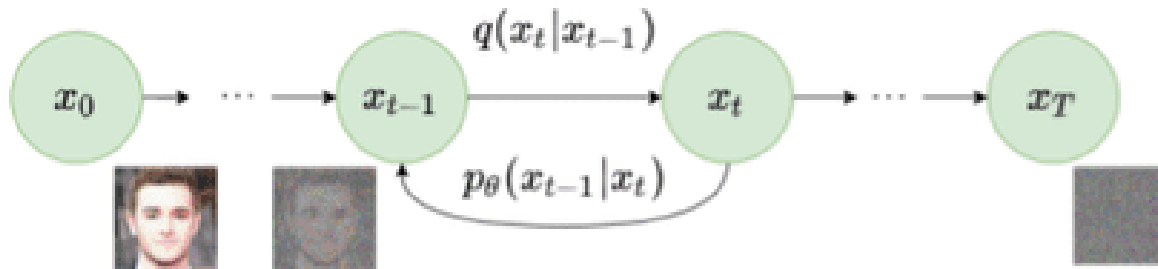
Source : <https://aithub.com/bentoml/stable-diffusion-bentoml/>



Source : Dall-E 2



Source : <https://ai.meta.com/blog/emu-text-to-video-generation-image-editing-research/>



Algorithm 1 Training

- 1: repeat
- 2: $x_0 \sim q(x_0)$
- 3: $t \sim \text{Uniform}(\{1, \dots, T\})$
- 4: $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
- 5: Take gradient descent step on $\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2$
- 6: until converged

Algorithm 2 Sampling

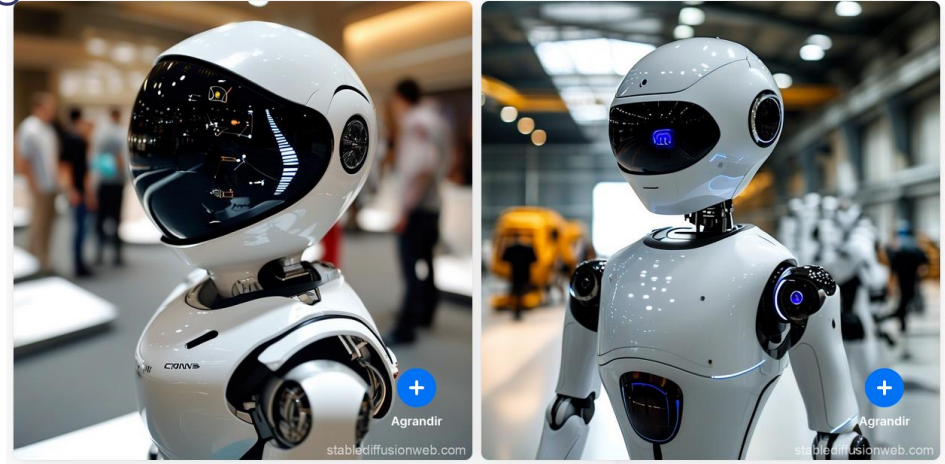
- 1: $x_T \sim \mathcal{N}(0, \mathbf{I})$
- 2: for $t = T, \dots, 1$ do
- 3: $z \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $z = 0$
- 4: $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
- 5: end for
- 6: return x_0

<https://openart.ai/create>



cours sur l'IA dispensés à des adultes dans le cadre professionnel avec des images de modèle d'IA CNN, RNN, DNN, GAN. L'image est réaliste.

<https://stablediffusionweb.com/fr/app/image-generator>



<https://stablediffusionweb.com/fr/app/image-generator>





- ➔ Recherchons quelle partie est importante
- ➔ Identifions les parties qui nécessitent notre attention



Question ? query



traitement du signal



Tous Images Vidéos Actualités Produits Livres Web Plus

Outils

Le traitement du signal **c'est la réalisation d'opérations sur le signal**. – Elaboration de signaux : Synthèse (de parole, de musique), modulation, codage. – Interprétation des signaux : filtrage, extraction/détection d'information, identification, analyse (spectrale ou temporelle) ou mesure.

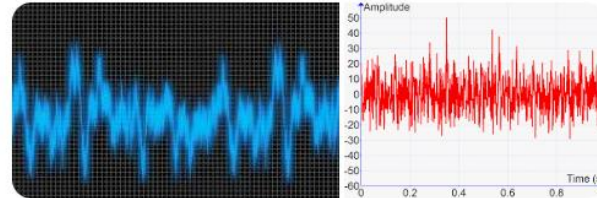


Le Mans Université

https://perso.univ-lemans.fr/doc/cpda_signal PDF

Traitement du signal

Key2 ?



Key1 ?

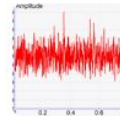


Wikipédia

https://fr.wikipedia.org/wiki/Traitement_du_signal

Traitement du signal

Le **traitement du signal** est la discipline qui développe et étudie les techniques de traitement, d'analyse et d'interprétation des signaux.



LIAS Laboratory

<https://www.lias-lab.fr/fredericlaunay/Cours> PDF

Traitement du signal : les fondamentaux

Initialement destiné à extraire le signal dans un bruit lors de mesures (capteurs), le traitement du signal est largement appliqué en Télécommunication dans ...
50 pages

Key3 ?



FUN MOOC

<https://www.fun-mooc.fr/cours/introduction-au-traitement-du-signal>

Introduction au traitement du signal - Cours

Les thèmes abordés sont la **représentation temporelle et spectrale des signaux**, le filtrage, la numérisation et les applications.

Key4 ?

Traitement du signal

Discipline :

Le traitement du signal est la discipline qui développe et étudie les techniques de traitement, d'analyse et d'interprétation des signaux. [Wikipédia](#)

Les internautes recherchent aussi



Apprentissag
e...



Design



Communicat
ion



Stratégie de
régulation

Commentaires

Réponses: Keys

Attention :
Meilleure Valeur calculée

Question ? query



traitement du signal



Tous Images Vidéos Actualités Produits Livres Web Plus

Outils

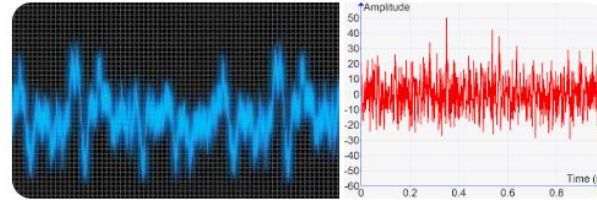
Le traitement du signal **c'est la réalisation d'opérations sur le signal**. – Elaboration de signaux : Synthèse (de parole, de musique), modulation, codage. – Interprétation des signaux : filtrage, extraction/détection d'information, identification, analyse (spectrale ou temporelle) ou mesure.



Le Mans Université

https://perso.univ-lemans.fr/doc/cpda_signal PDF

Traitement du signal



Key2 ? **Valeur 2: Value**

À propos des extraits optimisés • Commentaires

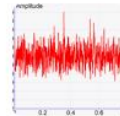


Wikipédia

https://fr.wikipedia.org/wiki/Traitement_du_signal

Traitement du signal

Le **traitement du signal** est la discipline qui développe et étudie les techniques de traitement, d'analyse et d'interprétation des signaux.



LIAS Laboratory

<https://www.lias-lab.fr/fredericlaunay/Cours> PDF

Traitement du signal : les fondamentaux

Initialement destiné à extraire le signal dans un bruit lors de mesures (capteurs), le traitement du signal est largement appliqué en Télécommunication dans ... 50 pages

Key3 ?

Valeur 3: Value



FUN MOOC

<https://www.fun-mooc.fr/cours/introduction-au-traitement-du-signal>

Introduction au traitement du signal - Cours

Les thèmes abordés sont la **représentation temporelle et spectrale des signaux**, le filtrage, la numérisation et les applications.

Key4 ?

Valeur 4: Value

Traitement du signal

Discipline :

Le traitement du signal est la discipline qui développe et étudie les techniques de traitement, d'analyse et d'interprétation des signaux. [Wikipédia](#)

Les internautes recherchent aussi



Apprentissag
e...



