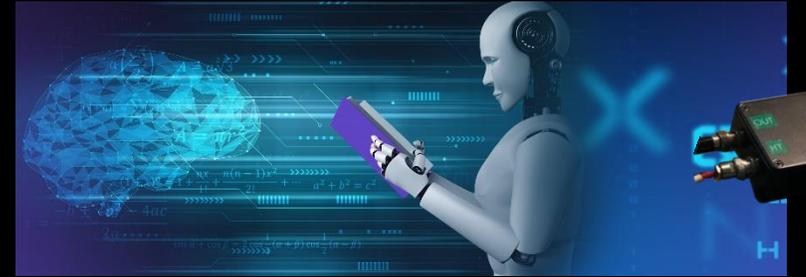
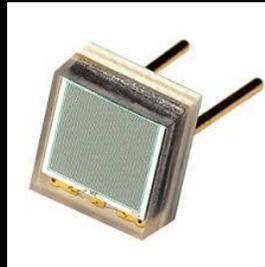
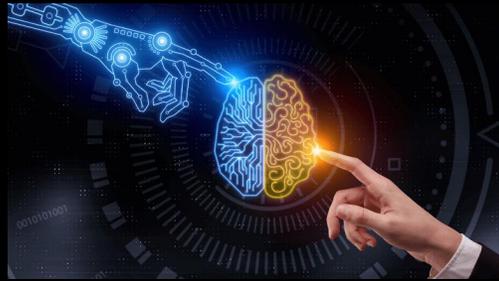
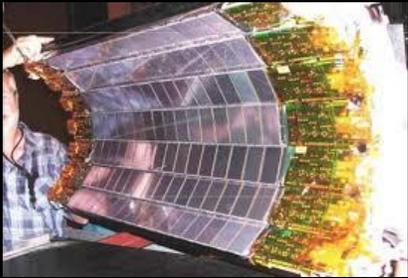


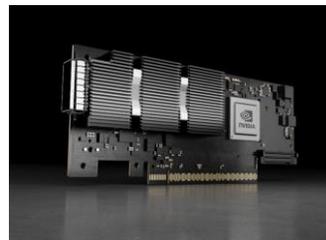
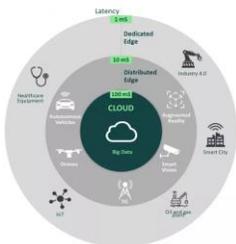
## Réseaux des détecteurs semi-conducteurs 1-2 Juillet 2025 – LP2IB

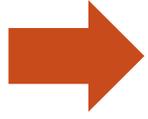


## L'intelligence Artificielle pour les détecteurs semi-conducteur – Principes et solutions–



SERVICE  
ELECTRONIQUE  
& ACQUISITION





Semi-conducteurs en physique

APD,

SiPM,

Diode PiN, DSSD, SiLi ....

Detecteur Diamant

...

Semi-conducteurs en physique → Caracteristiques

- Bruit,
- Résolution en énergie
- Résolution temporelle
- Multiplicité d'événements
- Traitement massif de données
- Influence de la chaine de lecture électronique



## Apport de l'IA pour surmonter ces défis



**On cherche un modèle pour prendre une décision:**

- ❑ Classification (qu'est ce que c'est)
- ❑ Régression (c'est quoi la réponse à une question )

Mathématiquement, c'est:

$$X_i \longrightarrow f(\bar{X}_i; \theta_j) \longrightarrow Y_i$$

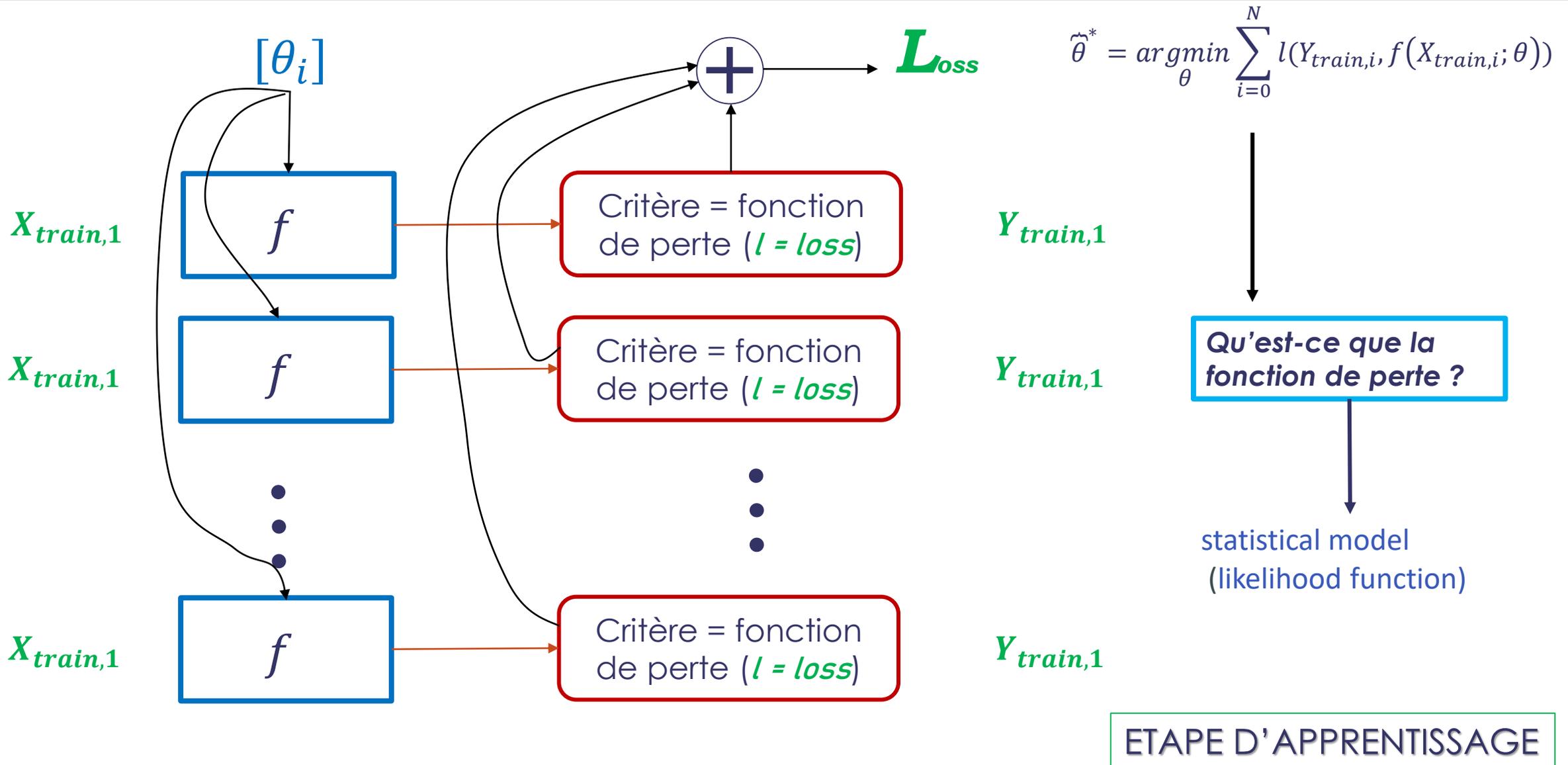
- On connaît  $\theta_j$ , on veut  $X_i \rightarrow$  on trouve à priori les  $Y_i$   Pb mal posé, déconvolution
- On connaît  $\theta_j$ , on veut  $Y_i \rightarrow$  on trouve à priori les  $X_i$   TNS/TNI Filtrage, convolution, IA
- On connaît  $X_i$ , on veut  $\theta_j \rightarrow$  on trouve à priori les  $Y_i$   (Apprentissage profond )



