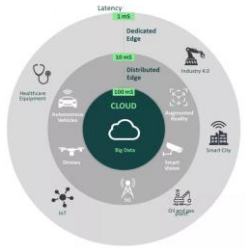
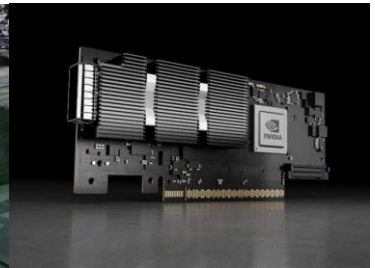
Frédéric Druillole – LP2I Bordeaux IN2P3 CNRS

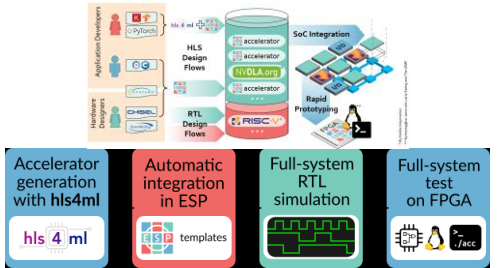
L'intelligence Artificielle dans les systèmes embarqués
– Principes – Aspects Matériels – Mes recherches–
– THINK –

Test of Hardware Inferring Neural networK

Reseau DAQ – Hardware-Firmware-Software-







Conclusion Phase 1

1 Mparam

Edge AI

ESP32

max78000

stm32

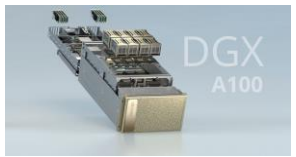
Embedded system

FPGA/ SoC

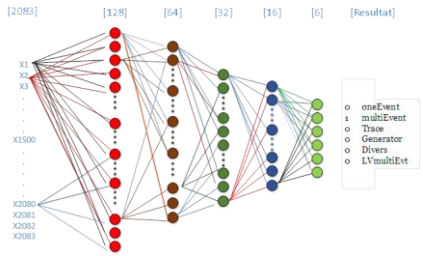
Puce neuromorphique

1 Gparam

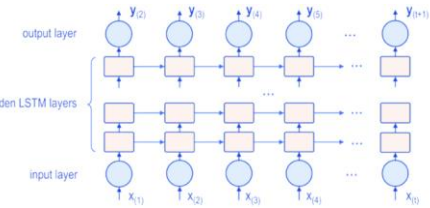
GPU



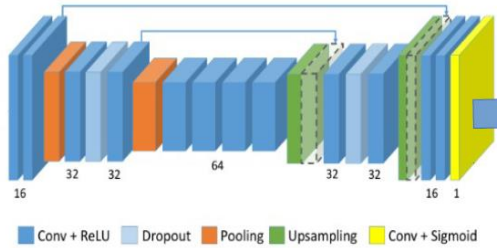
DEEP NEURAL NETWORK

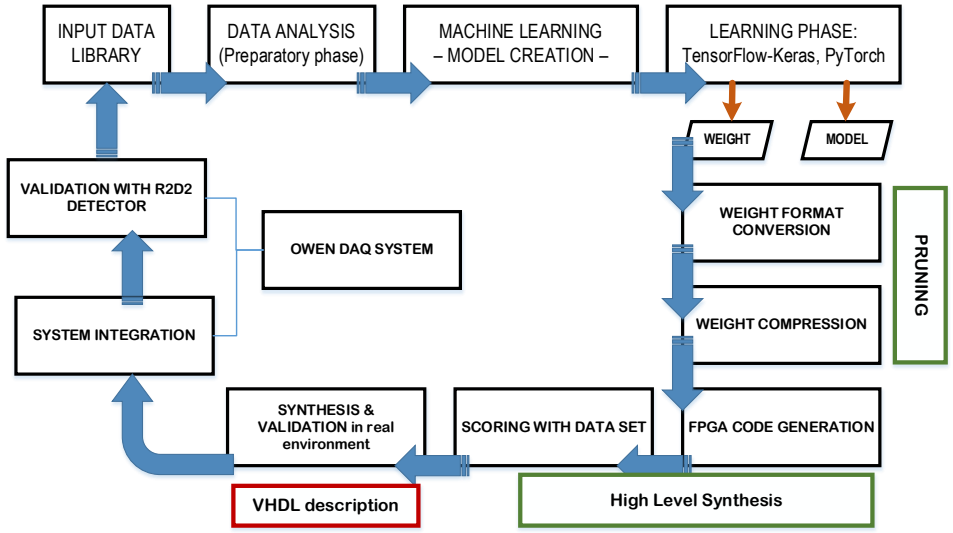


RECURRENT NEURAL NETWORK



CONVOLUTIONAL NEURAL NETWORK





Elagage (prunnig) : réduction du nombre de neurones

Quantization: Reduction de l'empreinte en mémoire (codage des coefficients et des données)

IA embarqué : 2 solutions

Spatial Accelerators

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

ASIC
Circuit neuromorphique
FPGA

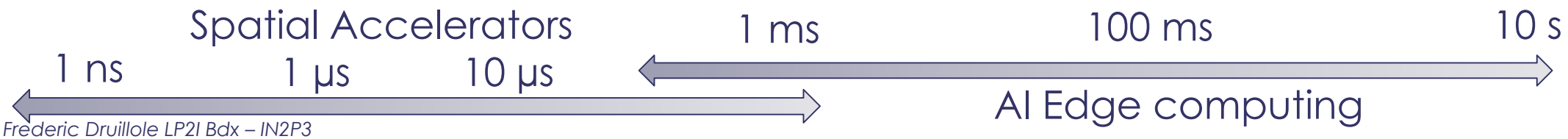
2 En cours de développement

Edge computing

Reste basé sur de la programmation logicielle

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

1 Pilote les développements actuels



Les Frameworks d'inférence des modèle d'IA

AI Edge computing

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

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

Spatial Accelerators

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

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

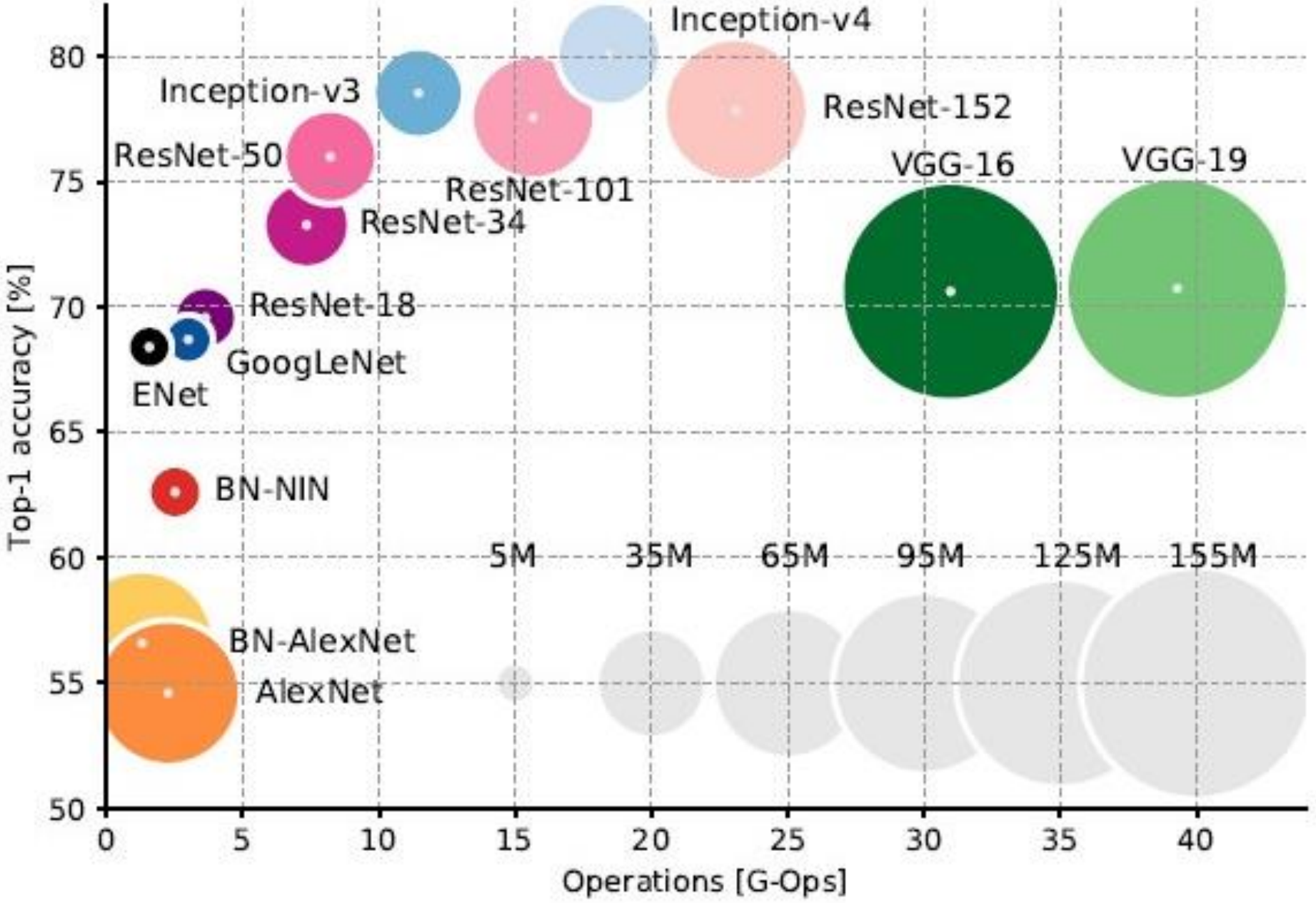
Framework servant à réaliser l'accélération des calculs des modèles IA pour l'inférence.
Utilise des modèles éprouvés
Difficile de créer des inférences sur des composants à faibles ressources (multi-voies, proche des capteurs)

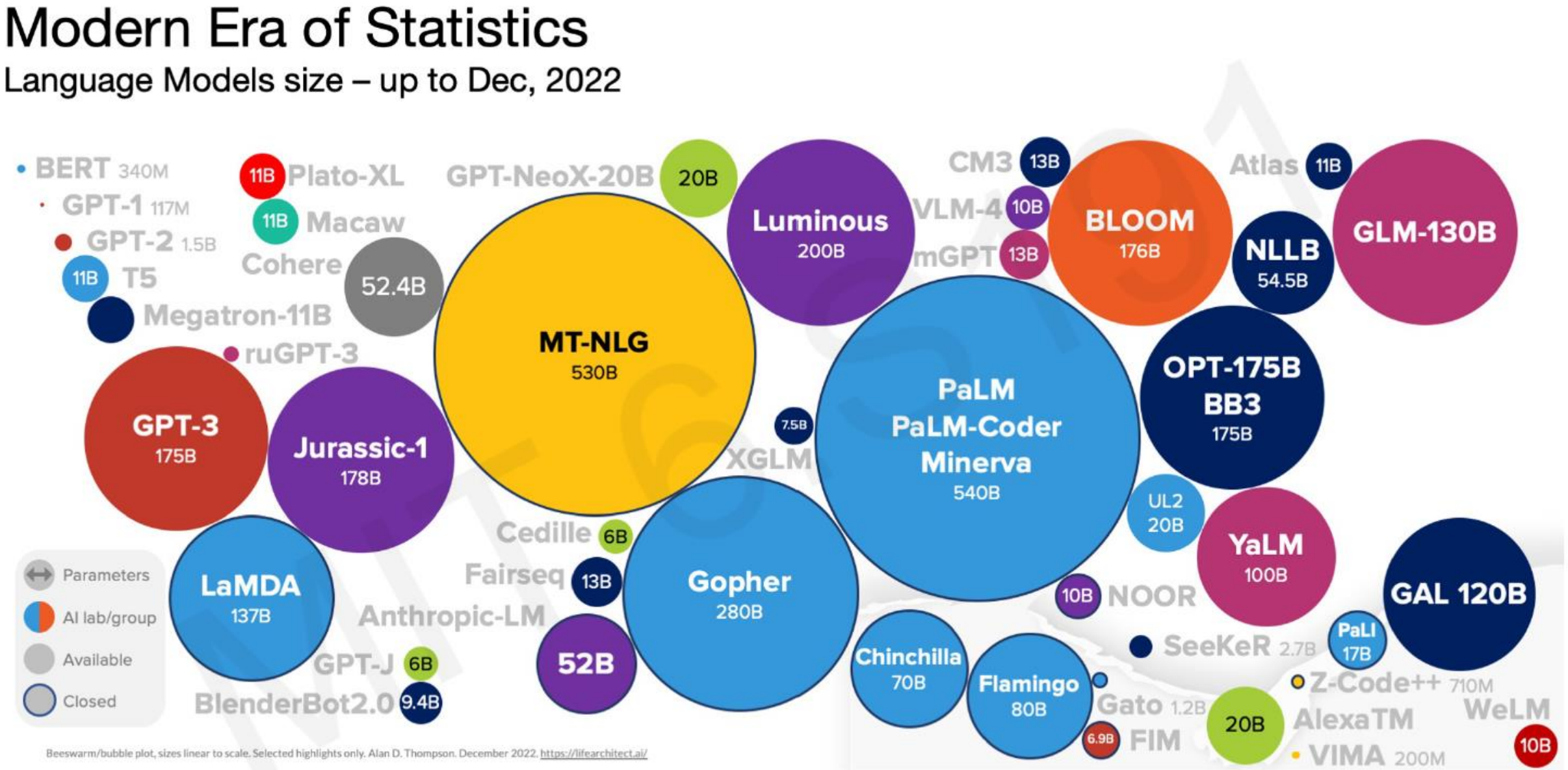
Création d'un framework dédié à Xilinx et Intel en VHDL