Design



Communicat
ion



Stratégie de
régulation

Commentaires

Réponses: Keys

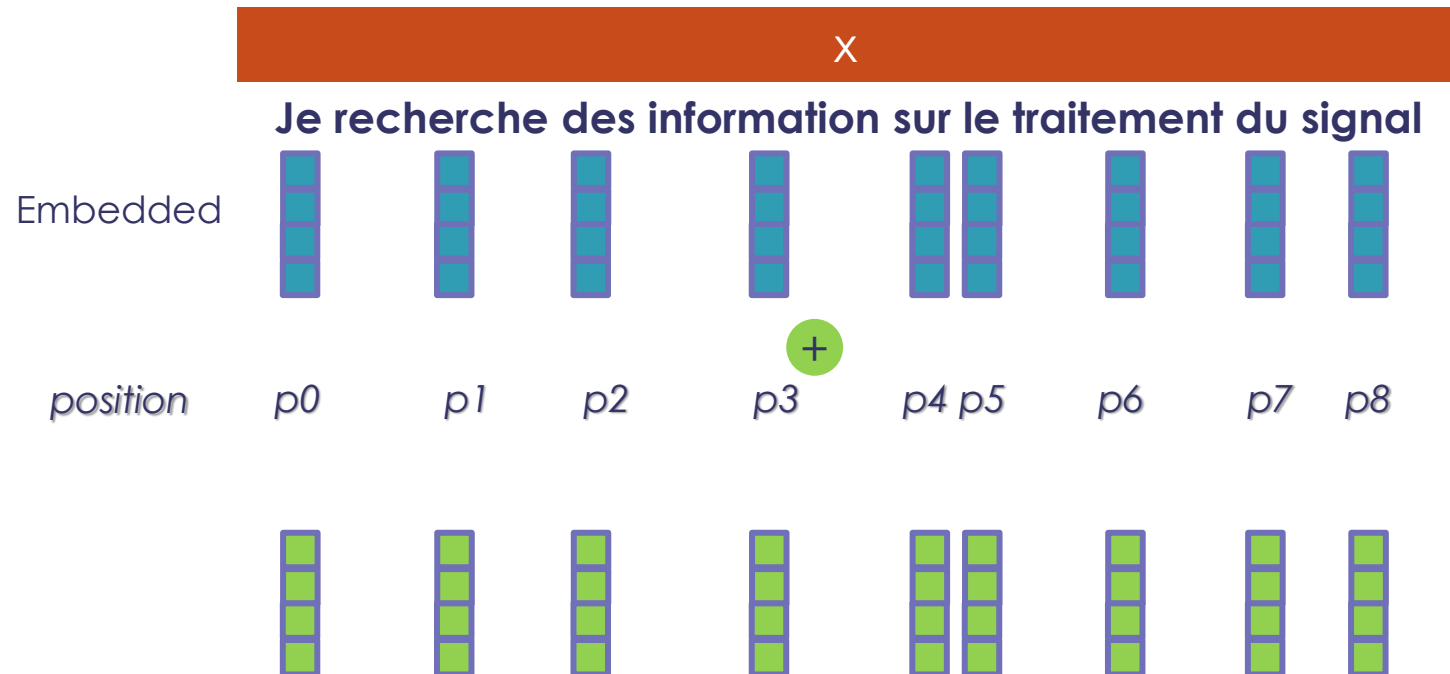
Key1 ?

Valeur 1: Value

Attention :
Meilleure Valeur calculée

L'attention est tout ce que nous avons besoin

1. Encoder la position et la sémantique des informations → Embedded
2. Extraire par calcul le trinôme (query, key, value) pour la recherche
3. Calculer les poids de l'attention
4. Extraire les caractéristiques avec une grande attention



→ Encodage sémantique + position

1. Encoder la position et la sémantique des informations → Embedded

2. Extraire par calcul le trinôme (query, key, value) pour la recherche

3. Calculer les poids de l'attention

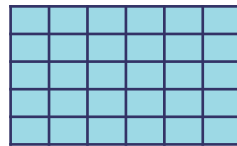
4. Extraire les caractéristiques avec une grande attention

Position

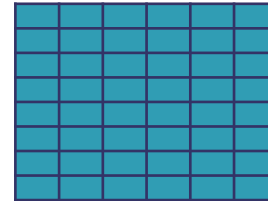


X

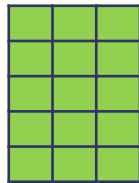
NN



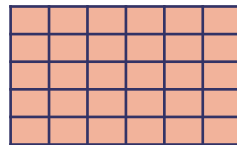
=



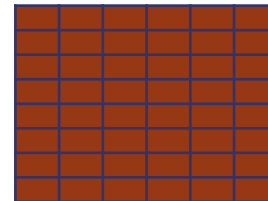
Q: Query



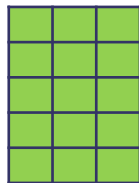
X



=



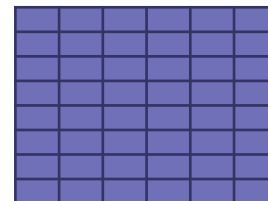
K: Key



X



=



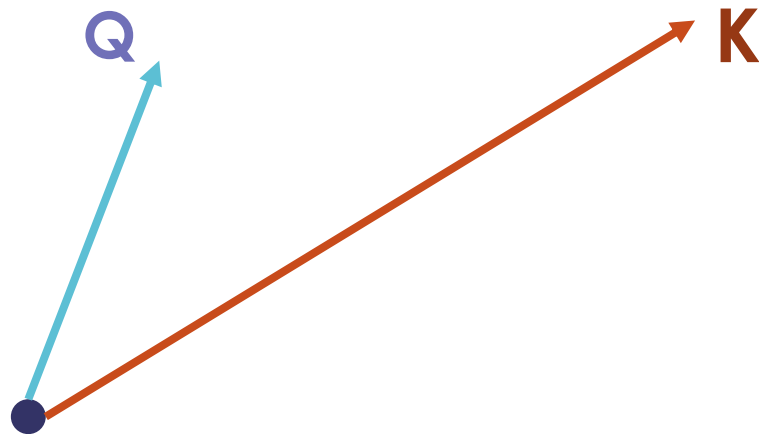
V: Value

1. Encoder la position et la sémantique des informations → Embedded
2. Extraire par calcul le trinôme (query, key, value) pour la recherche
3. Calculer les poids de l'attention

Cosine similarity = métrique de similarité

$$\frac{Q \cdot K^T}{MaE}$$

4. Extraire les caractéristiques avec une grande attention



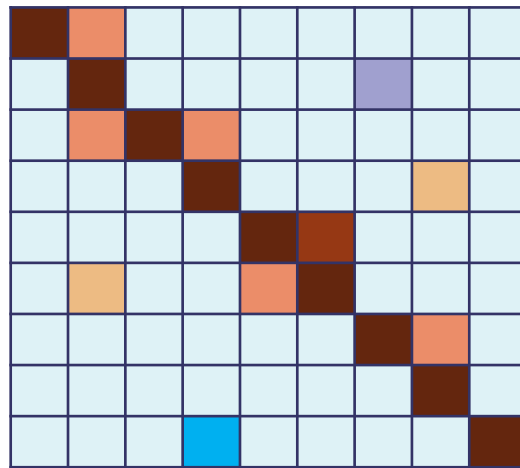
Je
recherche
des
information
sur
le
traitement
du
signal

	Je	recherche	des	information	sur	le	traitement	du	signal
Je	dark	orange							
recherche		dark					purple		
des		orange	dark	orange					
information				dark				orange	
sur					dark	dark			
le		orange			orange	dark			
traitement							dark	orange	
du								dark	
signal				blue					dark

$$\text{softmax}\left(\frac{Q \cdot K^T}{MaE}\right)$$

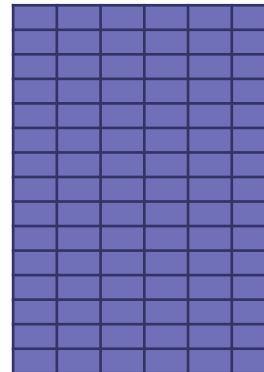
Poids d'attention

1. Encoder la position et la sémantique des informations → Embedded
2. Extraire par calcul le trinôme (query, key, value) pour la recherche
3. Calculer les poids de l'attention
4. Extraire les caractéristiques avec une grande attention