On utilise les méthodes d'optimisation

Dérivée

Minimise un critère



→ Selon les Y, on a des fonctions de perte dédiées:

- Relation « regression linéaires » entre les X et les Y:

$$MSE(\theta) = \frac{1}{N} \sum (Y_i - f(X_i; \theta))^2$$

$$RMSE(\theta) = \sqrt{\frac{1}{N} \sum (Y_i - f(X_i; \theta))^2}$$

Régularisation de Thikonov (Regularisation **Ridge**):

$$J(\theta) = MSE(\theta) + \alpha \frac{1}{2} \sum_{i=0}^N \theta_i^2$$

Régularisation « Least Absolute Shrinking and Selection Operator » **lasso**:

$$J(\theta) = MSE(\theta) + \alpha \frac{1}{2} \sum_{i=0}^N |\theta_i|$$

Mesure de distance ou normes  $\ell_k$ :  
 $\ell_1, \ell_2, \ell_0, \dots, \ell_{\infty}$

$$\alpha \frac{1}{2} \sum_{i=0}^N \theta_i^2 \longrightarrow \ell_2$$

Elastic Net

$$J(\theta) = MSE(\theta) + r\alpha \frac{1}{2} \sum_{i=0}^N |\theta_i| + \alpha \frac{1-r}{2} \sum_{i=0}^N \theta_i^2$$

→ Selon les  $Y$ , on a des fonctions de perte dédiées:

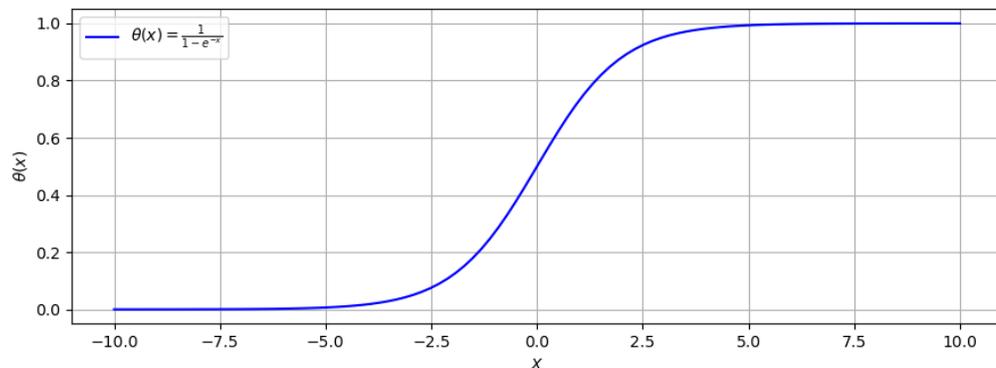
- Classification « regression logistique » (logit): On établit la probabilité d'appartenance à une classe

$$\text{Fonction d'activation} = \frac{1}{1 - \exp(-x)}$$

Fonction logistique

fonctions de perte

$$RMSE(\theta) = -\frac{1}{N} \sum [Y_i \cdot \log(\hat{p}) + (1 - Y_i) \log(1 - \hat{p})]$$