Critères de comparaison des matériels pour l'IA

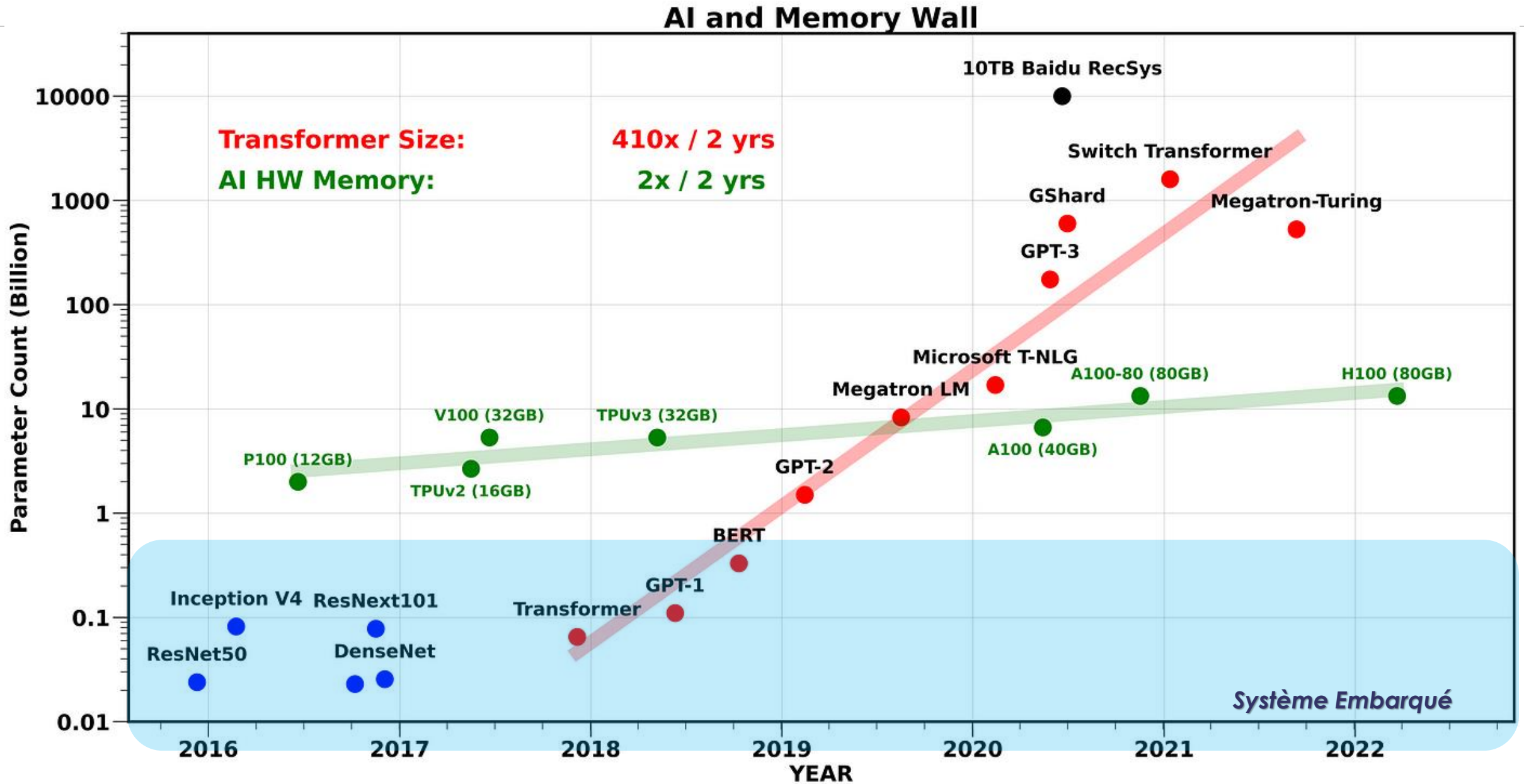
Critères	CPU	CPU+AI-CoPro	GPU	FPGA	ASIC
Adaptabilité (à une variété de modèles)	**Elevé**	**Elevé**	Moyen	**Faible**	Aucune
Puissance de calcul	**Faible**	Moyen-Elevé	**Elevé**	**Elevé**	Moyen
Latence	Moyen (ms)	Moyen (ms)	**Faible** (µs)	**Faible**(10 ns)	**Très faible**
Flux d'entrée	**Faible**	Moyen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

Les autres réseaux : Performances (avant 2020)

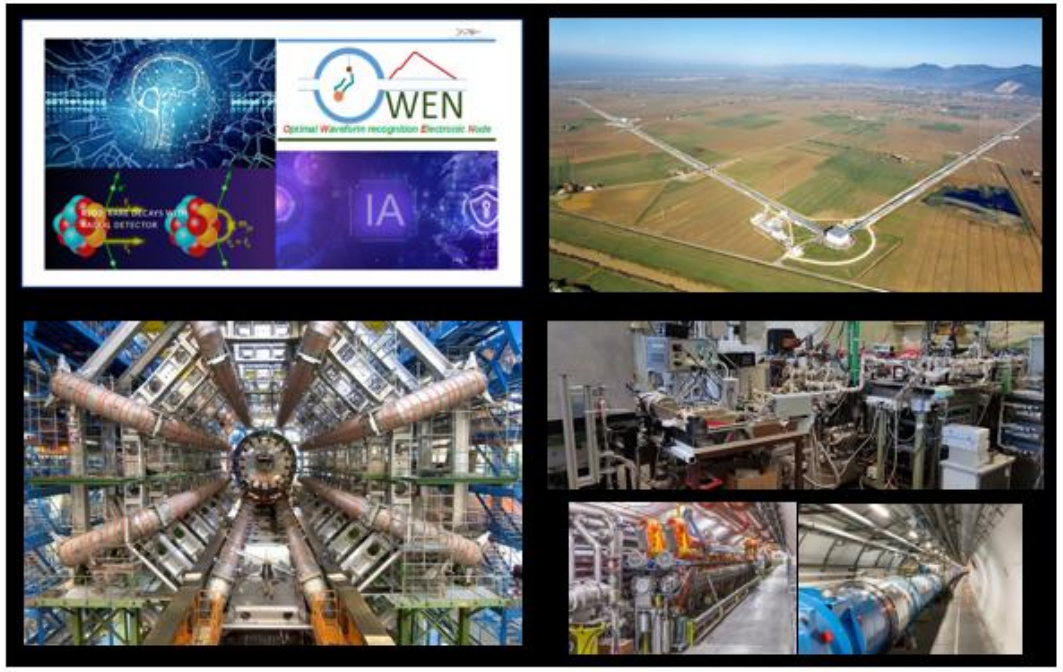



Modern Era of Statistics
Language Models size – up to Dec, 2022
BERT 340M
GPT-1 117M
GPT-2 1.5B
11B T5
Megatron-11B
11B Plato-XL
11B Macaw
Cohere
52.4B
ruGPT-3
GPT-3 175B
Jurassic-1 178B
LaMDA 137B
GPT-NeoX-20B 20B
MT-NLG 530B
Luminous 200B
CM3 13B
VLM-4 10B
mGPT 13B
BLOOM 176B
Atlas 11B
NLLB 54.5B
GLM-130B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

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



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.



Laboratoire Leprince-Ringuet — *F. Magniette, M. Melennec*

CENTRE DE PHYSIQUE DES PARTICULES DE MARSEILLE CPPM — *G. Aad, R. Bertrand*

IPHC Institut Pluridisciplinaire Hubert CURIEN STRASBOURG — *S. Viret, G. Galbit, Q. David (2025), X. Chen*

LP2i Bordeaux — *F. Druillole, B. Mansoux, J. Domange*

iP2i LES 2 INFINIS LYON — *S. Geiger, W. Perrin, M. Fotzé, J. Wurtz*

Conclusion Phase 1

1 Mparam

Edge AI

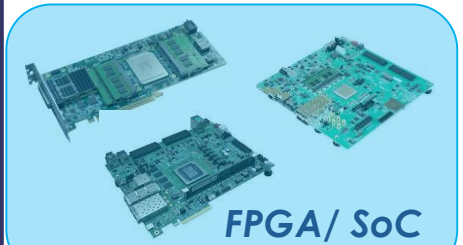
ESP32

Think2-1erAnnée

max78000

stm32

Embedded system



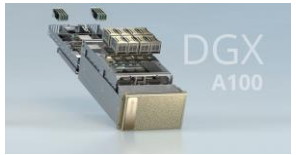
FPGA/ SoC

Puce neuromorphique

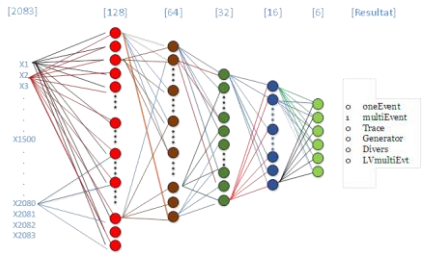
➔ Plateformes utilisées

1 Gparam

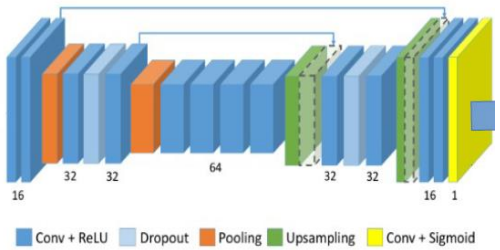
GPU



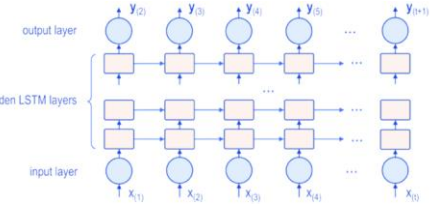
DEEP NEURAL NETWORK



CONVOLUTIONAL NEURAL NETWORK

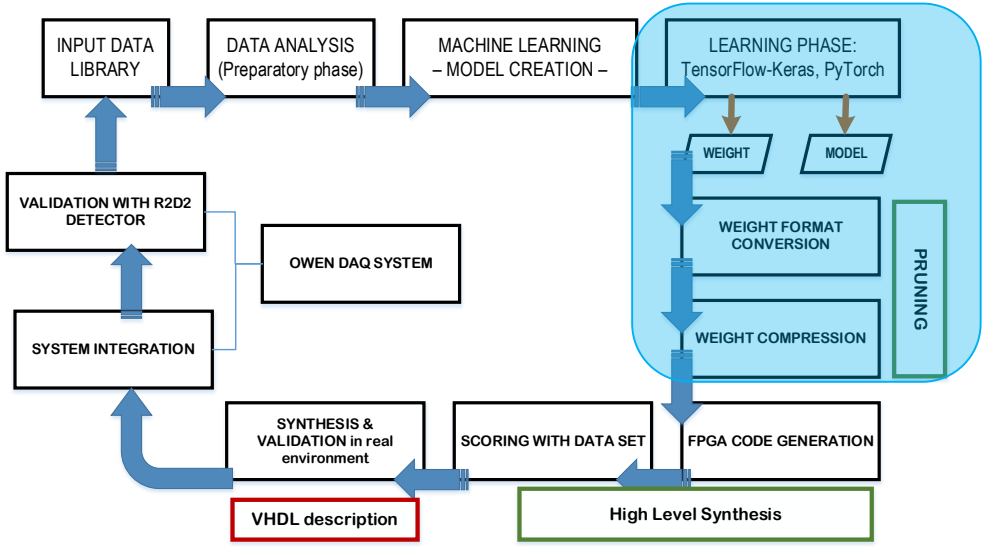


RECURRENT NEURAL NETWORK



Think Phase 2

➔ Optimisation des modèles



Thèmes	Responsable	Labos
Les particules de hautes énergies	George Aad	CPPM Presentation AG DI2I 2024
La physique des neutrinos	Fréderic Druillole	LP2I Bordeaux Presentation AG DI2I 2025
Les ondes gravitationnels	Sébastien Viret	IP2I Lyon Presentation AG DI2I 2025
L'instrument optimisé	Jocelyn Domange	LP2I Bordeaux