Poids d'attention

x



Value

=



Sortie

$$\text{softmax}\left(\frac{Q \cdot K^T}{MaE}\right) \cdot V$$

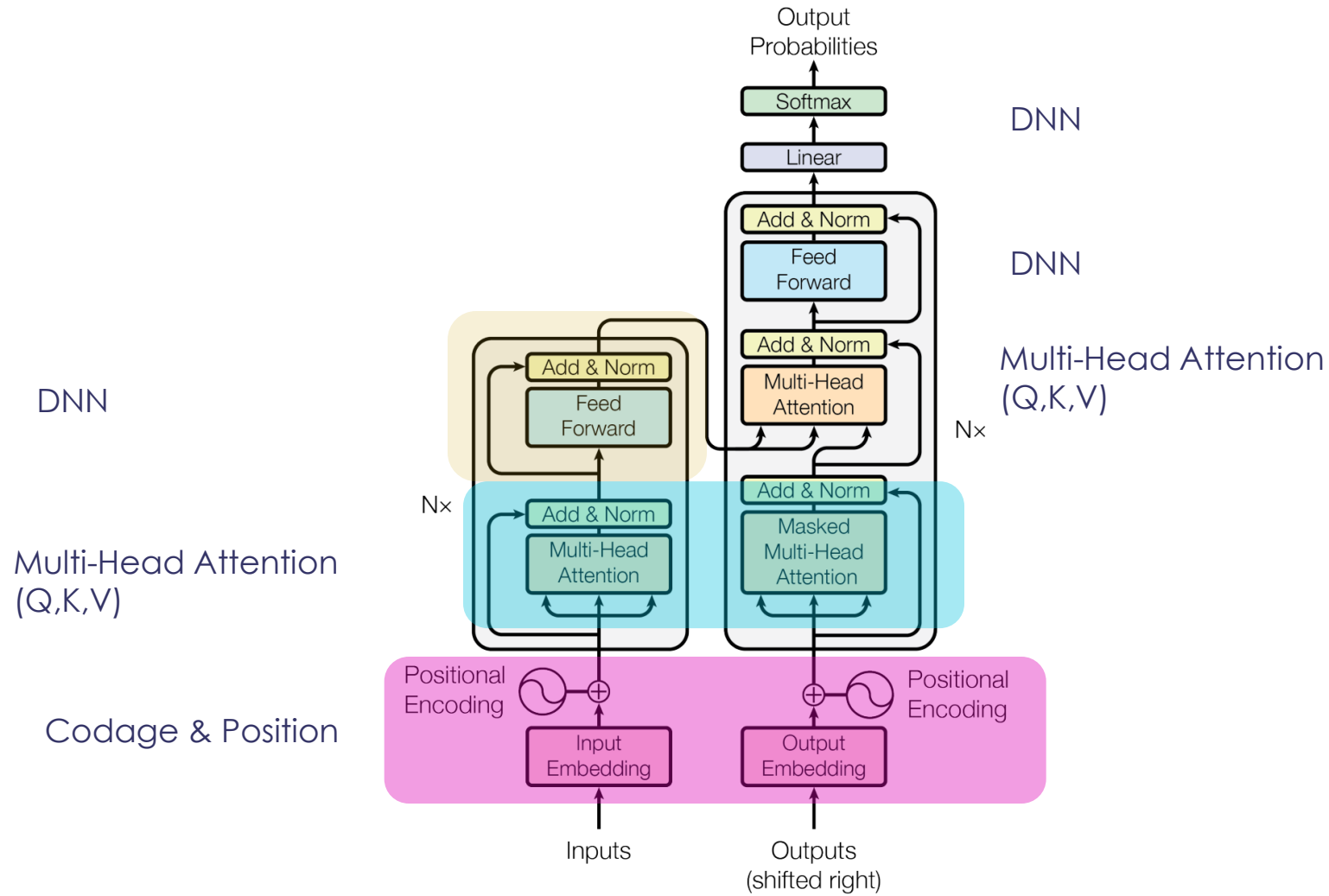
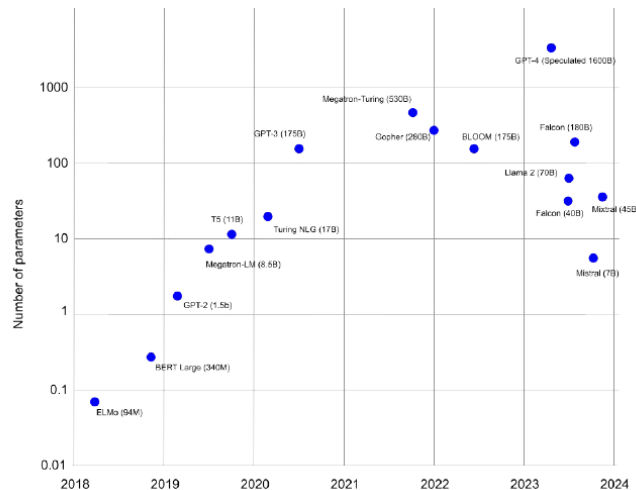
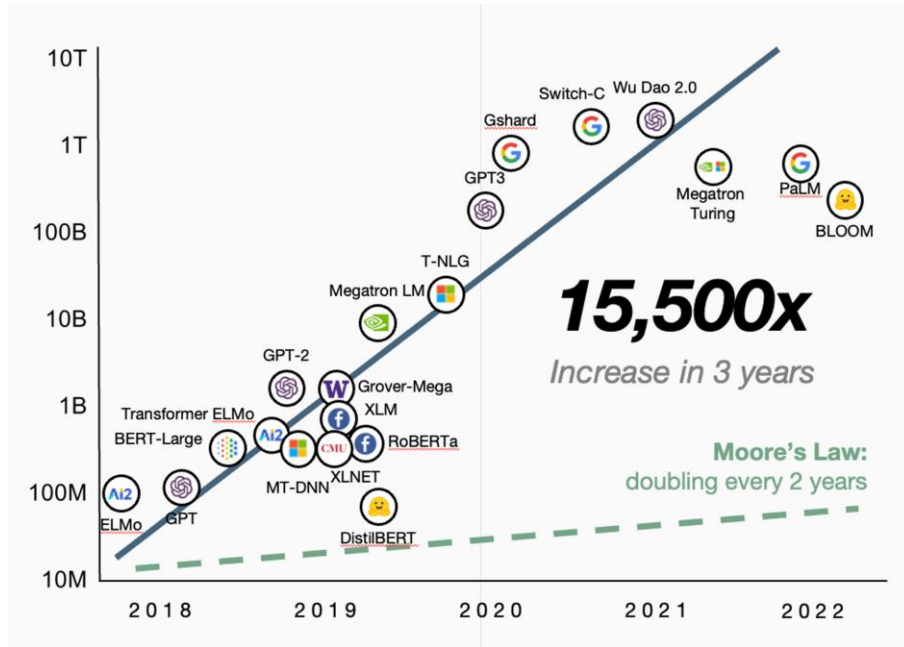
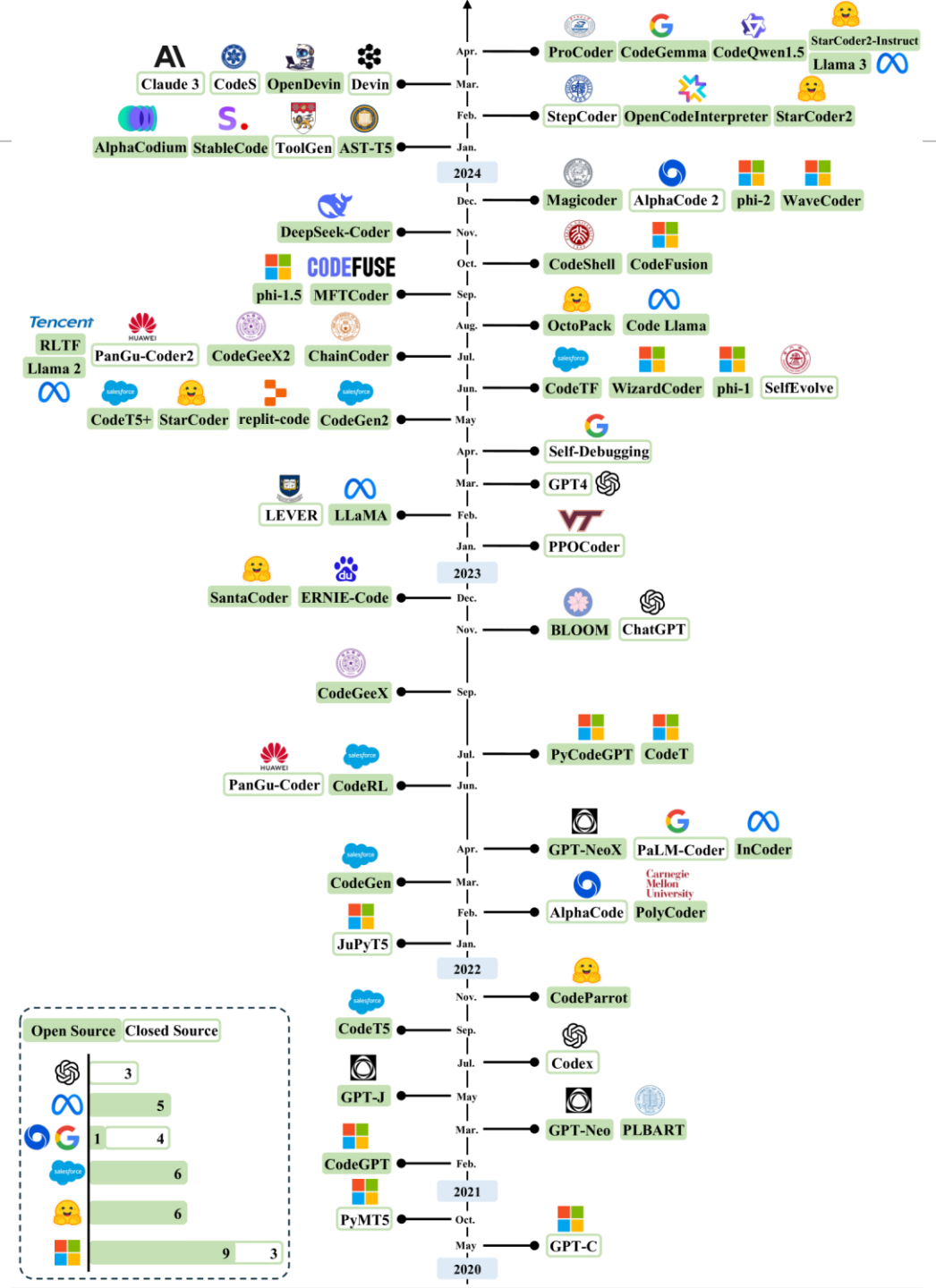


Figure 1: The Transformer - model architecture.

Les transformers !



➔ Vers l'IAG ?



What do you need to train a large language model ?

A truckload of data

Transformers are ravenous. You need to feed them with a substantial and hard-to-come-by dataset.



A mighty compute infrastructure

GPUs with high throughput and large VRAM can execute this training in a reasonable amount of time.



A copious amount of electricity

Storage, memory and compute power consume a lot of electricity.



An abundance of manpower

Cleaning the dataset, making experiments, monitoring SOTA advancements is a lot of work.