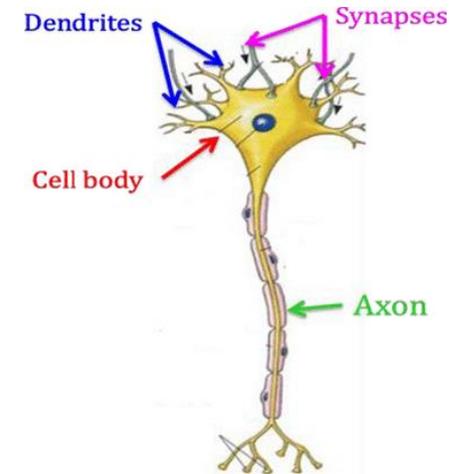
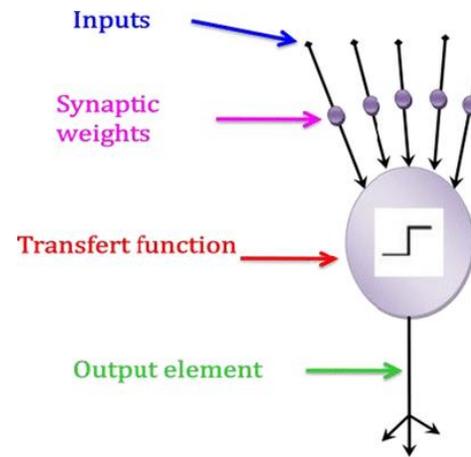
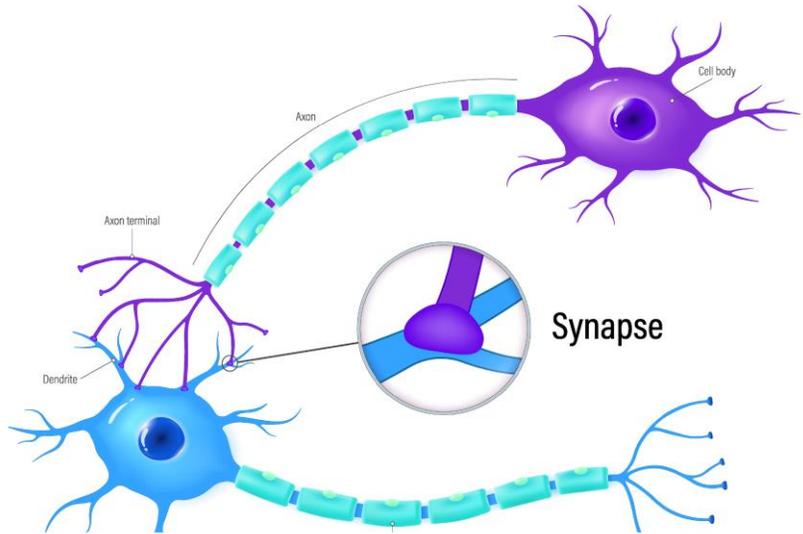
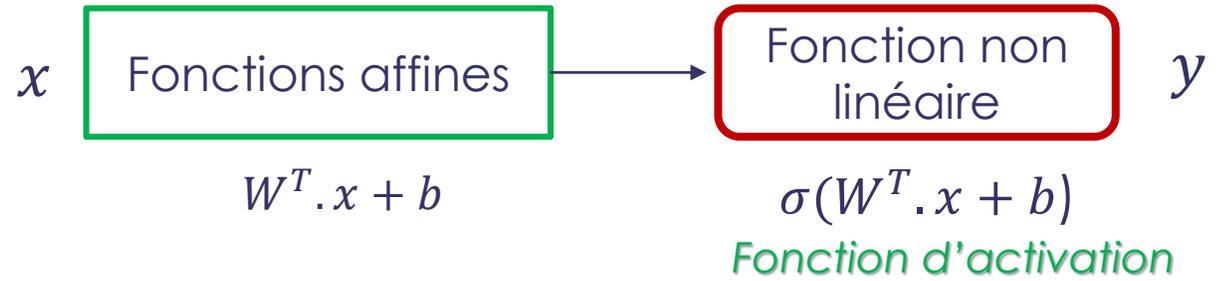


- Classification « regression softmax »: apprentissage avec plusieurs catégories en même temps

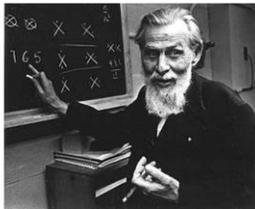
## Du neurone biologique au neurone formel



$$f[\theta_i]$$



McCullock & Pitts – 1943  
A logical calculus of the ideas immanent in nervous activity



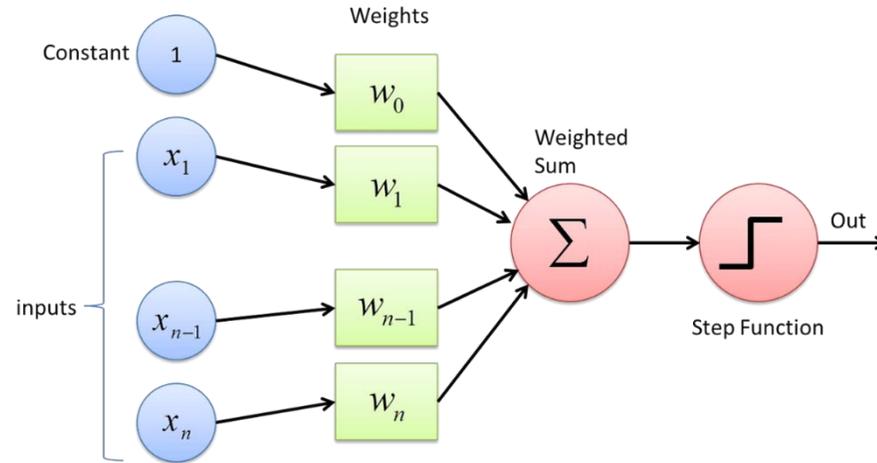
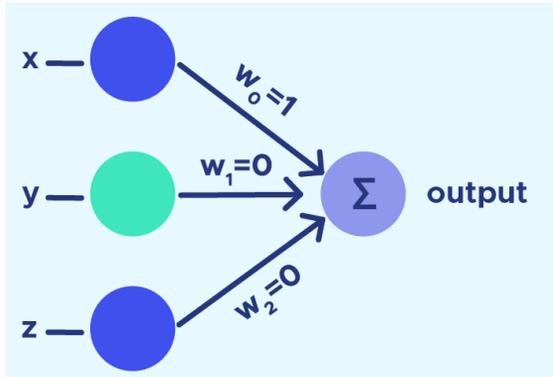
Mc Cullock



Pitts

$$Z_i = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \vdots \end{bmatrix} [X_1 X_2 \dots] + [b_1 b_2 b_3 \dots] \longrightarrow Y_i = \sigma(Z_i)$$