Statut THINK2 Presentation Journées des Metiers de l'electronique Juin 2025 Strasbourg

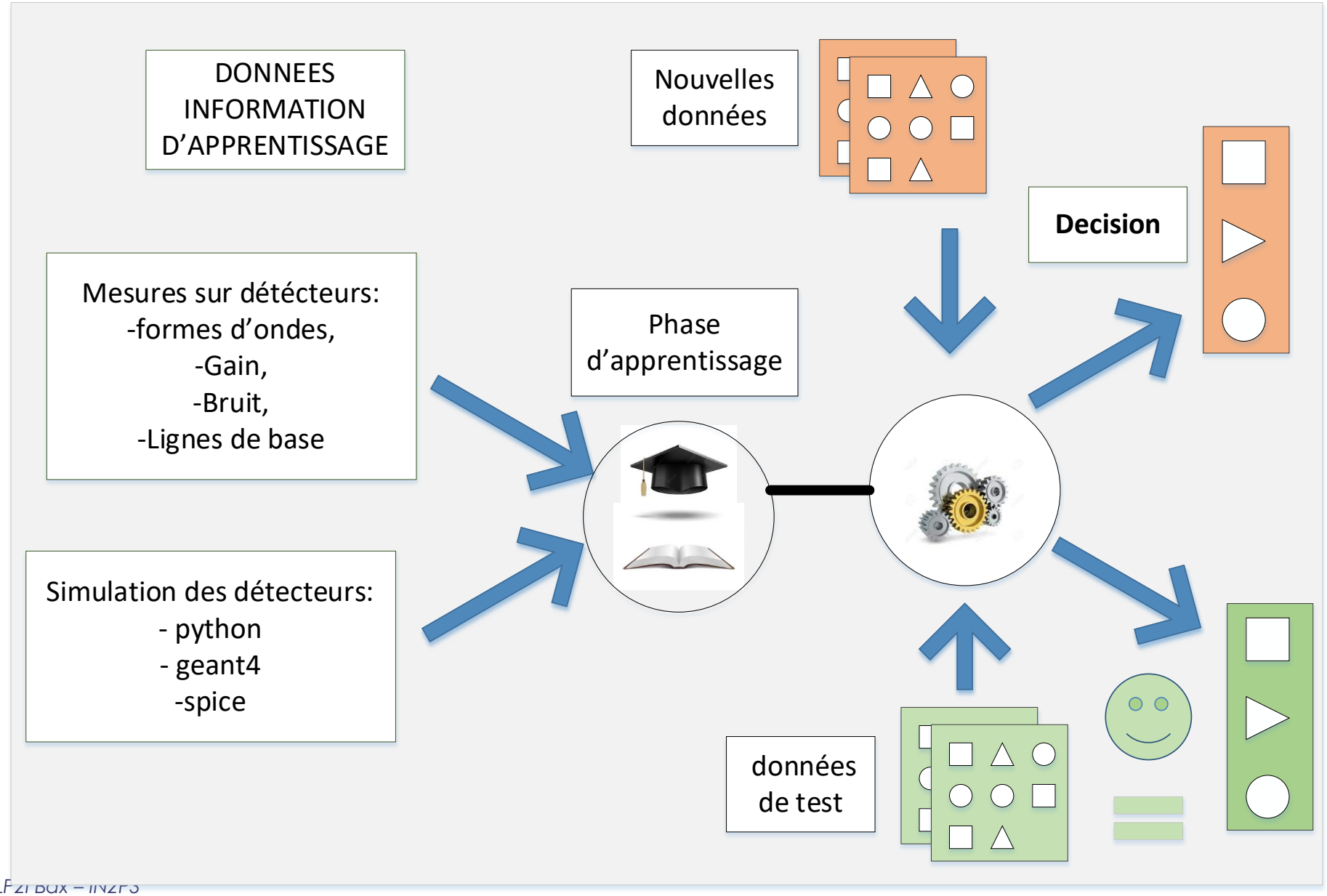
Actions entreprises dans THINK2

- Veille technologique
  - µProcesseurs et coProcesseur IA
  - RISC-V et IA
- Méthode d'optimisation des réseaux par optimisation baysienne:
  - Tutorial fait par Frédéric Magniette
  - Test applicatif au CPPM et IP2I
- Stratégies d'entrainement
- Travail sur le LAr d'Atlas
  - Amélioration des mesures avec presence de PileUp
    - G. Aad et R. Bertrand
    - Concours IR CPPM
- Travail sur OG
  - Optimisation du VHDL par python (G. Galbit DI2I)
  - Modèle Liquid Neural Network (S. Viret)

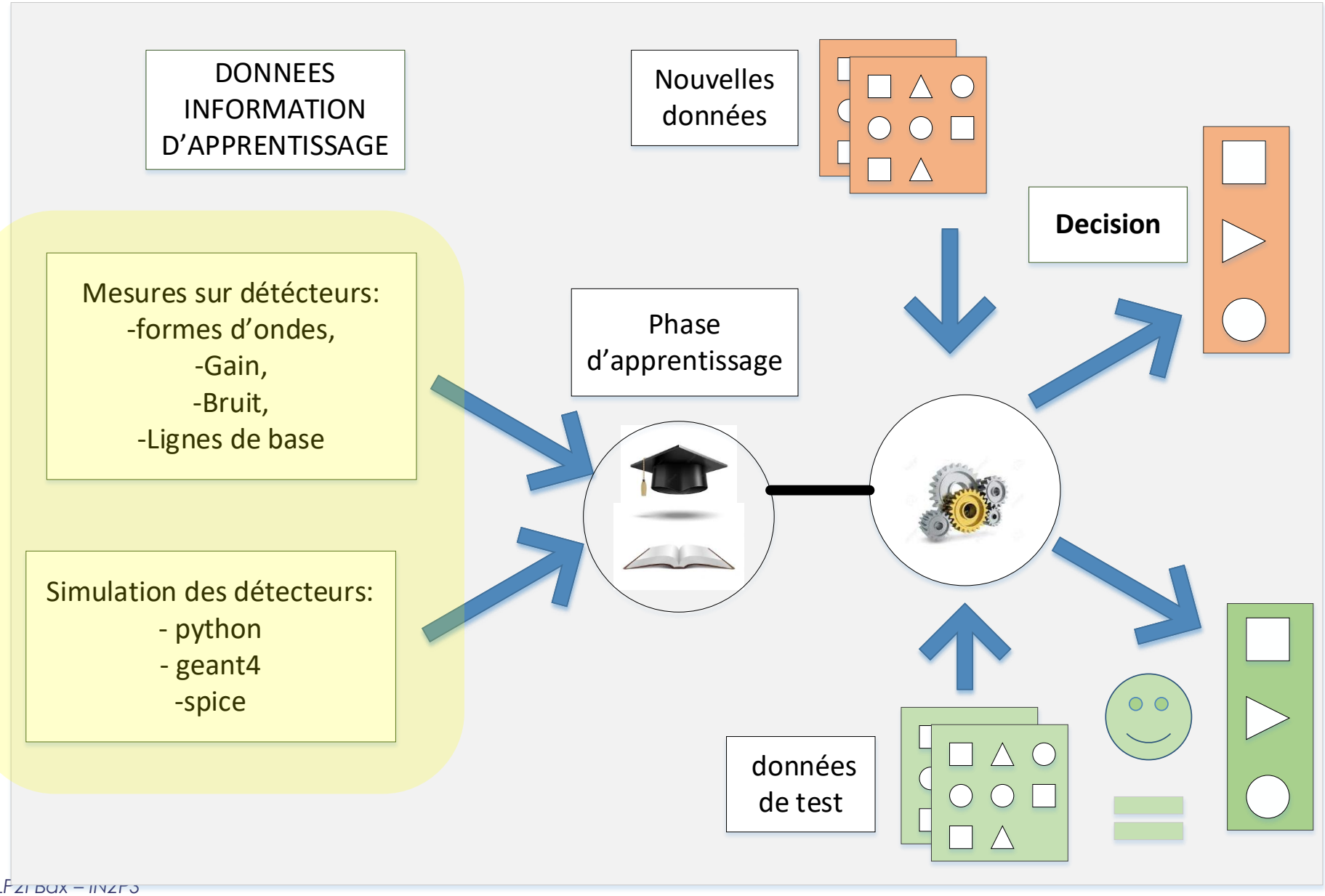
Actions entreprises dans THINK2

- Travail sur R2D2 (Neutrino)
  - Modele ANN avec des mesures
  - Génération de signaux par VAE et GAN
  - Création de nouveaux modèles (détection d'anomalie…)
- Travail sur Juno (Muons):
  - utiliser un réseau de neurones pour le trigger de niveau 2 du Top Tracker
  - Détecter des muons
- Creation d'un espace GitLab
  - Stockage des données d'apprentissage
  - Stockage des tutoriaux de modèles (Jupyter lab)
- Diffusion et acculturation
  - Demos
  - Présentation
- Modélisation détecteurs (J. Domange)
  - Données d'apprentissage
  - Prise de données réélles (Chambre à Brouillard, APD, SiPM)

Mes Recherches:

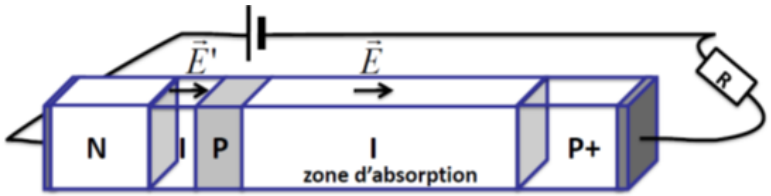


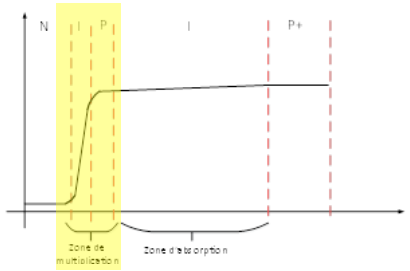
Création de données pour l'apprentissage



Création de données pour l'apprentissage: Modèlisation de détecteur APD en python

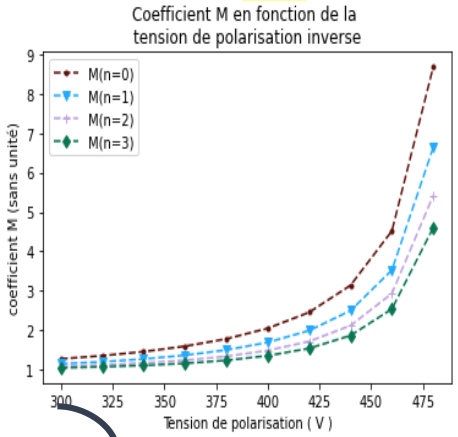
➔ Multiplication des électrons par avalanche du signal de ionisation dû au photon