Single Agent Interaction



Multi Agent Interaction

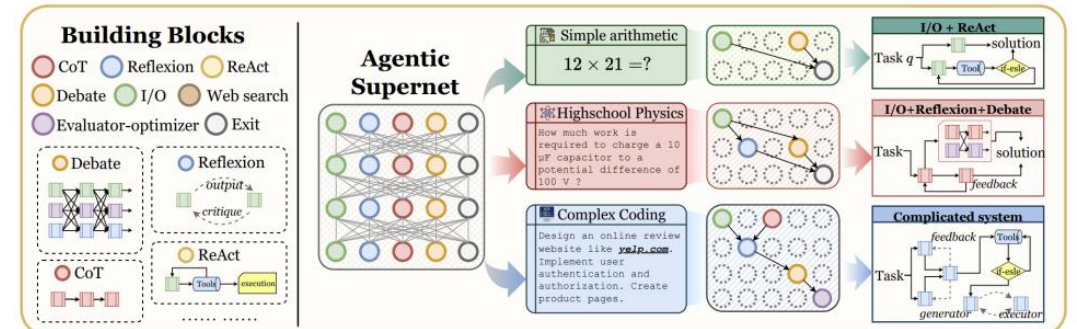
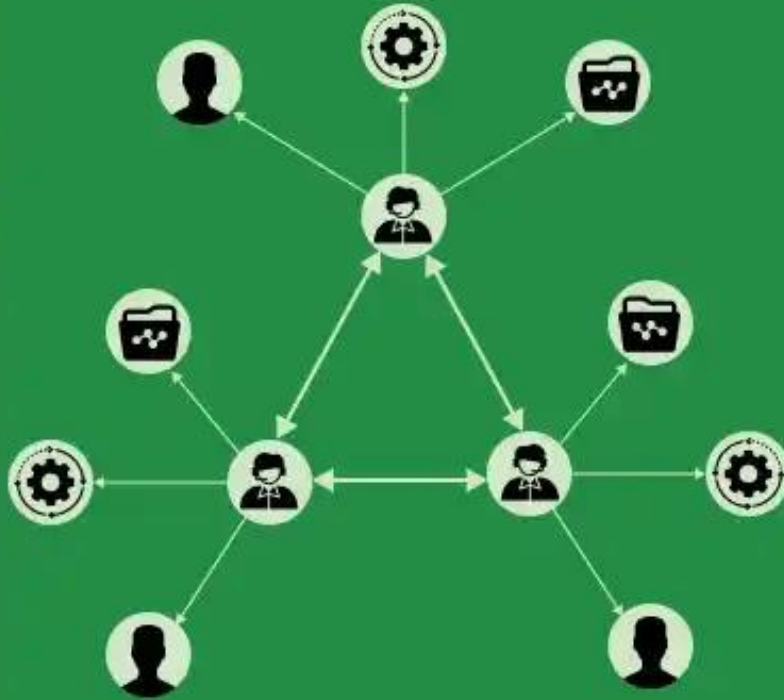
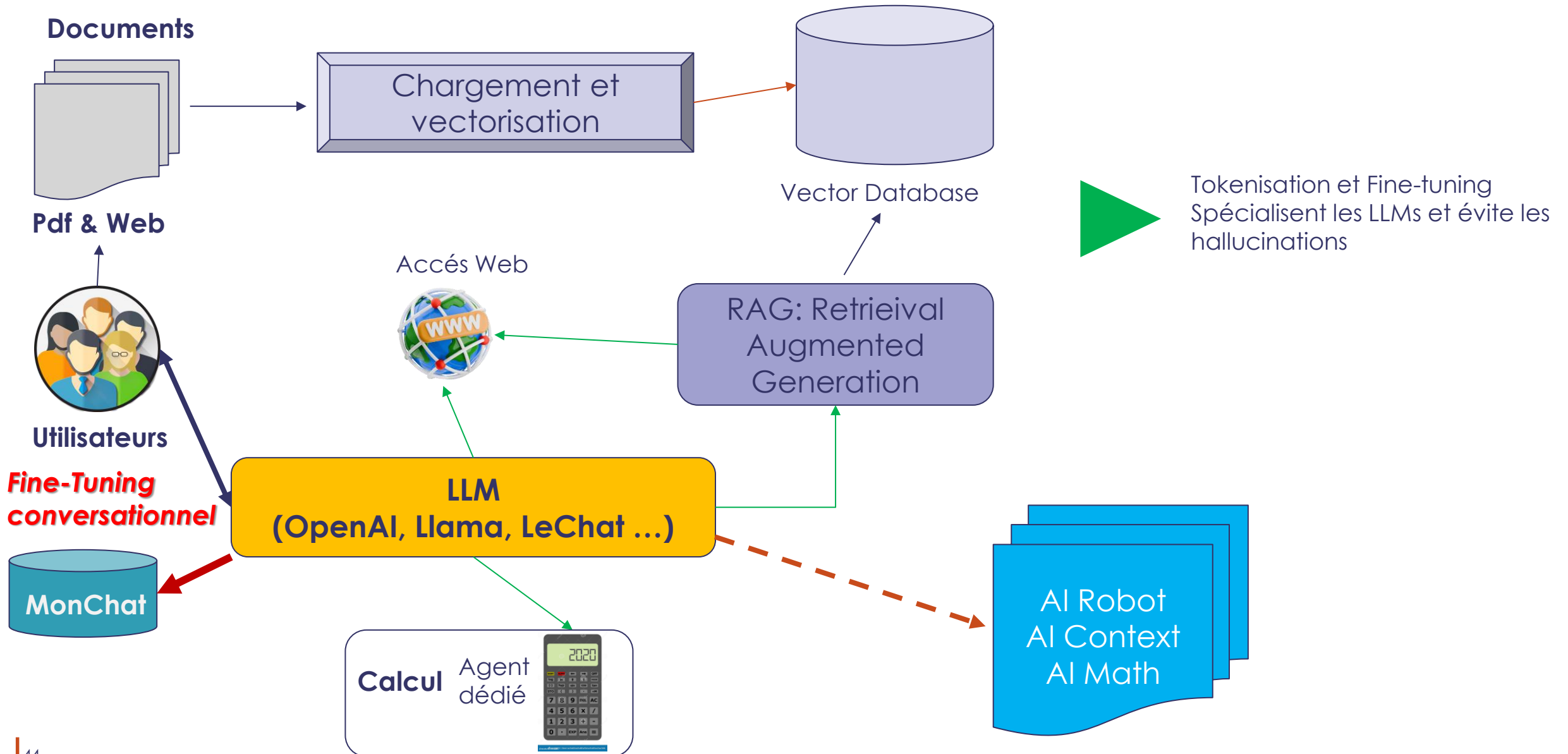
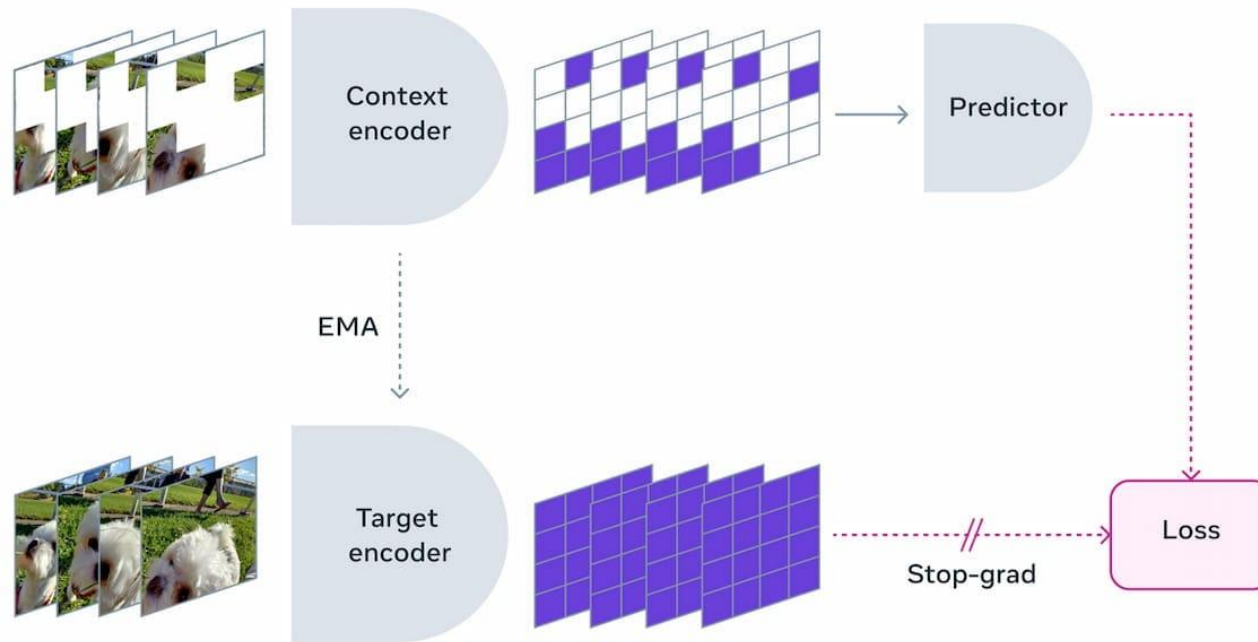


Figure 1. (Left) The building blocks of MaAS; (Right) When confronting different queries, the agentic supernet adaptively samples tailored multi-agent architecture in a query-dependent manner.



Apprendre avec les vidéo comme un humain !

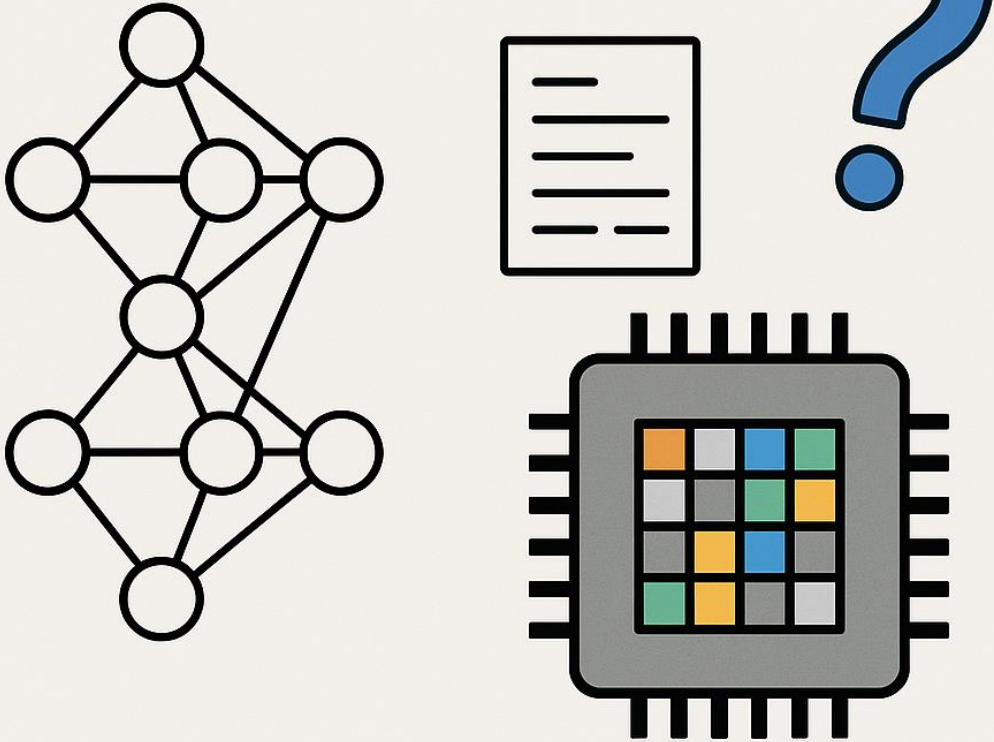


- 5 à 10 ans de dev!
- StartUp AMI (LeCun)
- 1 Milliards de \$ pour une idée
- Ca marche pour des robots

$9 \cdot 10^{14}$ Octets Web : Apprentissage par du Texte

- 10^{14} Octets vu par un enfant de 4 ans
- Quelques semaines pour apprendre à conduire
- Créer un modèle du monde avec des videos pour apprendre le contexte/les concepts et non pixels par pixels

How to implement a neural network on an FPGA with all the computations?