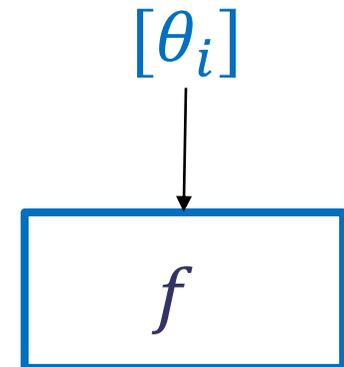
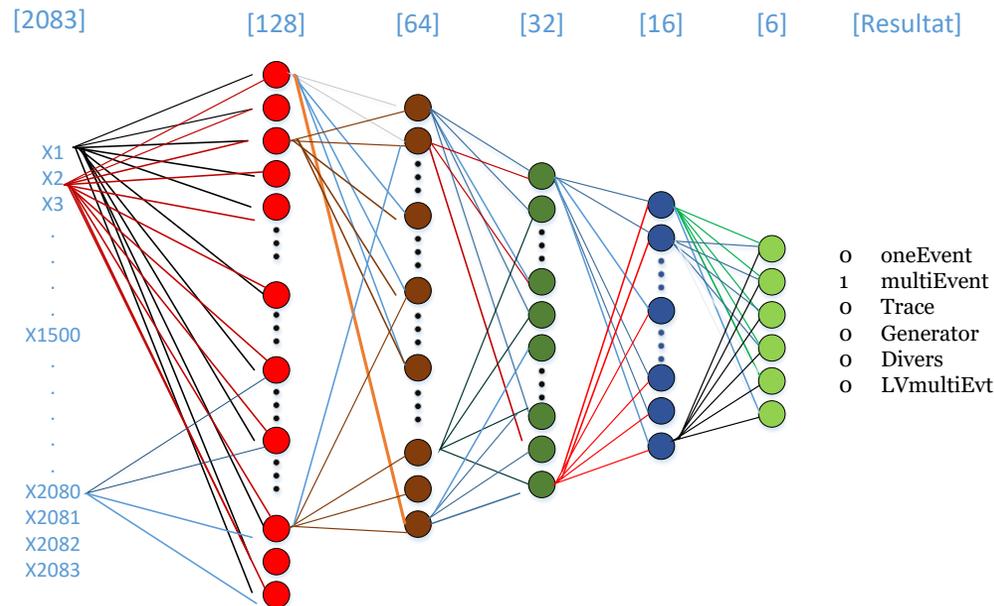
## Perceptron



Mais comment cela fonctionne ?

Pourquoi arrive-t-on à modéliser n'importe quelles fonctions avec ce simple modèle affine ?

Perceptron Multi-couche  
 PMC  
 Artificial Neural Network  
 Deep Neural Network

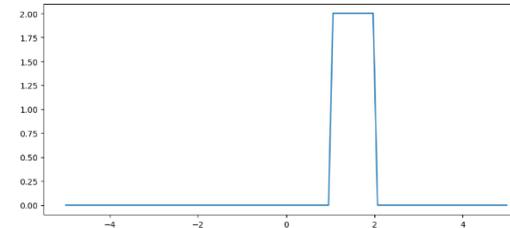
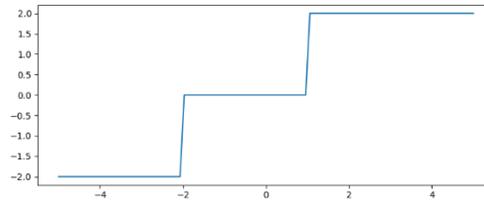
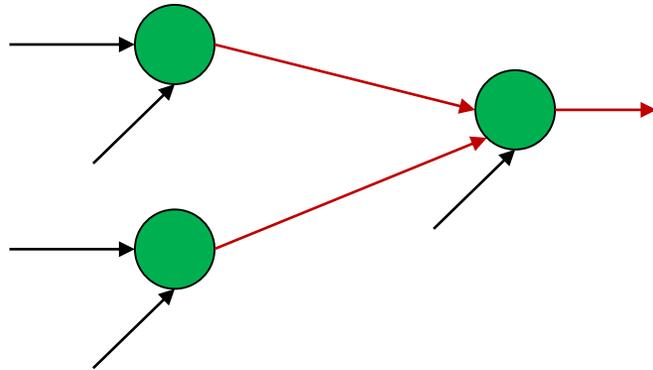
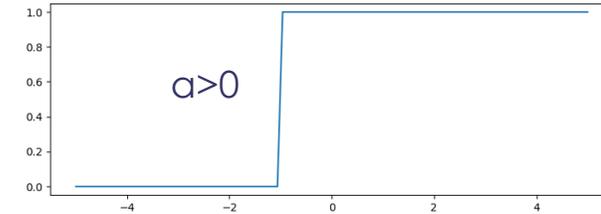
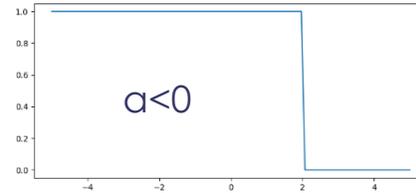
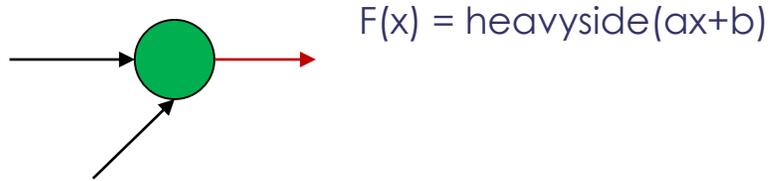