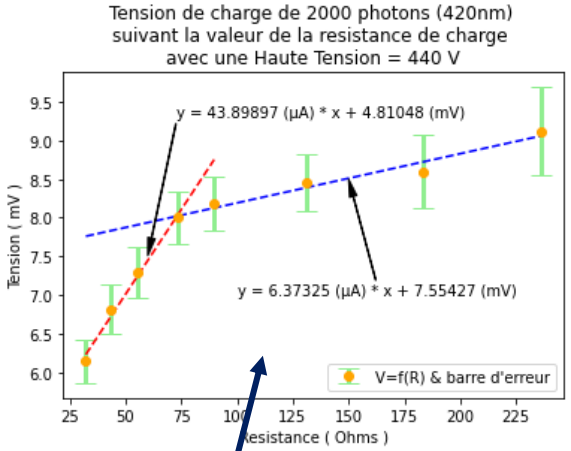




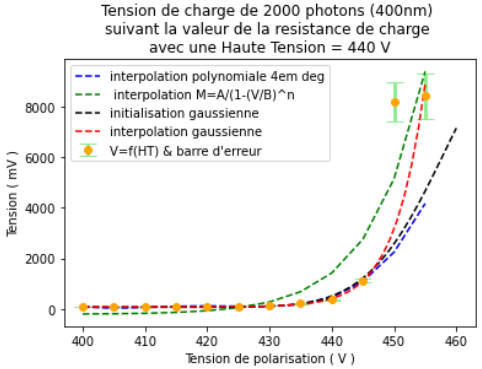
Coefficient de multiplication empirique

$$M = \frac{1}{1 - \left(\frac{V_{bias}}{V_{claquage}}\right)^n}$$



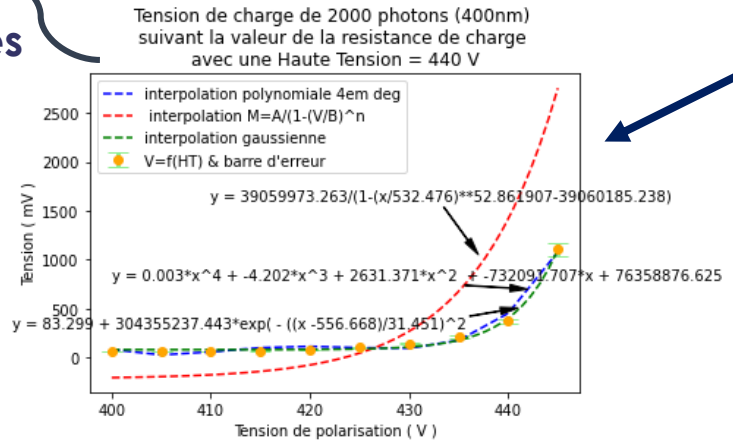


$$I_{ph} = M(V_b).R(\lambda).\frac{N.q}{h.\upsilon}.e^{-\frac{t}{\tau}}$$



Interpolation de M(Vbias) par la montée d'une fonction gaussienne marche mieux.

Mesures

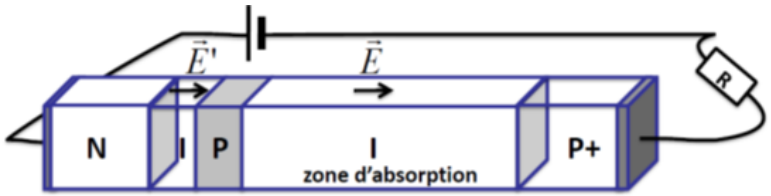


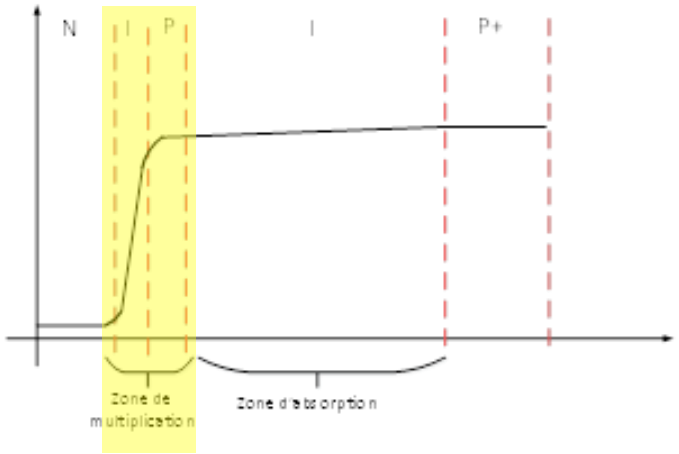
Limitation de la BP par Rcharge:

$T=R_{charge}$ x C_{APD}

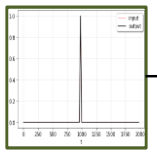
Création de données pour l'apprentissage: Modèlisation de détecteur APD en python

➔ Multiplication des électrons par avalanche du signal de ionisation dû au photon





Modèle complet

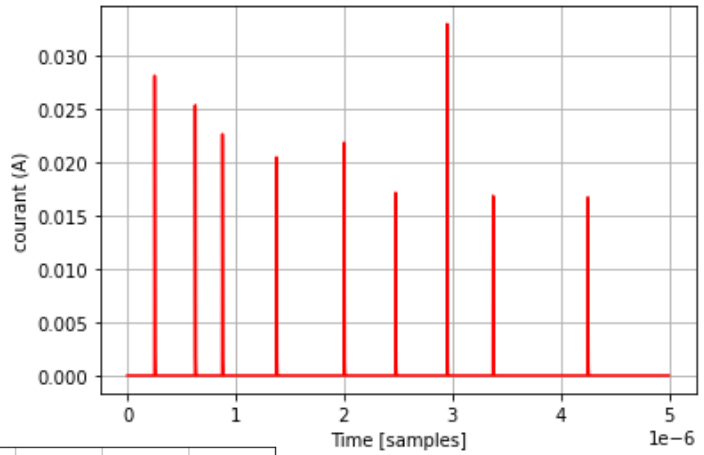


$$q.R.M(V_b).QE(\lambda).\frac{E_{photon}}{E_i.\Delta t}$$

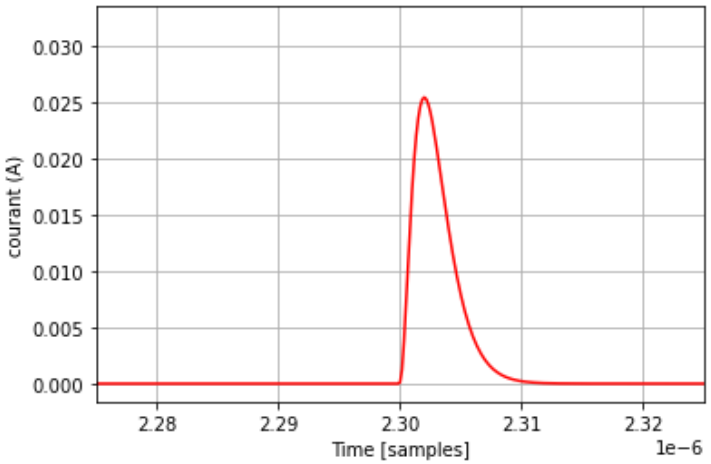
$$\frac{12}{(1+1e-9p)^3}$$

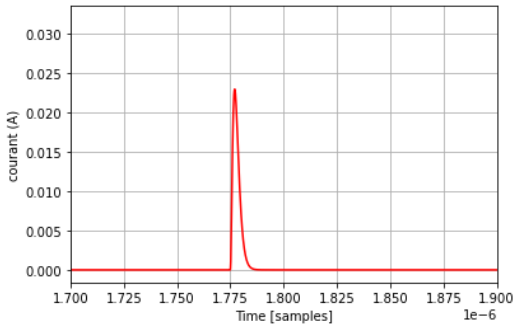
Vph

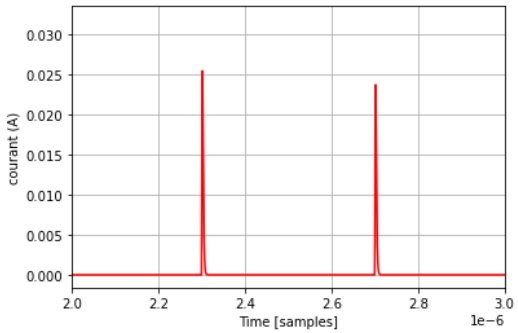
Offset



Temps par tirage poissonnien
Amplitude suivant Energie

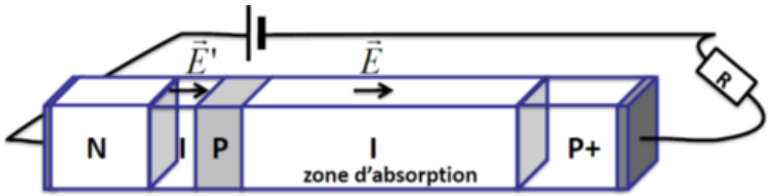


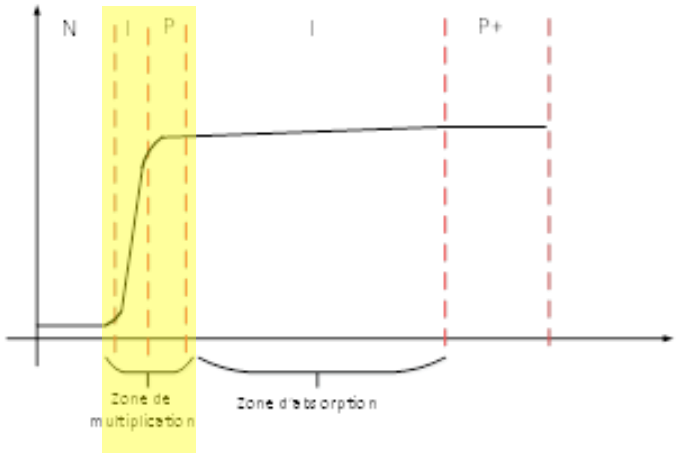


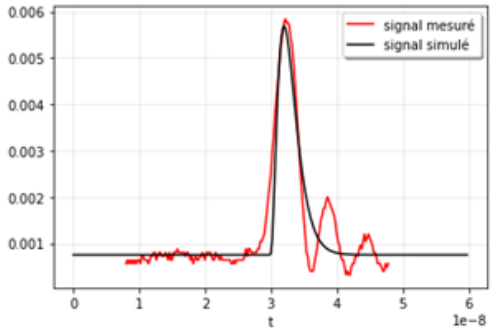


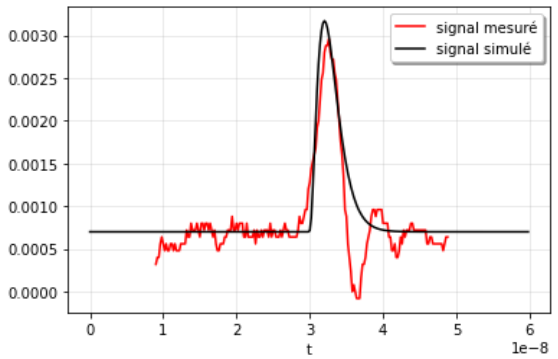
Création de données pour l'apprentissage: Modèlisation de détecteur APD en python

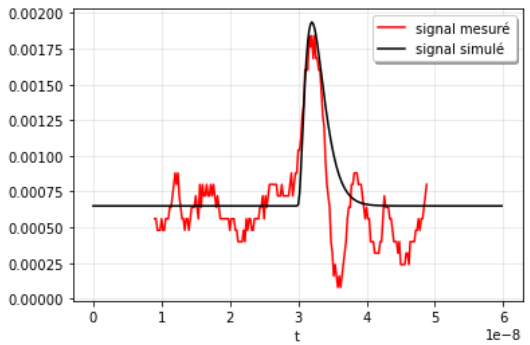
➔ Multiplication des électrons par avalanche du signal de ionisation dû au photon