How to implement a neural network on an FPGA with all the computations?



Questions to answer

- What component I focus on ?
 - What is its architecture ?
 - What is its ressources ?
 - What is the price I want to pay for ?
- What kind of model I want to implement ?
 - MLP – DNN
 - CNN
 - RNN
 - LLM
 - Auto-encoders
 - Etc

Knowledge of the market

I need an AI expert around me !
Or I become one of them !



DIGITAL DILEMMAS & OTHER CONUNDRUMS

(The sarcastic FPGA developer's guide)

What performances
do I expect?

Speed of
response -
Latency

Frequency of
data stream

Number of
model to
parallelize

What!
What!
Hugh?

How do the
componentt
receive input
data?

Do I need
pre-process-
to convert
my input
to model
expected
input

How do I
know it will
work?

→ I need
to foreen
what,
how?

What performances do I expect ?

Speed of response – Latency

Frequency of data stream

Number of model to parallelize

Hein! What ?
Hugh !

How do the component will receive input
data ?

Do I need pre-processing to convert my
input to the model expected input

How do I know it will work ?

....

Foressen what ?

The R&T project
"Test of hardware
Inferring Neural
Network" deals
with all these issues!



Formation et tests matériels, firmwares et logiciels pour l'implémentation d'algorithmes d'apprentissages profonds

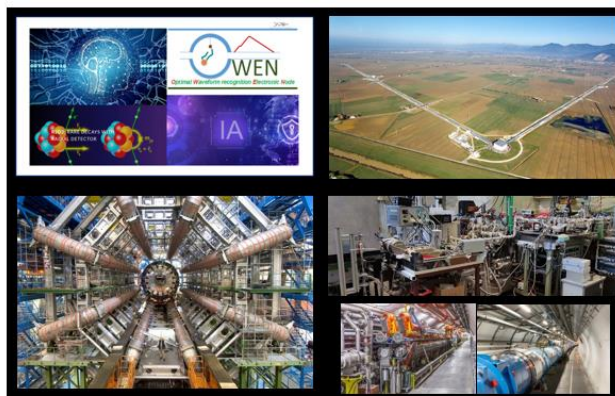


THINK

Testing Hardware Instantiations of Neural Kernels

2020-2023

L'ambition de THINK est de comprendre et évaluer la possibilité d'implanter des algorithmes de réseaux de neurones relativement en amont pour nos futurs besoins en instrumentation.



Membres du projet THINK :
J.-P. Cachemiche (CPPM)
V. Gligorov, O. Ledortz (LPNHE)
D. Etasse, J. Hommet (LPC Caen)
F. Bellachia, S. Lafrasse (LAPP)
R. Bouet, F. Druillole, A. Rebi (CENBG)
F. Magniette, E. Sauvan (LLR)
G. Aad, Y. Boursier, T. Calvet, E. Fortin, E. Monnier (CPPM)
J. Frontera (CEA IRFU)