## Théorème de l'approximateur universel

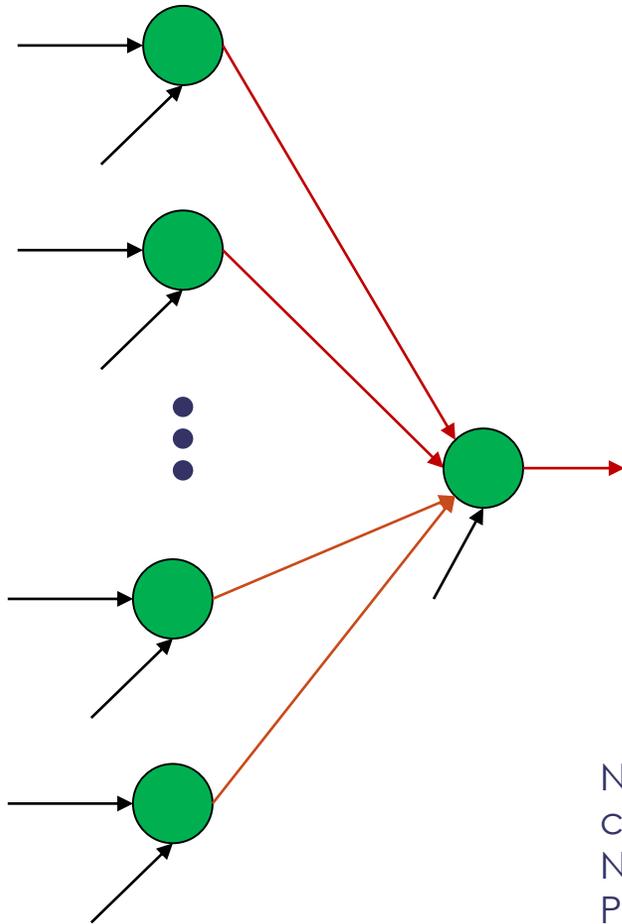
*toute fonction continue*

$$f: [a, b] \times [a, b] \rightarrow \mathbb{R}$$

*peut être approchée d'aussi près que l'on veut par une fonction  $F: [a, b] \times [a, b] \rightarrow \mathbb{R}$  réalisée apr un réseau de neurones.*



# Un approximateur universel

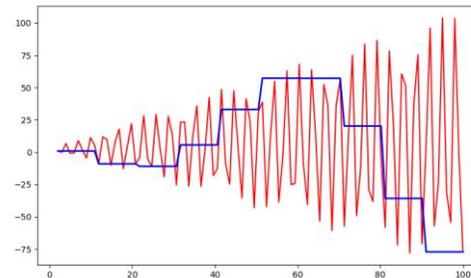
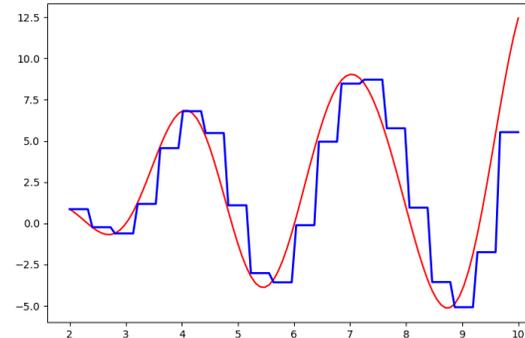
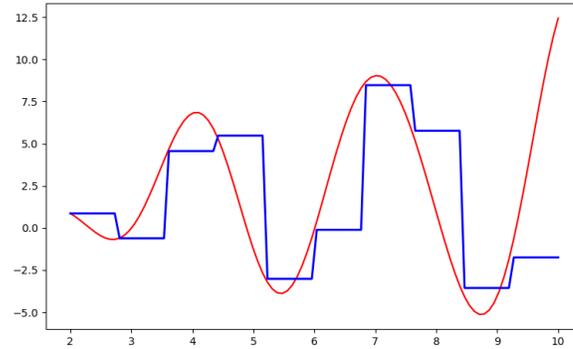


N neurones  
couche 0  
N = 10

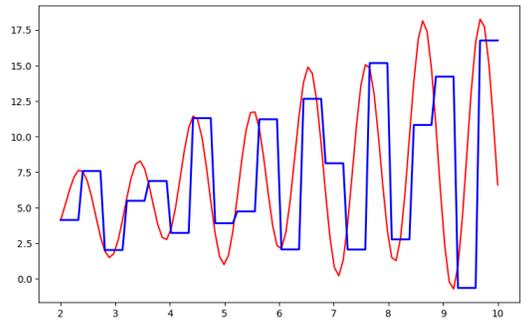
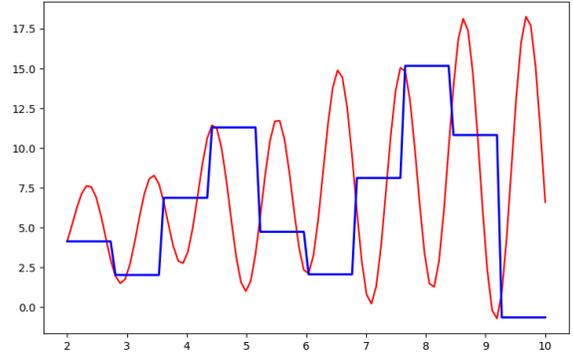
N neurones  
couche 0  
N = 20

N neurones  
couche 0  
N = 10  
Plus de X

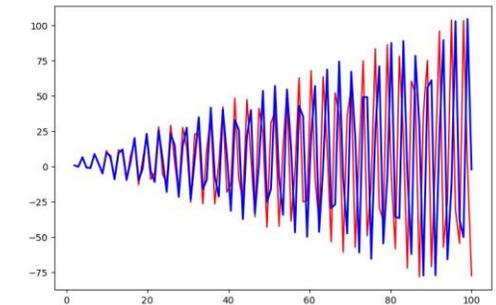
$$\text{np.cos}(3*x) + x*\text{np.sin}(2*x) + x**0.5$$



$$\text{np.cos}(3*x) + x*\text{np.sin}(6*x) + 3*x**0.5$$



N neurones  
couche 0  
N = 100  
Plus de X





On cherche un modèle pour prendre une décision:

- ❑ Classification (qu'est ce que c'est)
- ❑ Régression (c'est quoi la réponse à une question)

Mathématiquement, c'est:

$$X_i \longrightarrow f(\bar{X}_i; \theta_j) \longrightarrow Y_i$$

On connaît  $X_i$ , on veut  $\theta_j \rightarrow$  on trouve à priori les  $Y_i$   (Apprentissage profond)

Les applications IA aux détecteurs semi-conducteurs:

- ✓ **Vaincre les limites des méthodes classiques (Pb mal posé au sens d'Hadamart)**
- ✓ **Déconvolution**
- ✓ **Identification des signaux**
- ✓ **Débruitage**
- ✓ **Amélioration des mesures (Calibration, détection d'anomalies)**
- ✓ **Simulation, génération de signaux**
- ✓ **Gestion de grand volume de données**

# Vaincre les limites des méthodes classiques

→ Le concept mathématique de problème bien posé provient d'une définition de Hadamard qui pensait que les modèles mathématiques de phénomènes physiques devraient avoir les propriétés suivantes :

Une solution existe ;

La solution est unique ;

La solution dépend de façon continue des données dans le cadre d'une topologie raisonnable.