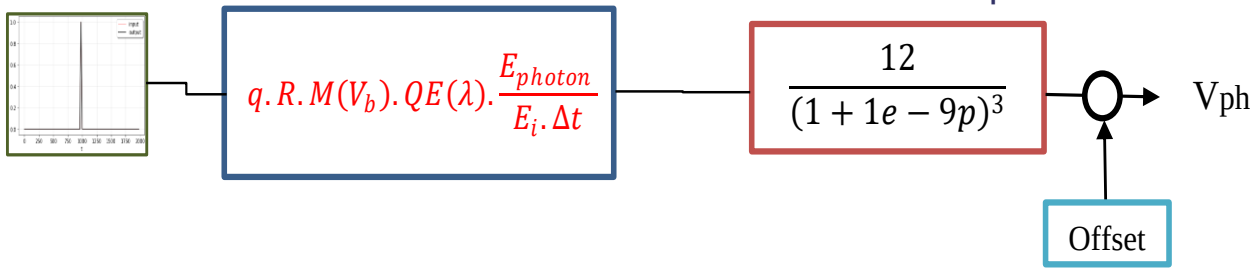


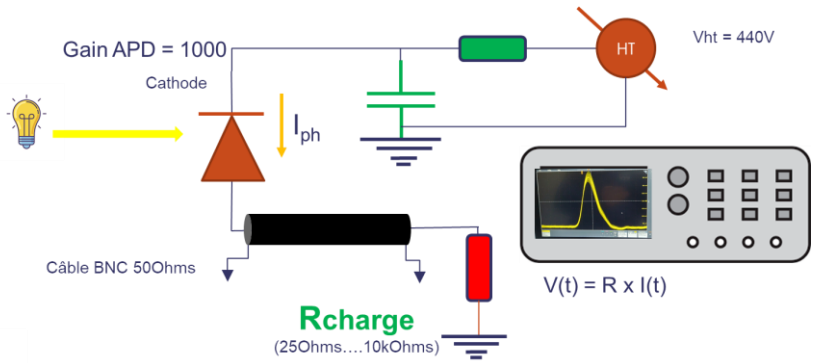






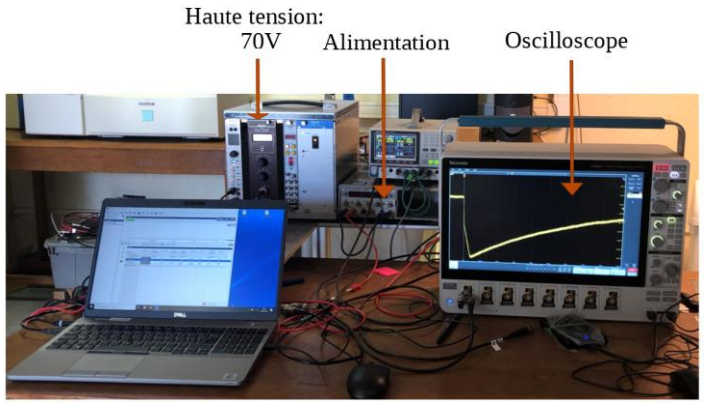
Modèle complet

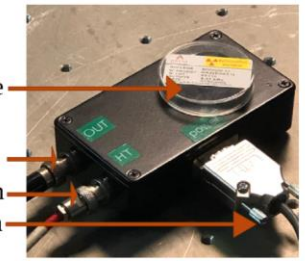


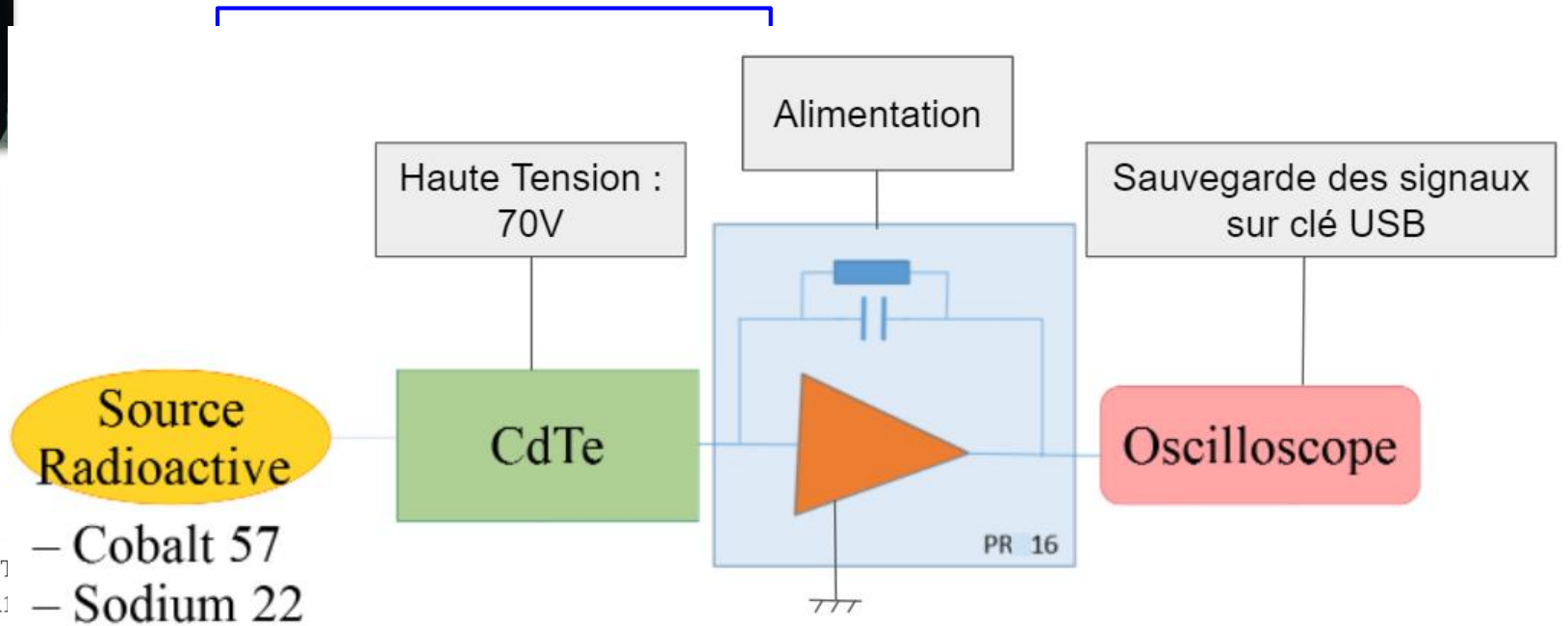


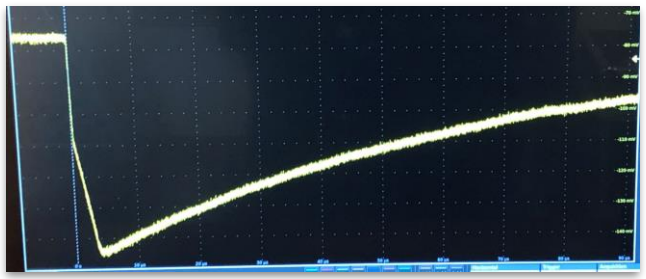
Création de données pour l'apprentissage: Modèlisation de détecteur CdTe en python

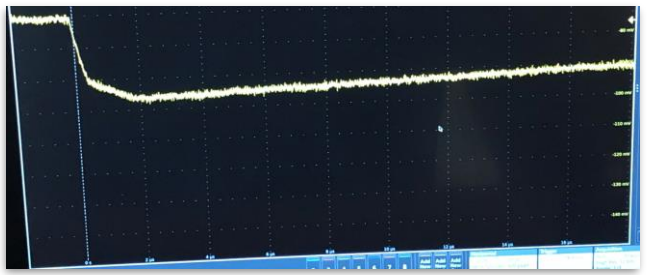
Schéma du montage :

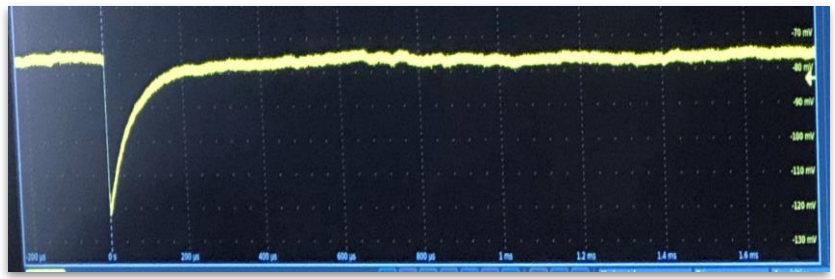












Création de données pour l'apprentissage: Modèlisation de détecteur CdTe en python

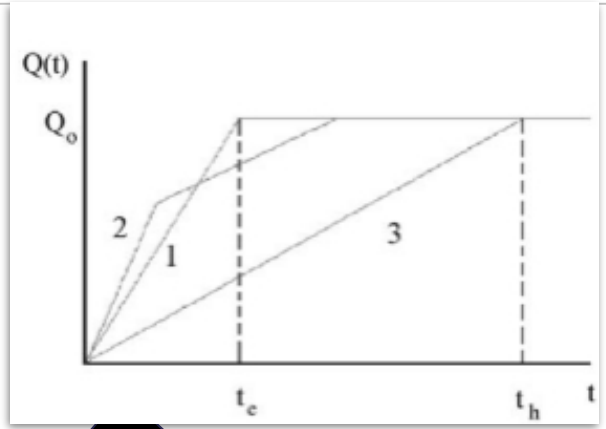
Modèle théorique : Modèle de Hecht

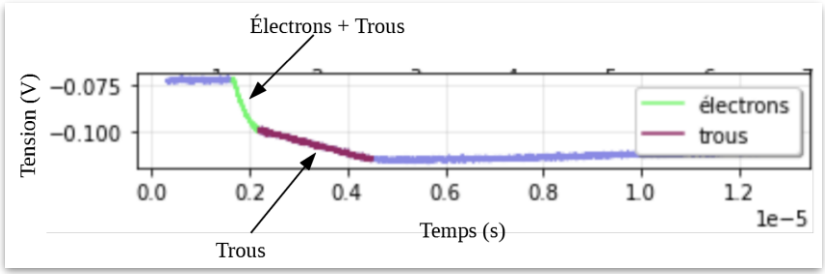
$$Q(t) = \frac{E_p \cdot q}{E_i \cdot L} \cdot \left[\lambda_e \left(1 - e^{\frac{-(L - x_0)}{\lambda_e}} \right) + \lambda_h \left(1 - e^{\frac{-x_0}{\lambda_h}} \right) \right]$$

Charge déposée par le photon détecté par unité de longueur

Partie électrons

Partie trous

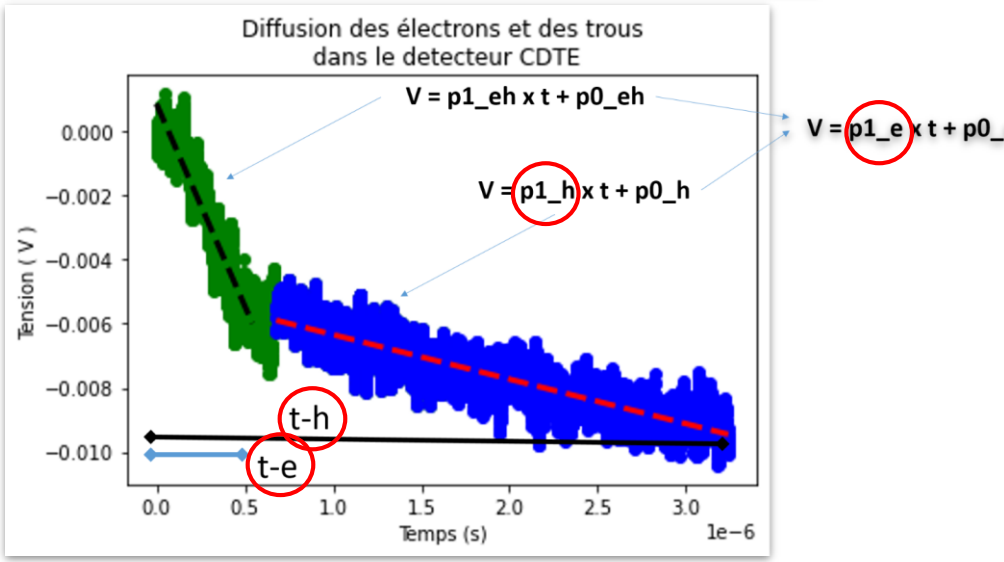




E_p : Énergie émise par la source radioactive (eV)
E_i : Énergie de ionisation du détecteur (4,43eV)
λ_e : Longueur de diffusion moyenne des électrons (m)
λ_h : Longueur de diffusion moyenne des trous (m)
τ_e : Temps de piégeage des électrons (s)
τ_h : Temps de piégeage des trous (s)
μ_e : Mobilité des électrons (m^2/Vs)
μ_h : Mobilité des trous (m^2/Vs)
L : Longueur du détecteur (2mm)
x_0 : Profondeur d'interaction de la particule (m)
q : Charge de la particule (C)

$$\lambda_e = \mu_e \cdot \tau_e \cdot E_{électrique}$$

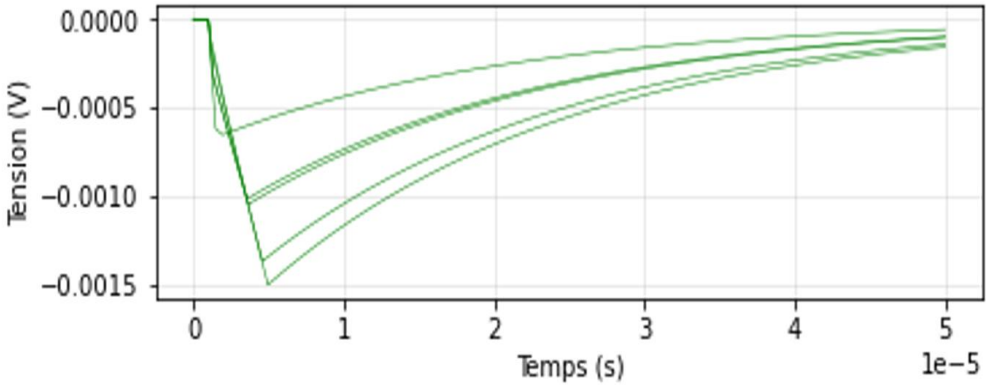
$$\lambda_h = \mu_h \cdot \tau_h \cdot E_{électrique}$$