S. Geiger, W. Perrin, M. Fotzé,
J. Wurtz

F. Magniette, M. Melenne

2024-2026

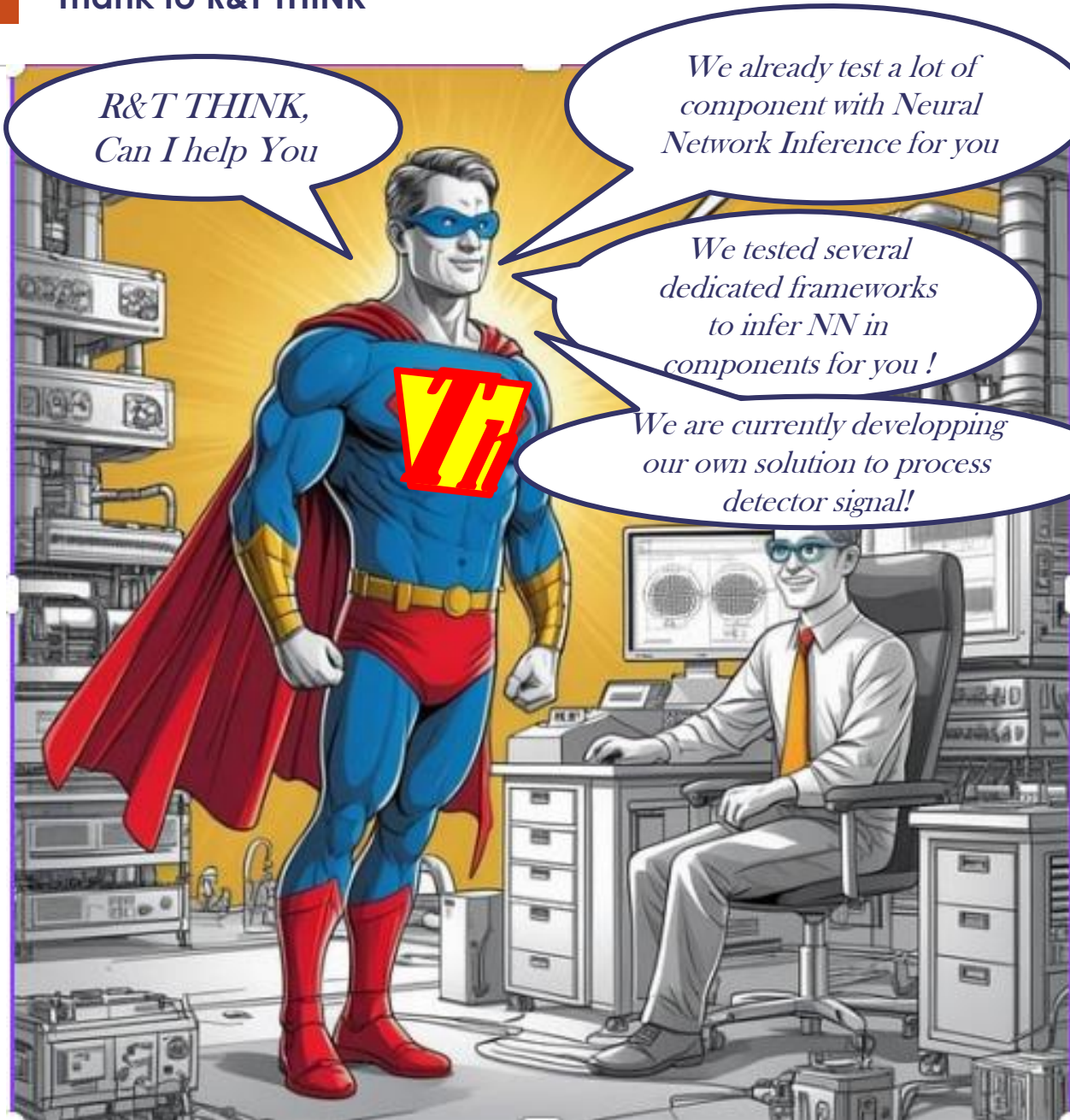
G. Aad, R. Bertrand

F. Druillole, B. Mansoux, J.
Domange

S. Viret, G. Galbit, Q. David
(2025), X. Chen



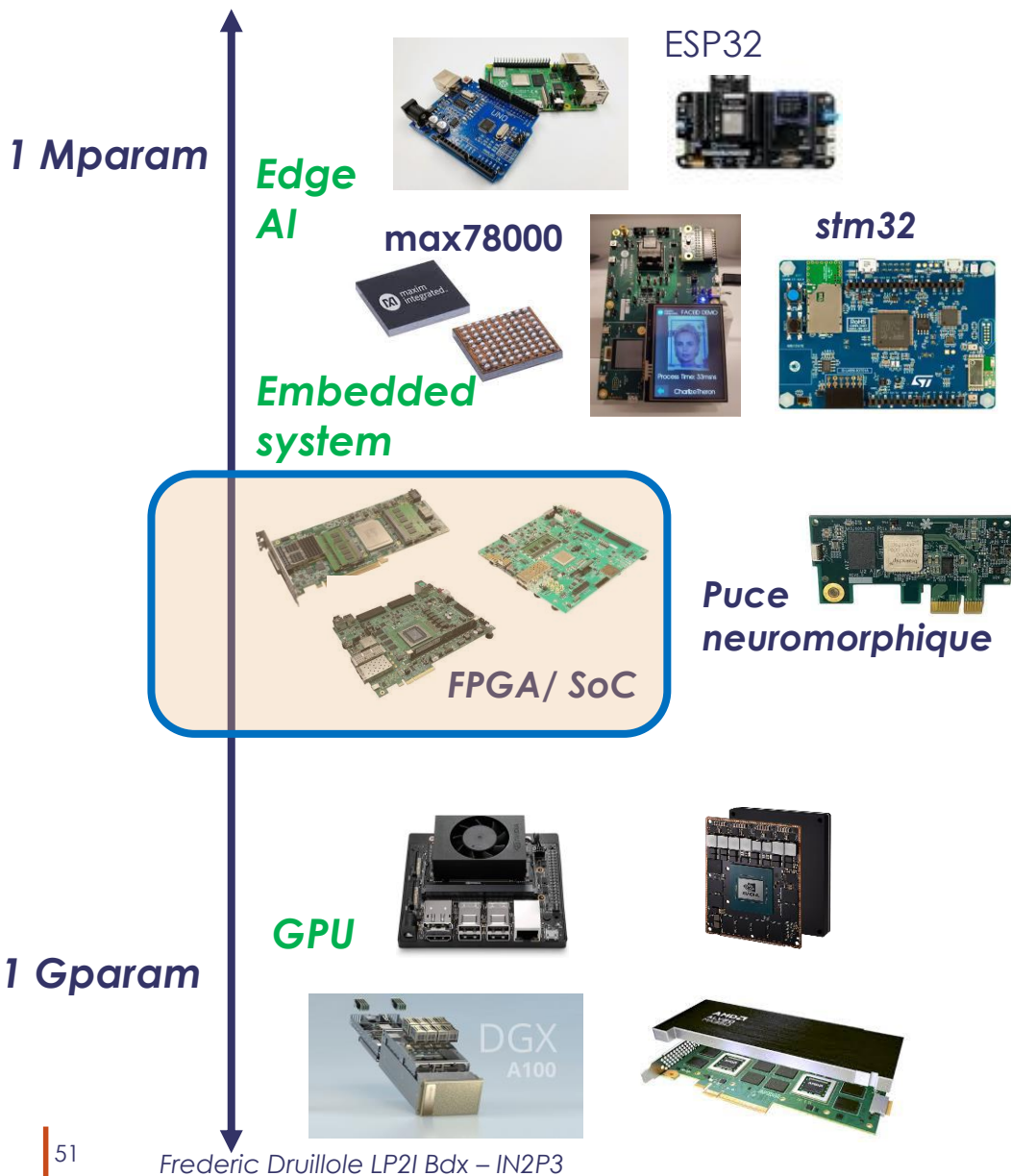
Thank to R&T THINK



Fredéric DUMOLLE L2 Z1 Box - IN2FS



Choose the platform – Action to do



OPTIMIZE THE MODEL

Input

Neural Network model with a huge number of neurones

Materials:

Create a synthetized model with adequate ressources for the AI platform

Objectives

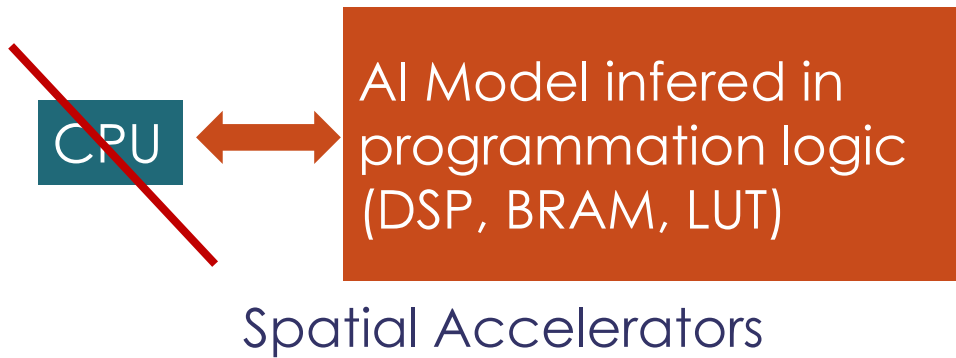
Transfer of the NN model form computer to component level

Optimize the model to find a compromize between performances and component ressources

Methods

Pruning:
Remove weight near 0

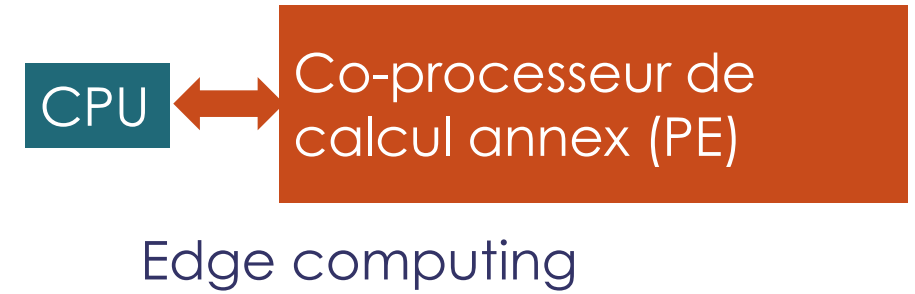
Quantization:
Coding data and weight in reduced format
Fixed point



Based on hardware functions dedicated to calculus without software

ASIC
Neuromorphic engines
FPGA

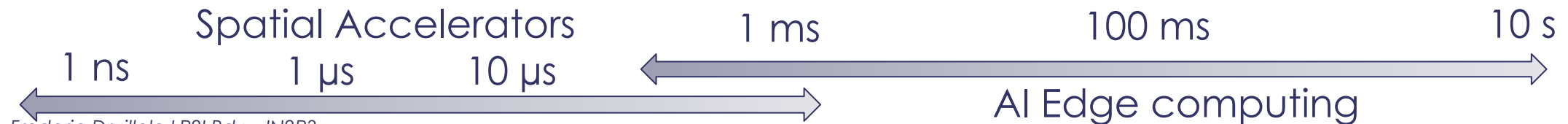
2 In development



Based on software

MMPA
GPU
FPGA
PU redesigned for AI (TPU, KPU...)

1 Standard design



Spatial Accelerators

- **VIVADO HLS**
 - HLS4ML
 - FINN (Xilinx R&D)
 - ChaiDNN
 - **HADDOC2**
 - **DNNWeaver**
 - ...
- } Consortium ?

AI Edge computing

- **TensorFlowLite**
 - **Ax8i (max78000)**
 - **ST Microelectronics**
 - CubeIDE + Cube-AI
 - **VITIS AI (AMD)**
 - **VITIS AI (AMD)**
 - **OpenVino (Intel)**
 - **N2D2 – DeepGreen Eclipse Aidge**
 - **Edge Impulse (société)**
- } Solution Européenne Open

➔ Framework to help accelerating calculus of AI models to infer them.
Use well-know models (DNN, CNN)
Complex to infer models in low resources components (Our need: multi-channels, near detector)