→ Problème inverse: déterminer  $M(y, x, \theta, \epsilon) = 0$  avec

$x(t)$  : une grandeur non directement observable à déterminer,

à partir d'un ensemble

fini de mesures d'une grandeur observée  $y(u)$  dépendant de paramètres  $\theta$

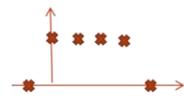
Et ayant un bruit  $\epsilon$

→ Déconvolution en temps réel

Imperfection des filtrages numériques:

- Imperfection des ADC et DAC
- Fonction de transfert avec coefficients finis
- Détermination des coefficients des modèles

Troncature :



$$x_N[k] = x_{\infty}[k] \cdot w_T[k] \text{ avec } w_T \text{ la fonction de Dirichlet :}$$

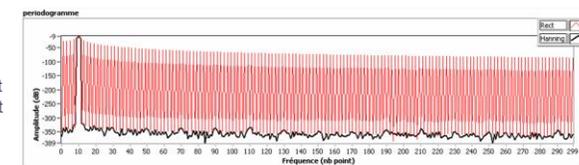
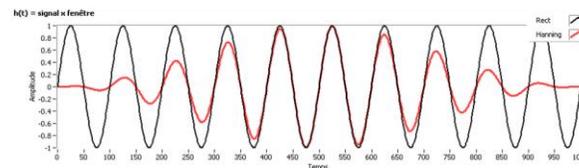
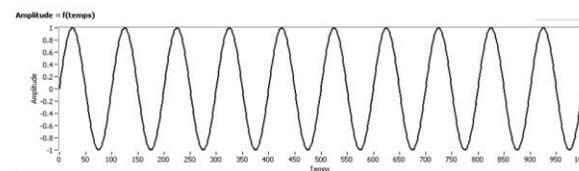
En fréquence, on obtient un produit de convolution :

$$X_N(v) = (X_{\infty} * W_N)(v) \text{ avec}$$

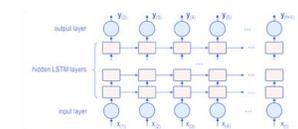
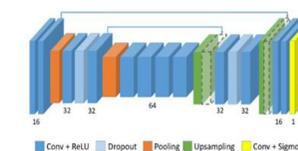
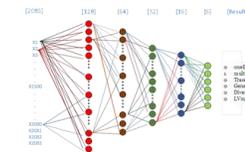
$$W_N(v) = N \cdot \frac{\sin(\pi N v)}{N \cdot \sin(\pi v)} e^{-i2\pi v \frac{N-1}{2}}$$

→ Principe de Gabor : On ne peut être localisé en temps si on est localisé en fréquence :