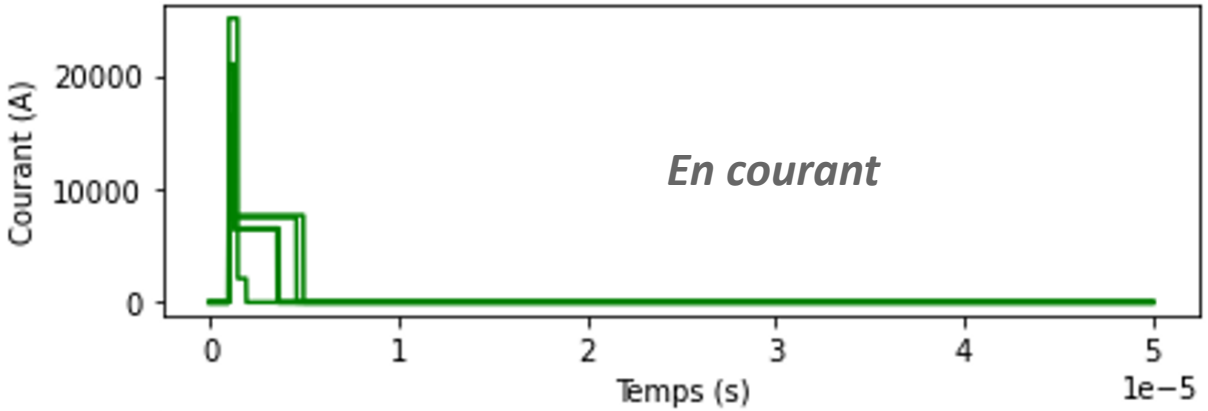
Création de données pour l'apprentissage: Modèlisation de détecteur CdTe en python

- **Résultats : Exemple modélisation de Hecht pour E = 122keV à profondeur de 1 mm :**

– Avec valeurs théoriques :

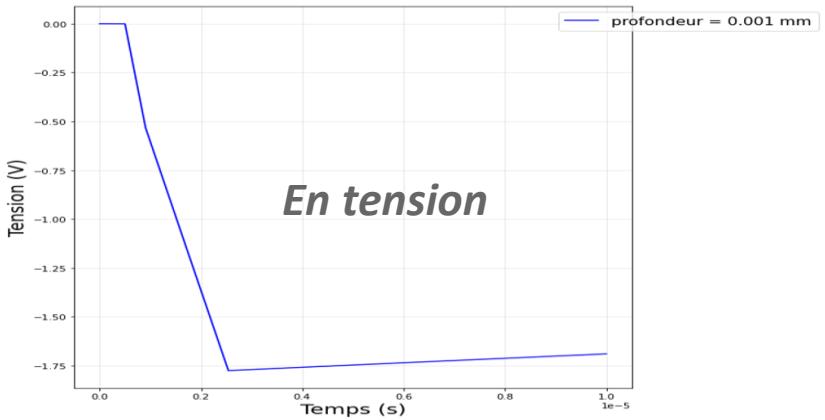


En tension

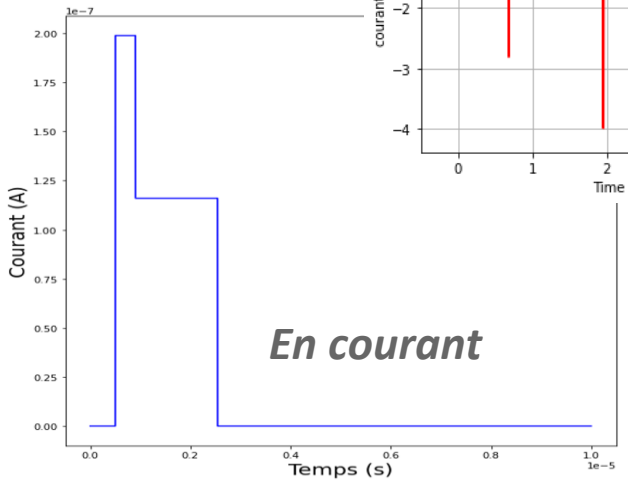


En courant

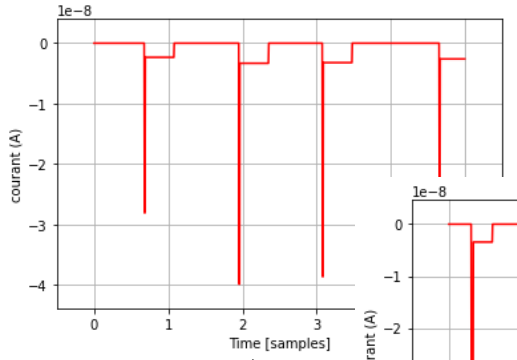
– Avec valeurs expérimentales :

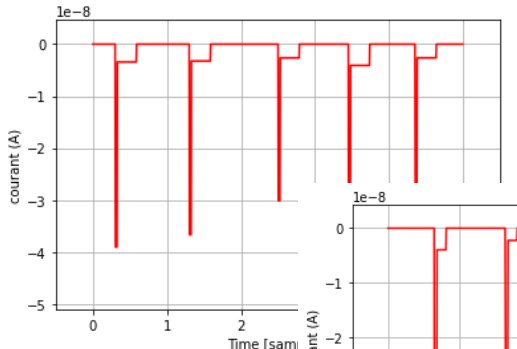


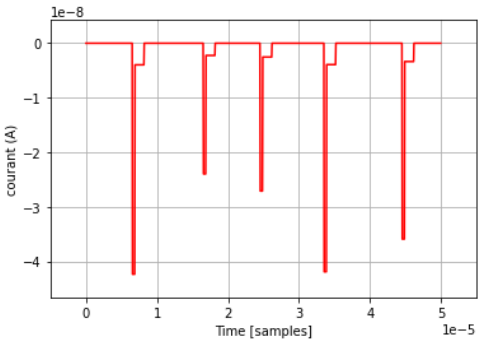
En tension



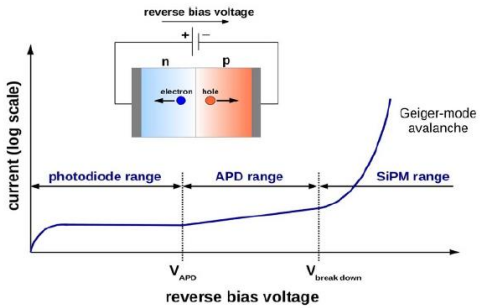
En courant

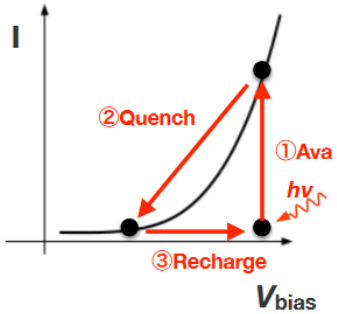


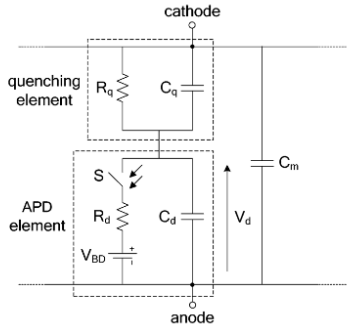


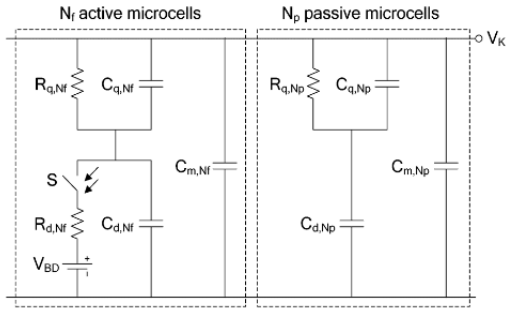


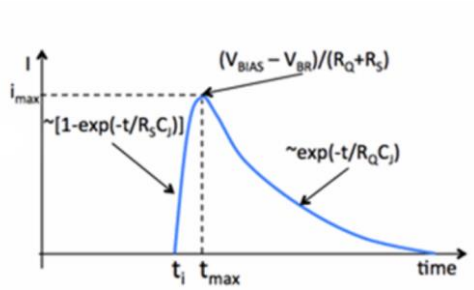
Création de données pour l'apprentissage: Modélisation de détecteur SiPM en python

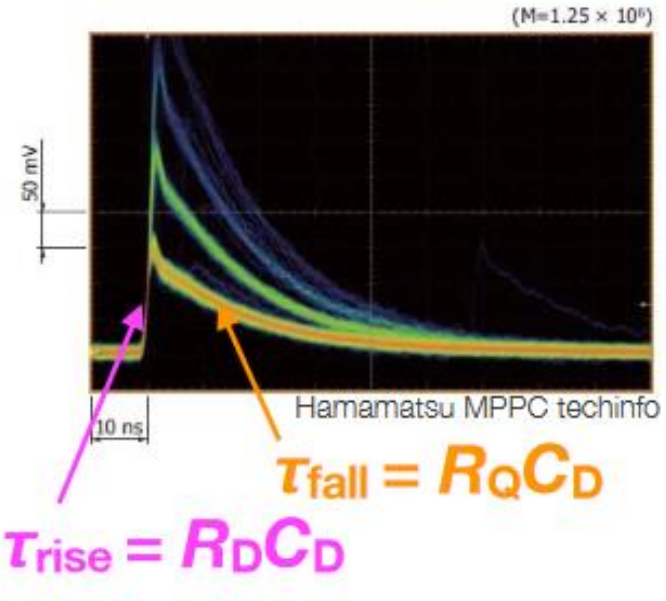








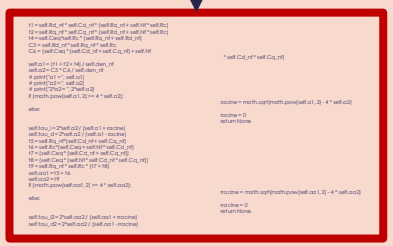


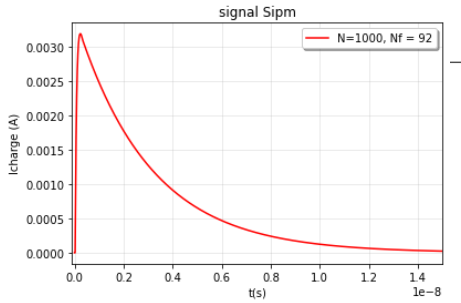


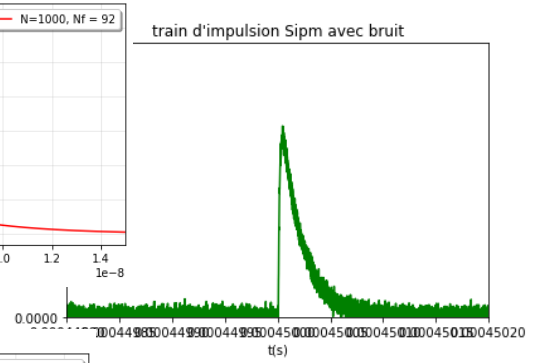

```
sipm_prop.setRd(300)
sipm_prop.setRq(300e3)
sipm_prop.setCd(200e-15)
sipm_prop.setCq(20e-15)
sipm_prop.setCm(5e-15)
sipm_prop.setVov(2)
sipm_prop.setPropSimu('NbreCellules', 100)
sipm_prop.setPropSimu('NbreCelluleDeclenchee', 1)
sipm_prop.setPropSimu('NbreCellules', 1000)
sipm_prop.setPropSimu('NbreCelluleDeclenchee', 1)
sipm_prop.setPropSimu('courantLimite', 100e-6)
sipm_prop.setProperty("temps_de_descente_lent", 3e-9)
sipm_prop.setProperty("tempsRecuperation", 3e-10)
```