➔ Think P2: Creation of framework dedicated to ALMD-Xilinx & Intel in VHDL

100

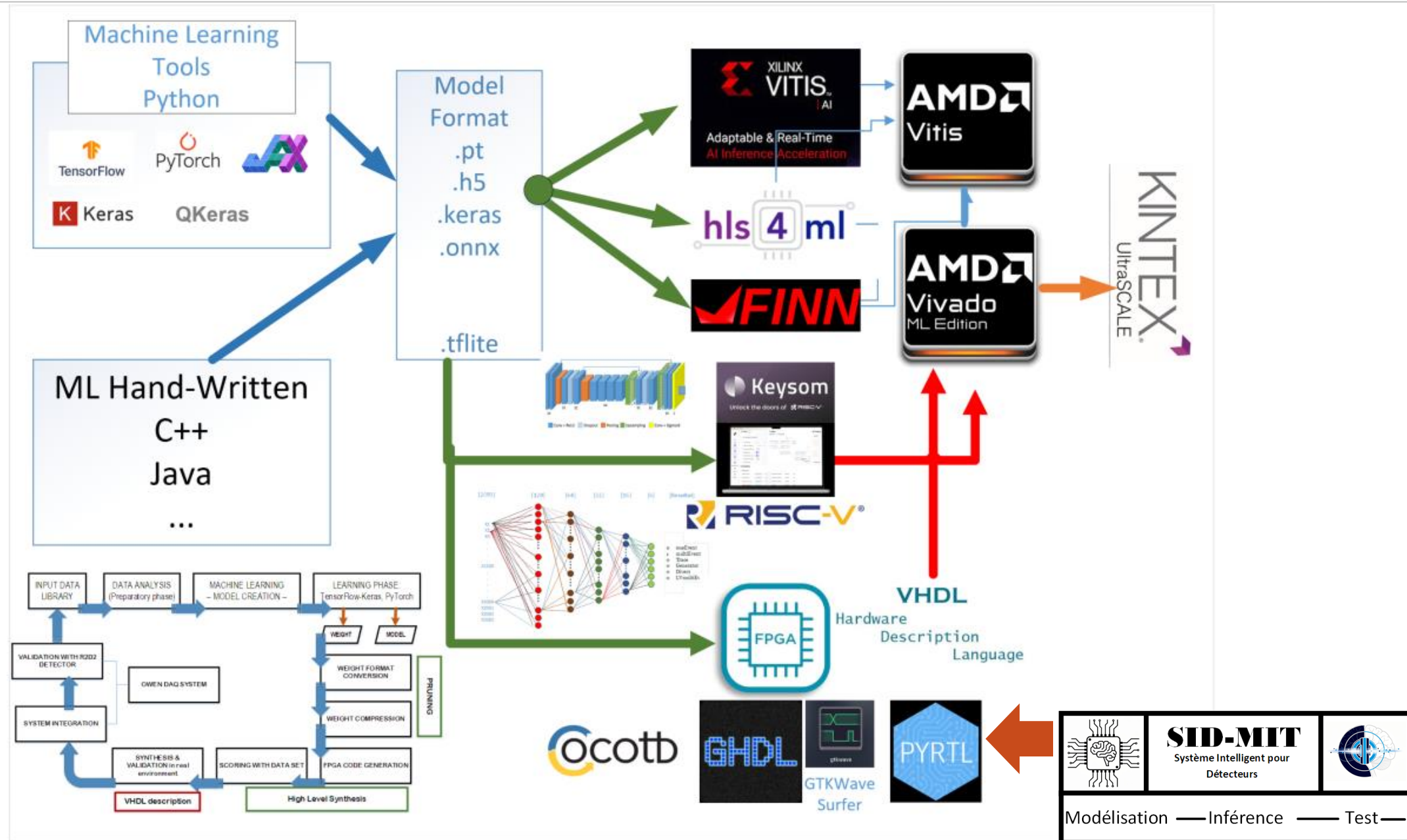


IA Components comparaison

Critères	CPU	CPU+AI-CoPro	GPU	FPGA	ASIC
Adaptability (to various model)	High	High	Moyen	Low	Aucune
Calculus power	low	Mean-High	High	High	Mean
Latency	Mean (ms)	Mean (ms)	Faible (μs)	Faible(10 ns)	Très faible
Input stream	Low	Mean	High	High	High
Parallelism	Low	Mean	High	High	High
Electrical power consumption	Mean	Mean	Mean	Mean	High
Easy to use	Facile	Moyen	Mean	Complex	Very Complex
Density of model	Moyen	Low	Low	High	High



Inference sur FPGA



- 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%

Une question de budget (nombre de voies DAQ)

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

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

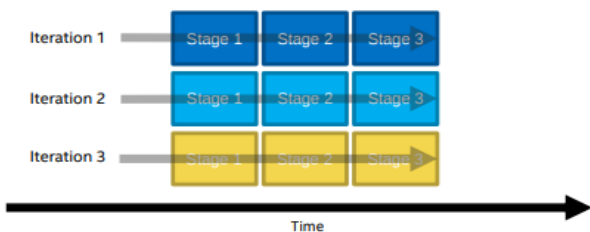
	ALUTs	FFs	RAMs	MLABs	DSPs
Ch1 vanilla	602 (0%)	547 (0%)	4 (0%)	2 (0%)	1.5 (0%)
Ch1 pipeline	610 (0%)	624 (0%)	4 (0%)	5 (0%)	1.5 (0%)
Ch1 unroll	515 (0%)	245 (0%)	4 (0%)	1 (0%)	0 (0%)
Ch1 u+p	515 (0%)	245 (0%)	4 (0%)	1 (0%)	0 (0%)

	ALUTs	FFs	RAMs	MLABs	DSPs
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					

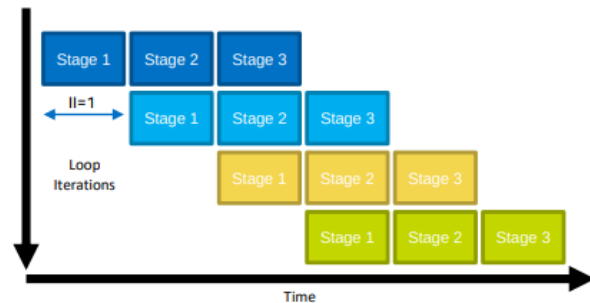
	ALUTs	FFs	RAMs	MLABs	DSPs
Ch3 vanilla	1 408 (0%)	1 809 (0%)	30 (1%)	12 (0%)	2.5 (0%)
Ch3 pipeline	2 093 (0%)	4 460 (0%)	32 (1%)	48 (0%)	2.5 (0%)
Ch3 unroll	118 041 (14%)	36 737 (2%)	5 (0%)	37 (0%)	0 (0%)
Ch3 u+p	27 524 (3%)	42 546 (2%)	1 855 (68%)	298 (1%)	75 (5%)

	ALUTs	FFs	RAMs	MLABs	DSPs
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%)

Unroll



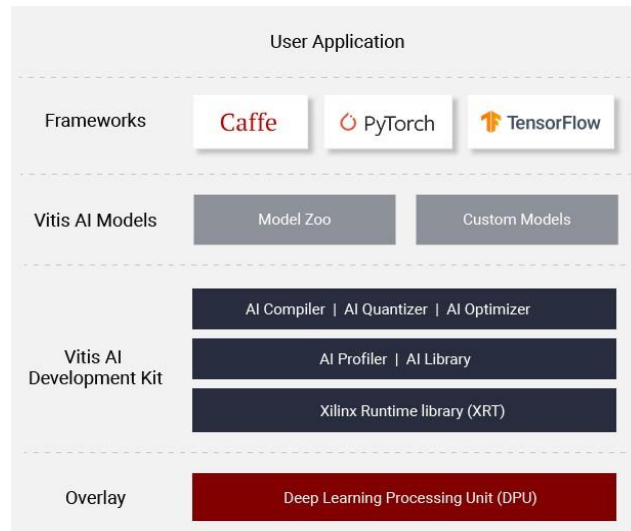
Pipeline



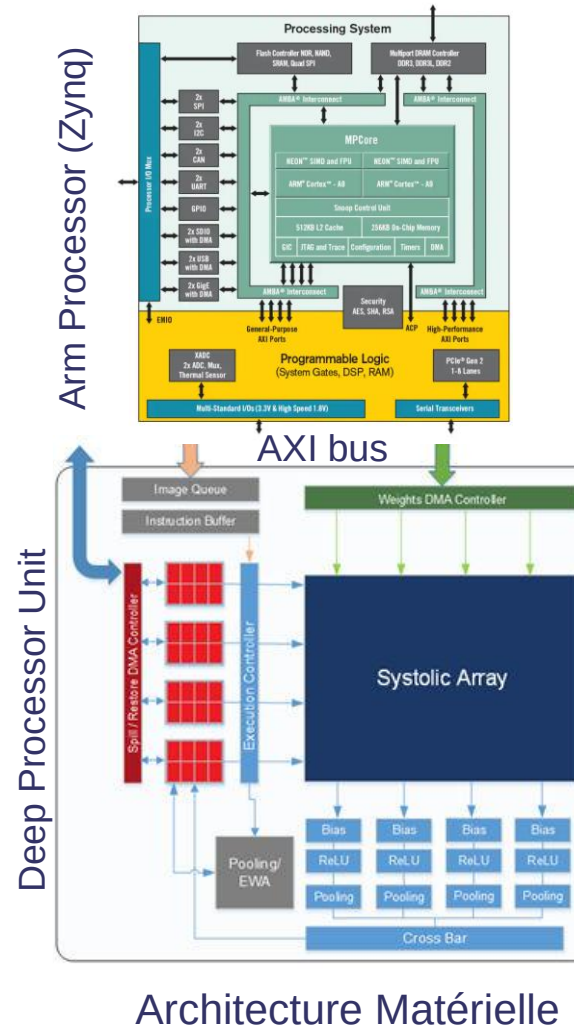
Question of HLS optimisation

Architecture Vitis AI (edge computing)

https://www.xilinx.com/html_doc/s/vitis_ai/1_4/zmw1606771874842.html



- **tensorflow** : model.pb
- **pytorch** : model.xmodel
- **caffe** :
model.prototxt/model.caffemodel



VITIS AI PROCESS
(Xilinx)

Keras/PyTorch Model in
floating point

preprocessing

Quantize Weight and
calibrate Activation
function

Generate DPU model

Compile C++ code with
optimization

Démonstration avec le “model Zoo” : Resnet50

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



Avion 91%



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

Corail cerveau 98%



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

oseille 39% /
vache 31%



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

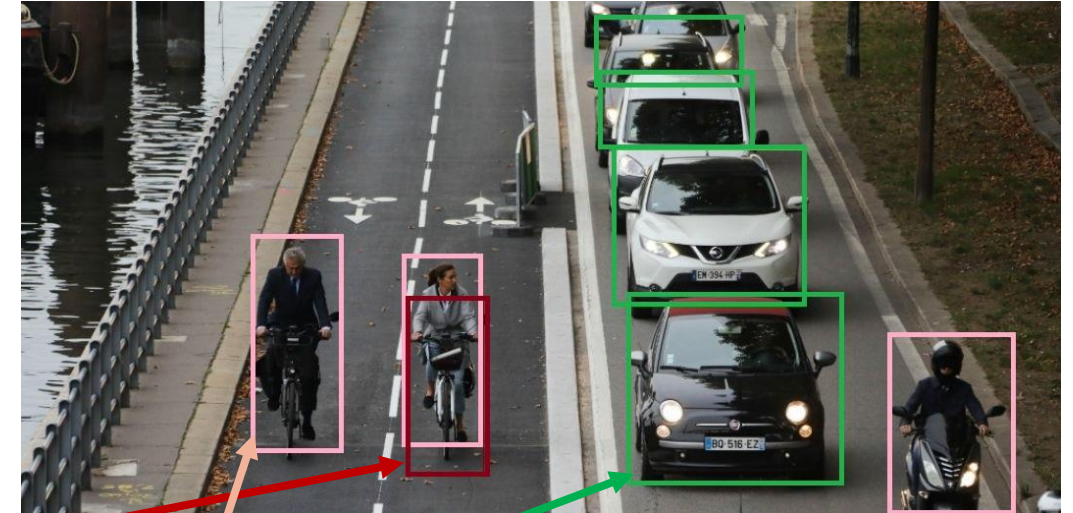


→main.cc de Resnet50 :