$$\Delta t^2 \cdot \Delta f^2 \geq \frac{1}{16\pi^2}$$

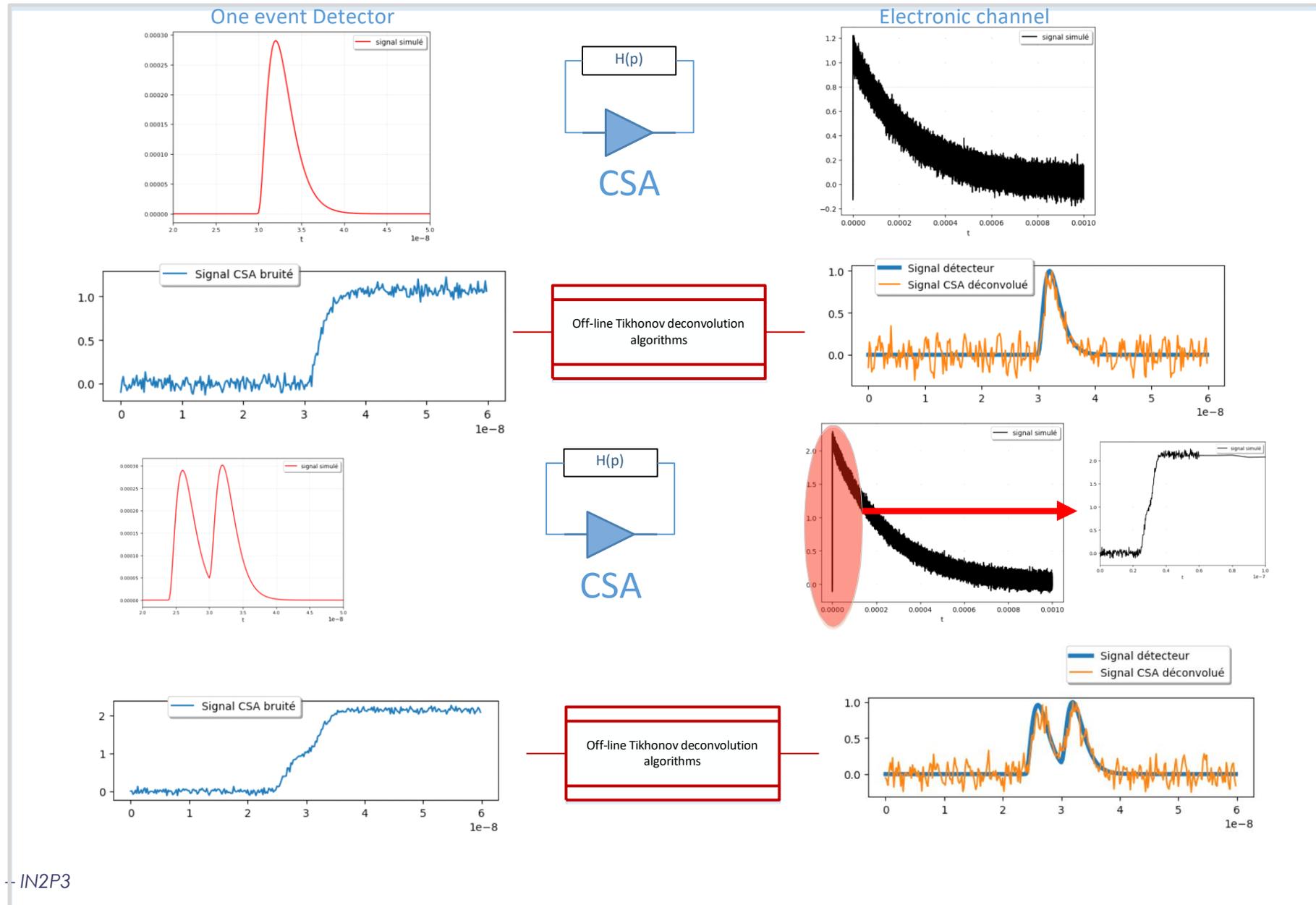


Créer par IA un modèle:  
 $X_i \longrightarrow f(\bar{X}_i; \theta_j) \longrightarrow Y_i$

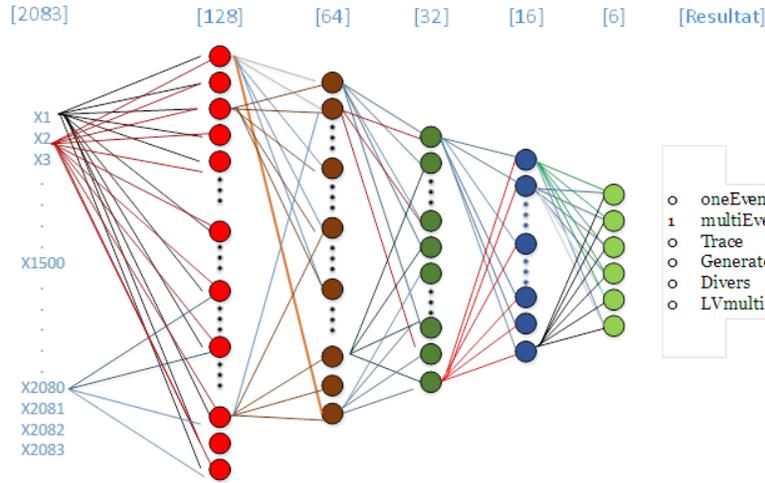
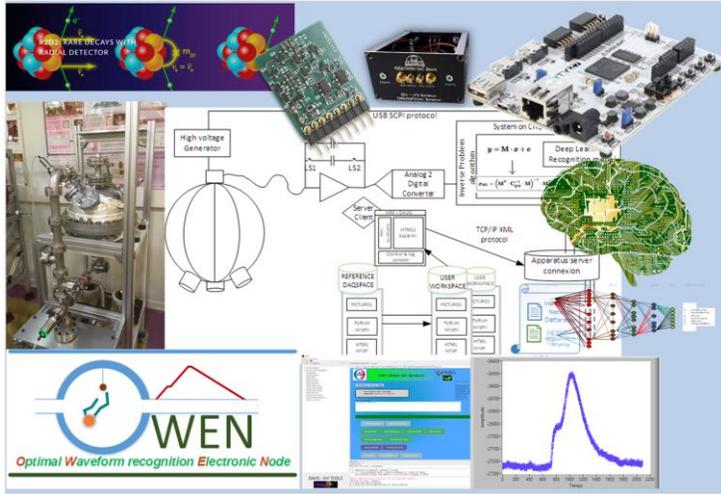


Problème mal posé  
au sens  
d'Hadamard:

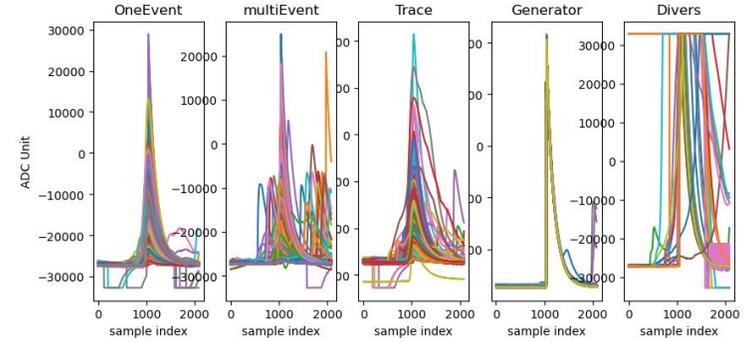
→ Algorithme Off-  
Line



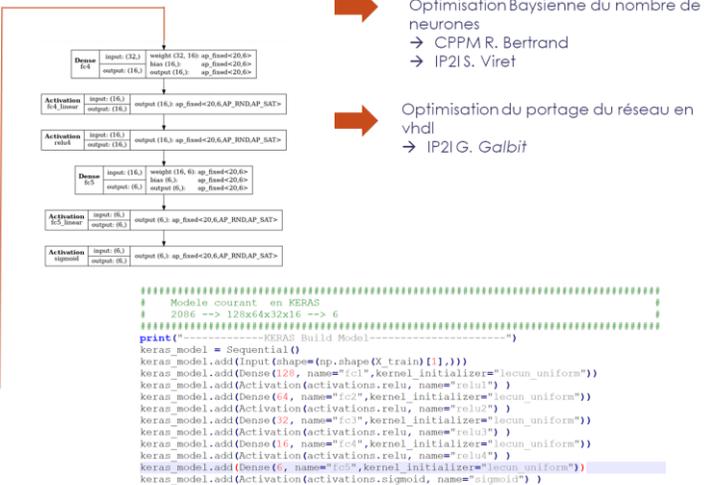
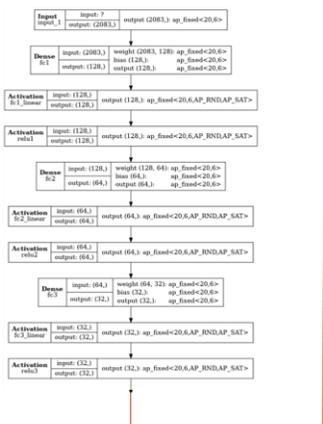
# Identification des signaux: R2D2: Application Neutrino