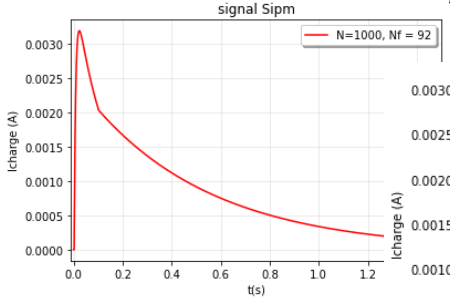
Calcul:
-Ceq = f(Cd, Cq, Cm)
-H_o,-H_{RL}
-I_{RL},-I_{RC}
-I_{peak}

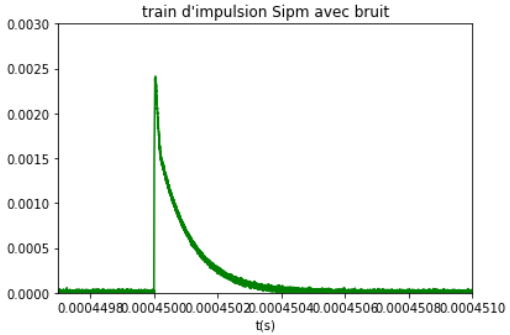
Modèle Signal Analogique

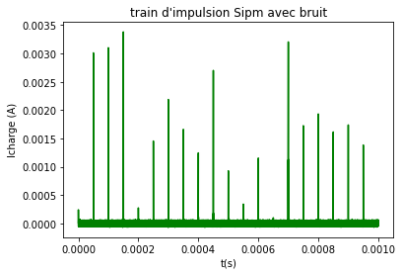






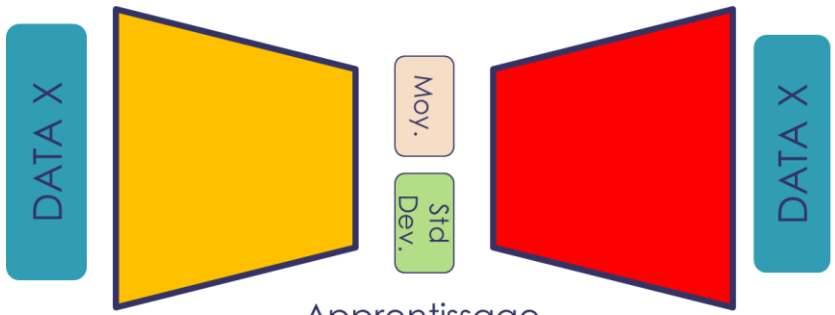


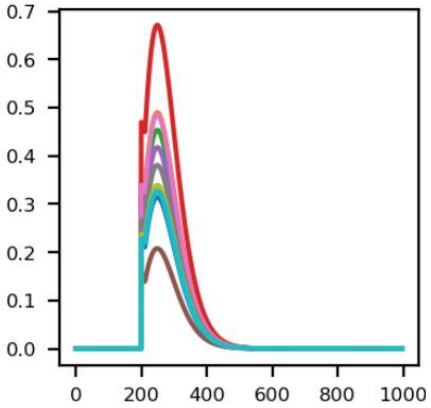




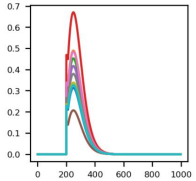
Pipeline de données: Génération de données

Variationnal Auto-Encoder



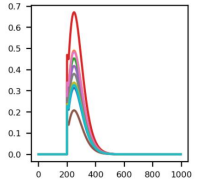


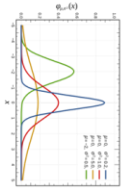
Apprentissage

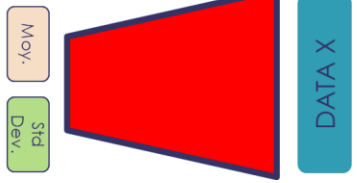


ENCODEUR

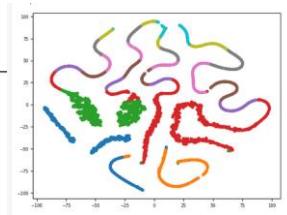
DECODEUR



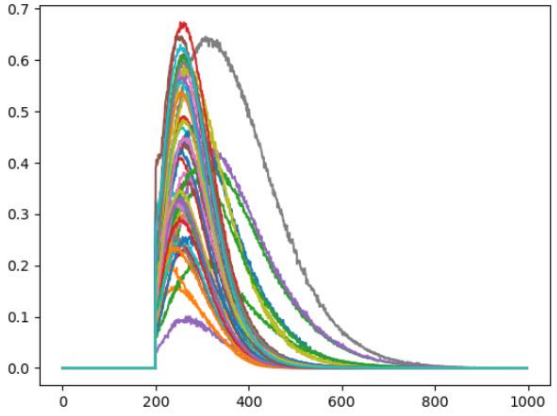




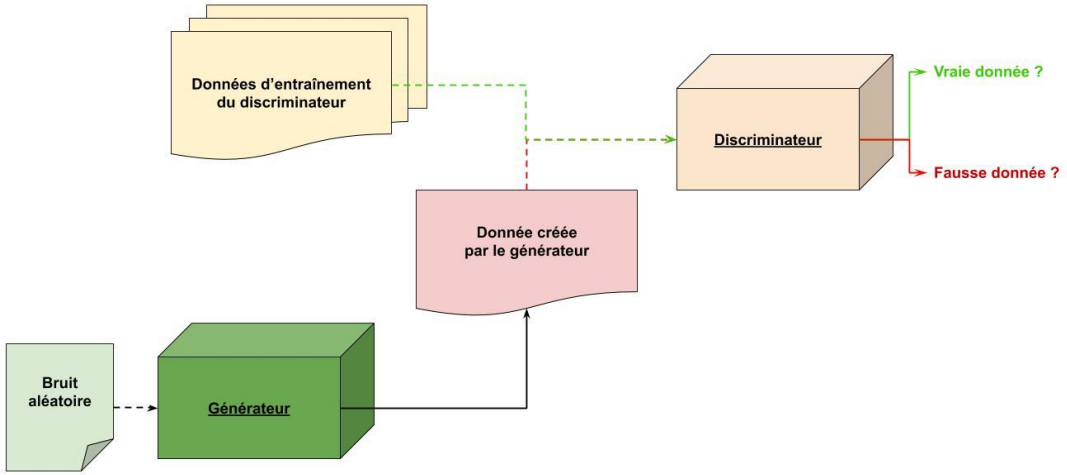
T-SNE

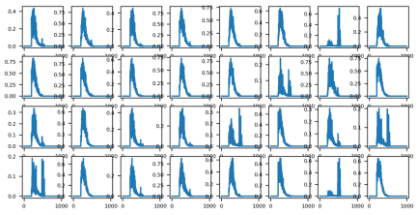


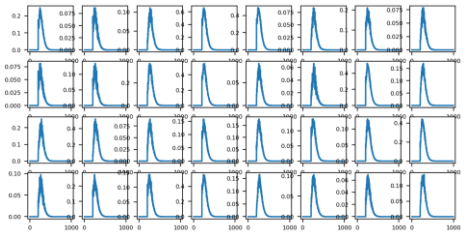
ANN



Generative Adversal network

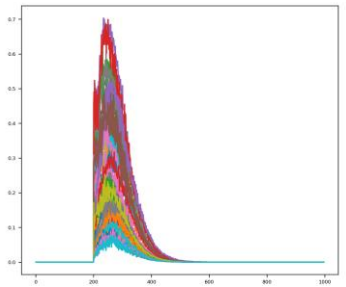




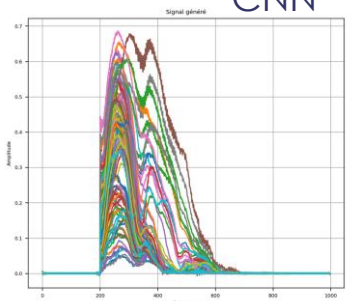


Learning phase

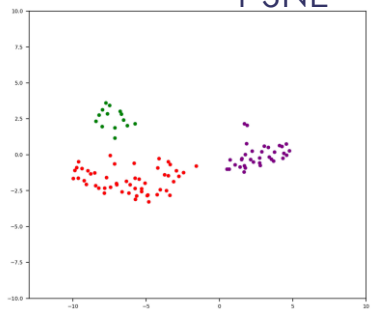
ANN



CNN



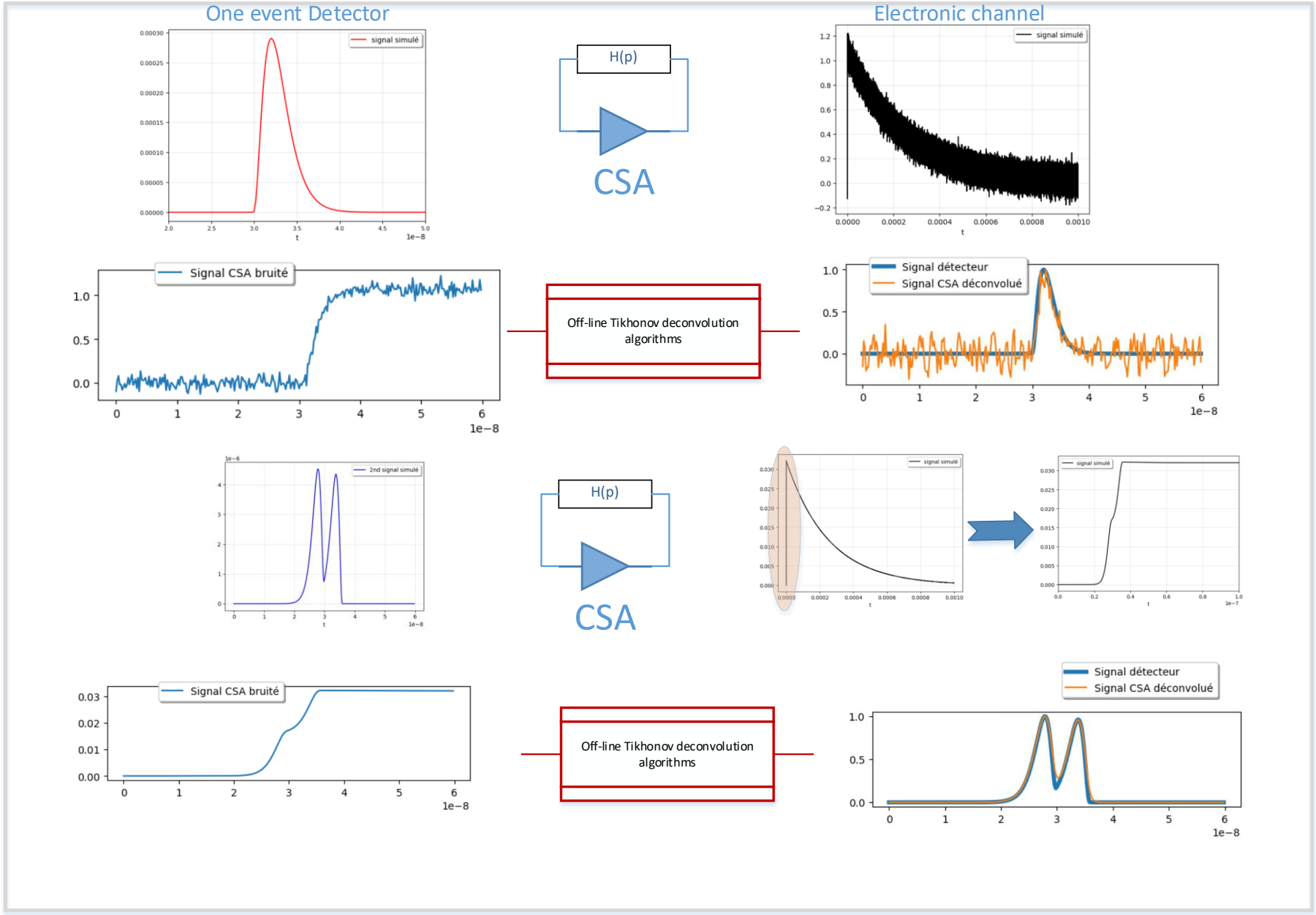
T-SNE



Déconvolution par algo de Tikhonov (Off-Line)