- **ListImages()** : fonction qui charge les images que l'on veut traiter.
- **LoadWords()** : fonction qui charge les différentes catégories que connaît 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 les données nécessaires.

```
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 image */
    cv::imshow("Classification of ResNet50", imageList[i]);
    cv::waitKey(10000);
}
```

YOLO : traitement multicible sur images

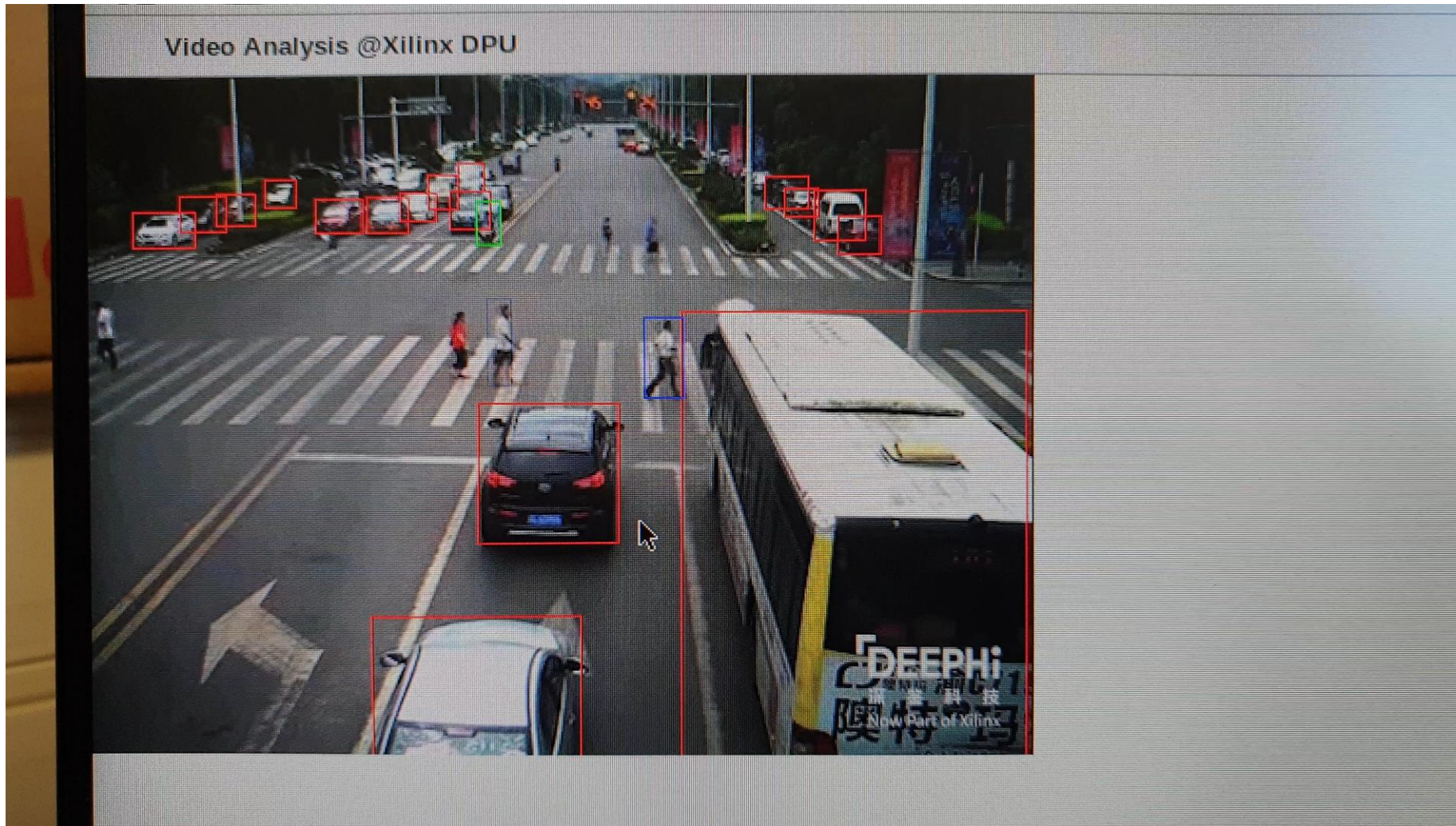


RESULT : 1	395.669 302.705 471.263 483.167	0.779274
RESULT : 6	586.168 16.2274 734.577 84.4693	0.979318
RESULT : 6	604.624 151.448 797.66 310.705	0.968632
RESULT : 6	618.412 299.663 837.151 491.764	0.928414
RESULT : 6	589.709 72.6563 747.69 149.985	0.607446
RESULT : 14	236.443 240.04 327.627 457.719	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



Traitement vidéo en ligne sur ZCU102

YOLO : traitement multicible sur video



Traitement vidéo en ligne sur ZCU102

Conclusion

- Les réseaux de neurones
- Les bases DNN, CNN et RNN
- Les modèles génératifs
- La phase d'apprentissage
- Gestion des informations sous forme numériques

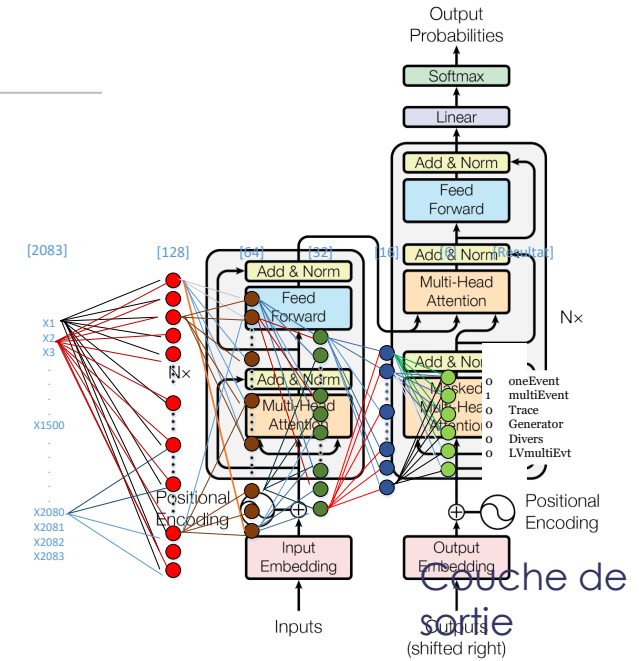
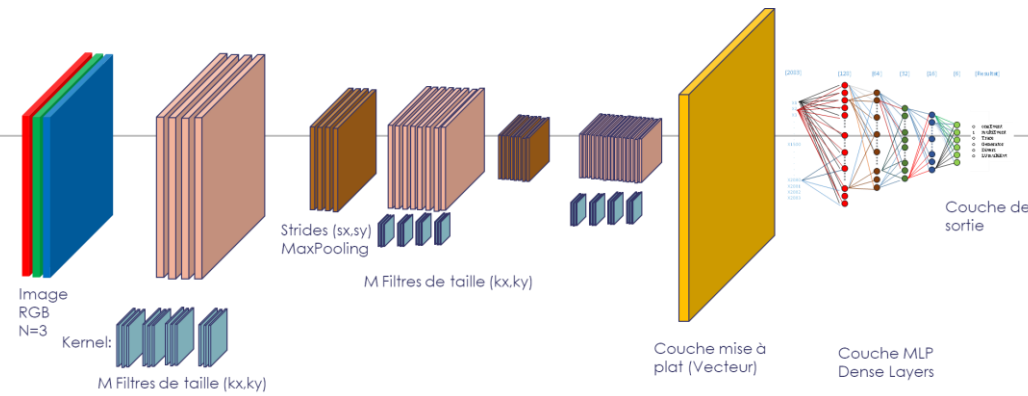
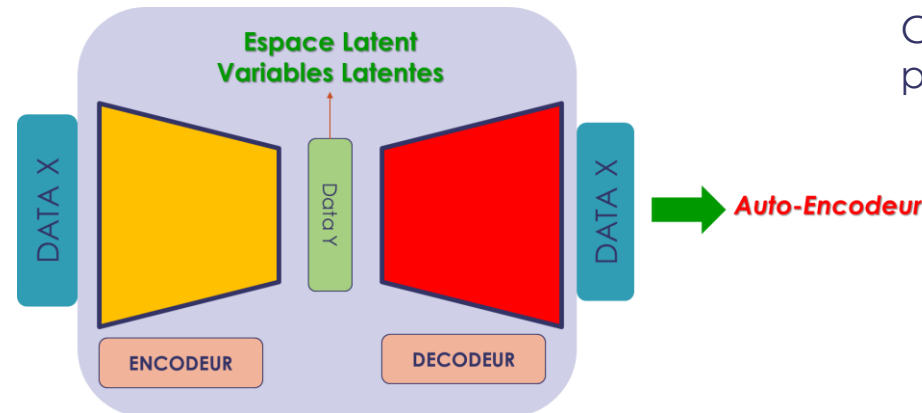
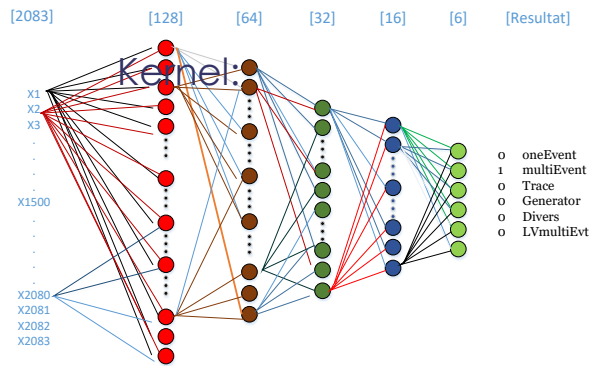


Figure 1: The Transformer - model architecture.



Couche mise à plat (Vecteur)

Couche MLP Dense Layers