Problème mal posé au sens d'Hadamard:

➔ Algorithme Off-Line

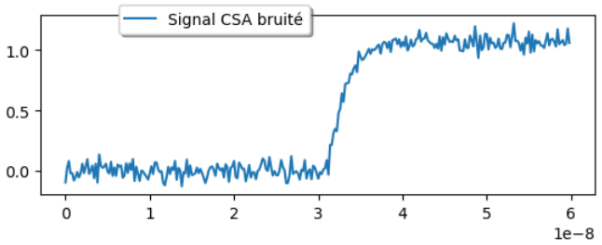


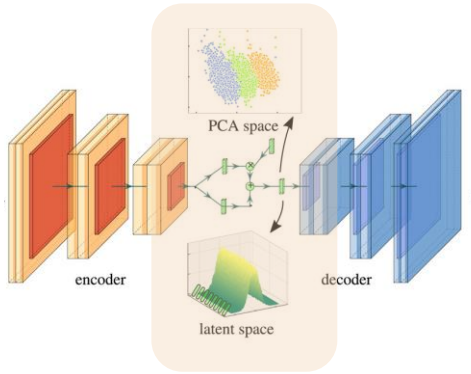
Déconvolution par IA: Travail en cours

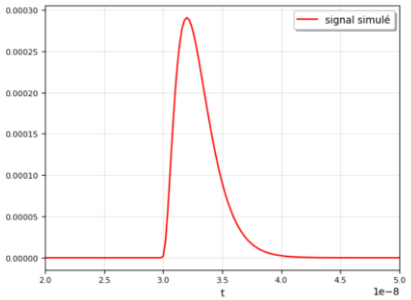
Problème mal posé au sens d'Hadamard:

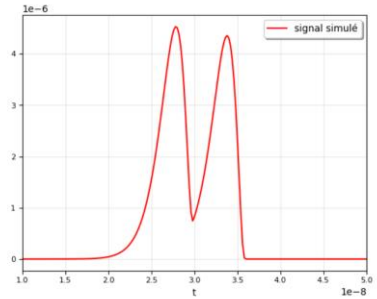
➔ Algorithme On-Line

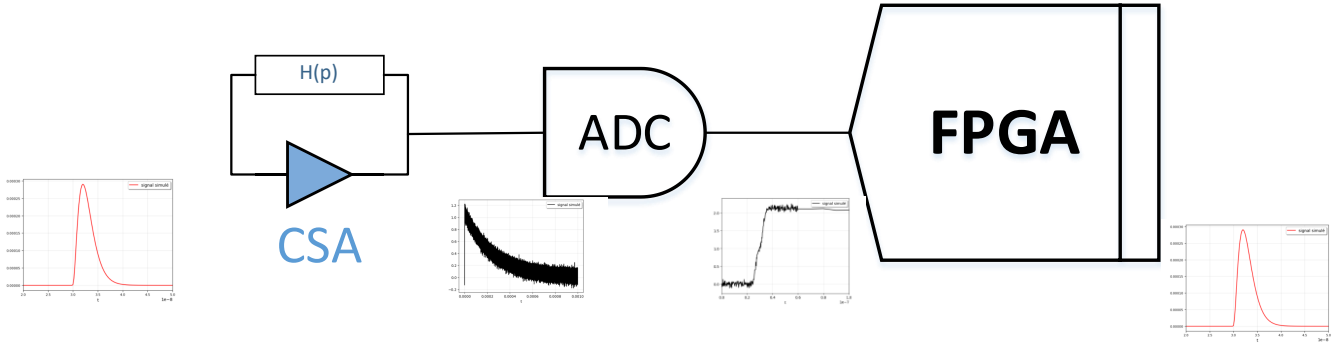
Auto-encoders denoising and deconvolution





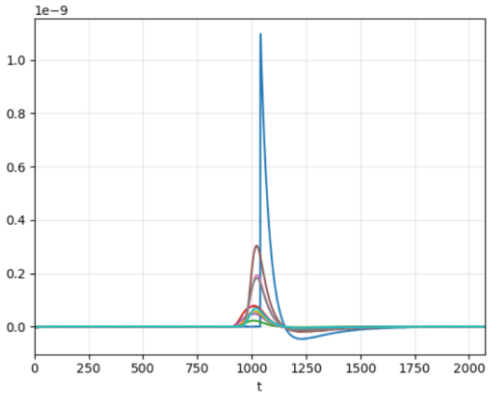




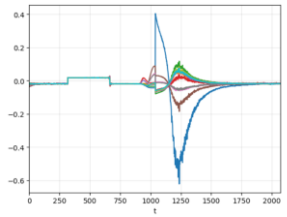


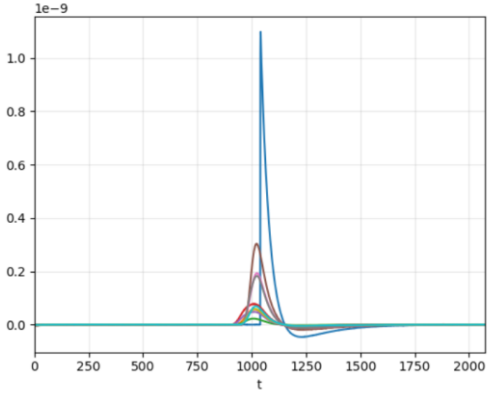
Détection d'anomalie

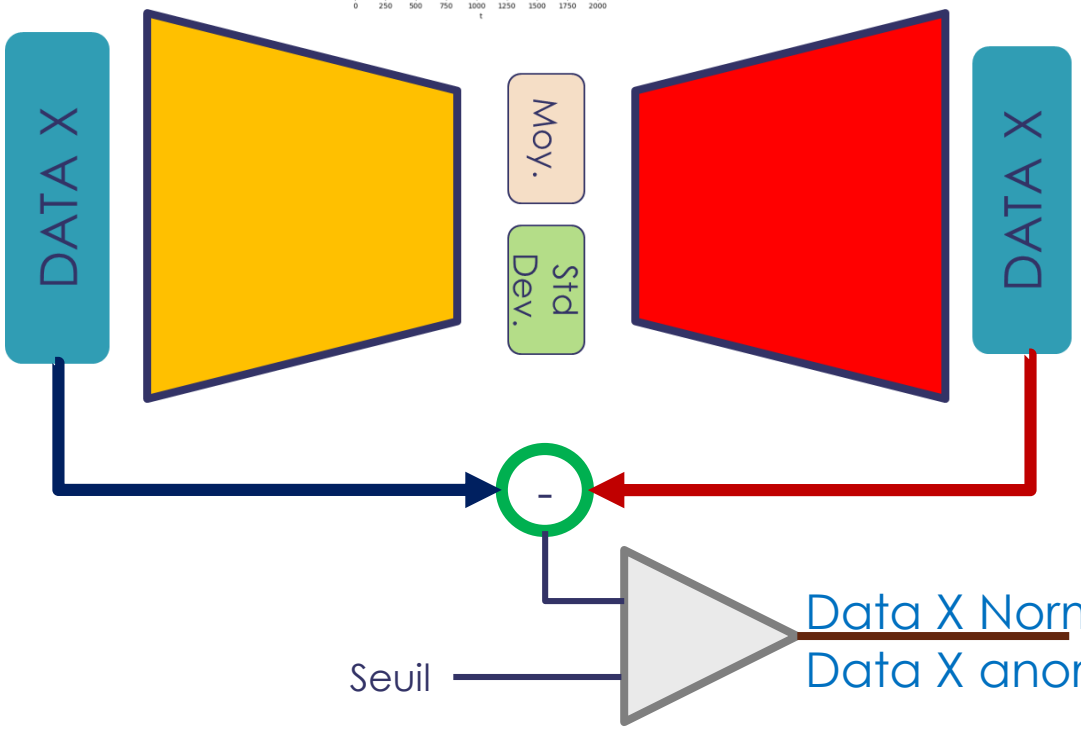
Inférence avec les vrai mesures

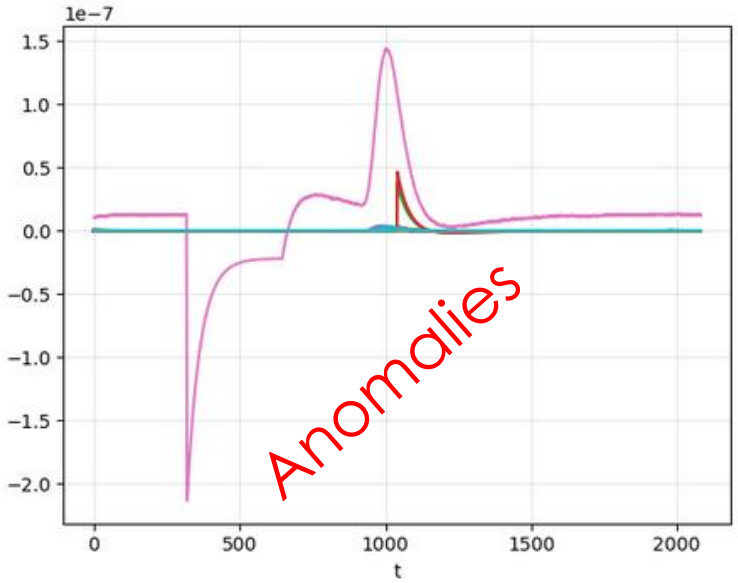


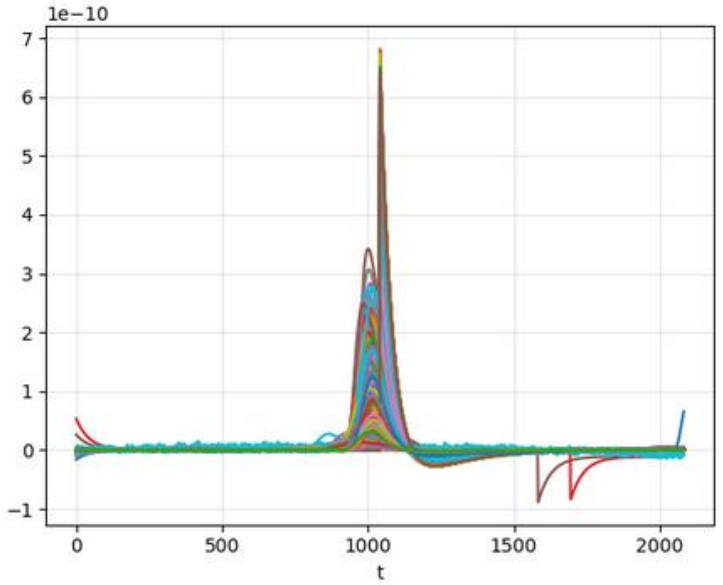
Apprentissage











Conclusion

- J'ai créé des modèles de détécteurs pour préparer les données d'apprentissage
  - APD
  - CdTe
  - SiPM

- J'ai testé des modèles d'IA pour:
  - Générer des données par IA
  - Classifier des signaux
  - Détecter des signaux anormaux

- J'ai étudié le principe de la déconvolution par IA

→ Développement des applications SID (système intelligent pour détecteur)
→ Mettre en œuvre les codes, les données pour une application réelle:
  → R2D2: signaux double béta
  → Optimisation de spectre par déconvolution
  → Amélioration du rapport signal sur bruit sur SiPM et APD.

Conclusion

- **Utiliser de l'IA pour le traitement des signaux de détecteurs, c'est:**
  - **Connaitre la physique de création des signaux dans le détecteurs**
  - **Modèliser et simuler la création de signaux en sortie de détecteurs**
  - **Ajouter la chaîne électronique complète**
  - **Effectuer des vraies mesures pour confirmer les formes des signaux.**

- On peut appliquer les réseaux de neurones pour:
  - Générer de plus grandes quantités de données
  - Avoir plus de diversités dans les données générées
  - Créer des fonctions (approximateur universel) pour:
    - Trouver les données anormales
    - Catégoriser les évenements
    - Élaborer de nouvelles fonctions de transfert à partir des données

- Inférer des réseaux de neurones au plus près des détecteurs nécessite:
  - Une méthode d'optimisation des modèles (elagage, quantisation)
  - Un savoir faire en IA (experiences/ exploration)
  - Une connaissance des composants et de leur SDK