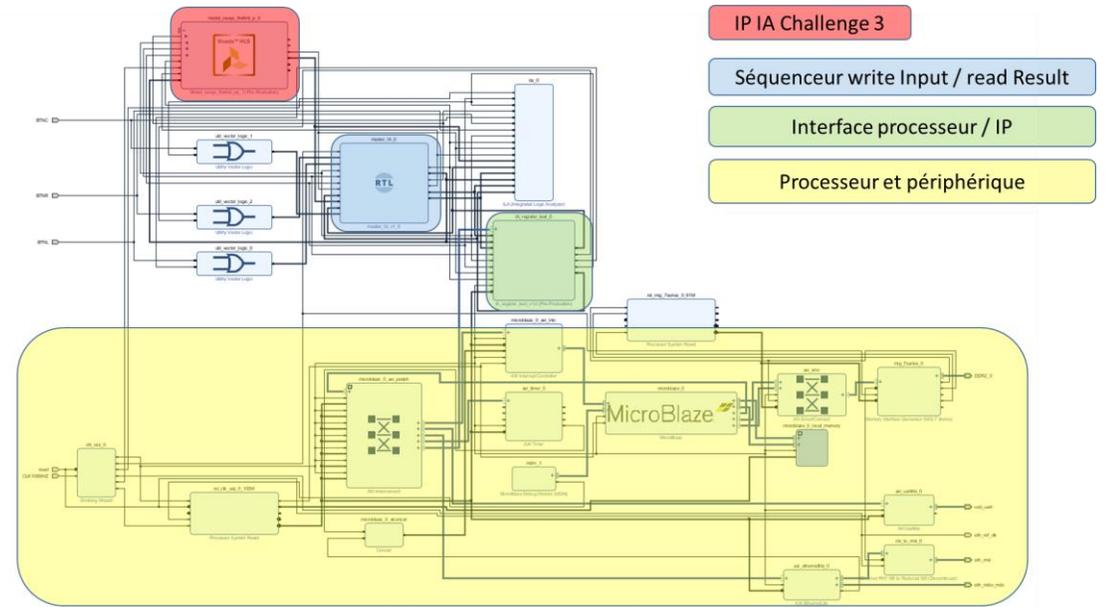
- oneEvent
- multiEvent
- Trace
- Generator
- Divers
- LVmultiEvt



## Le réseau R2D2 OWEN IA



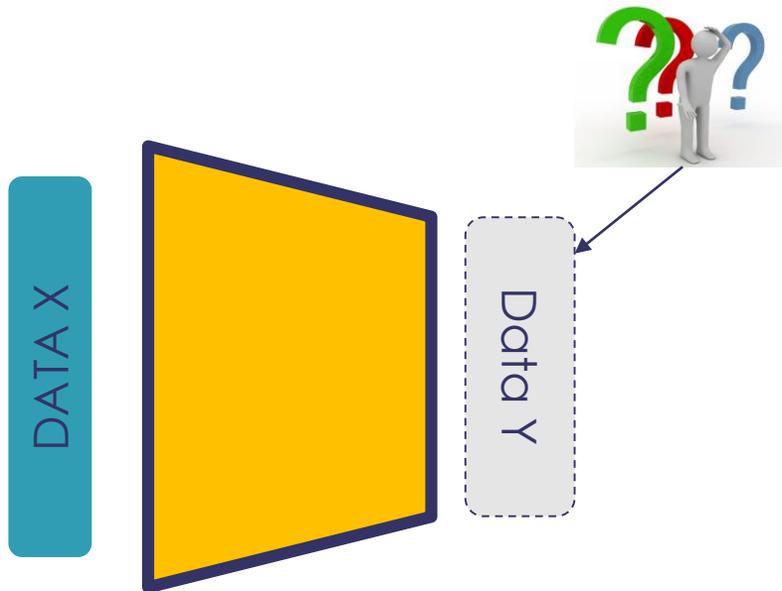
73%



➔ 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

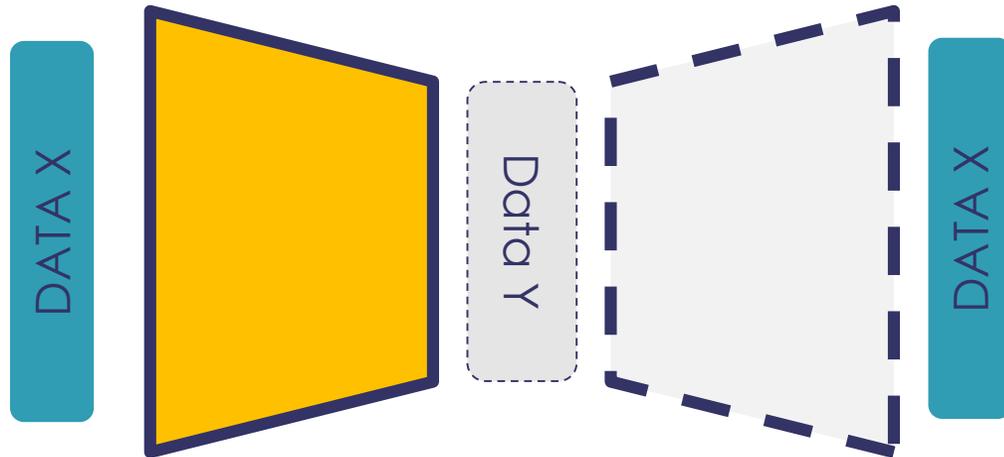


➔ Et si l'IA avait de l'imagination

On ne connaît que les X



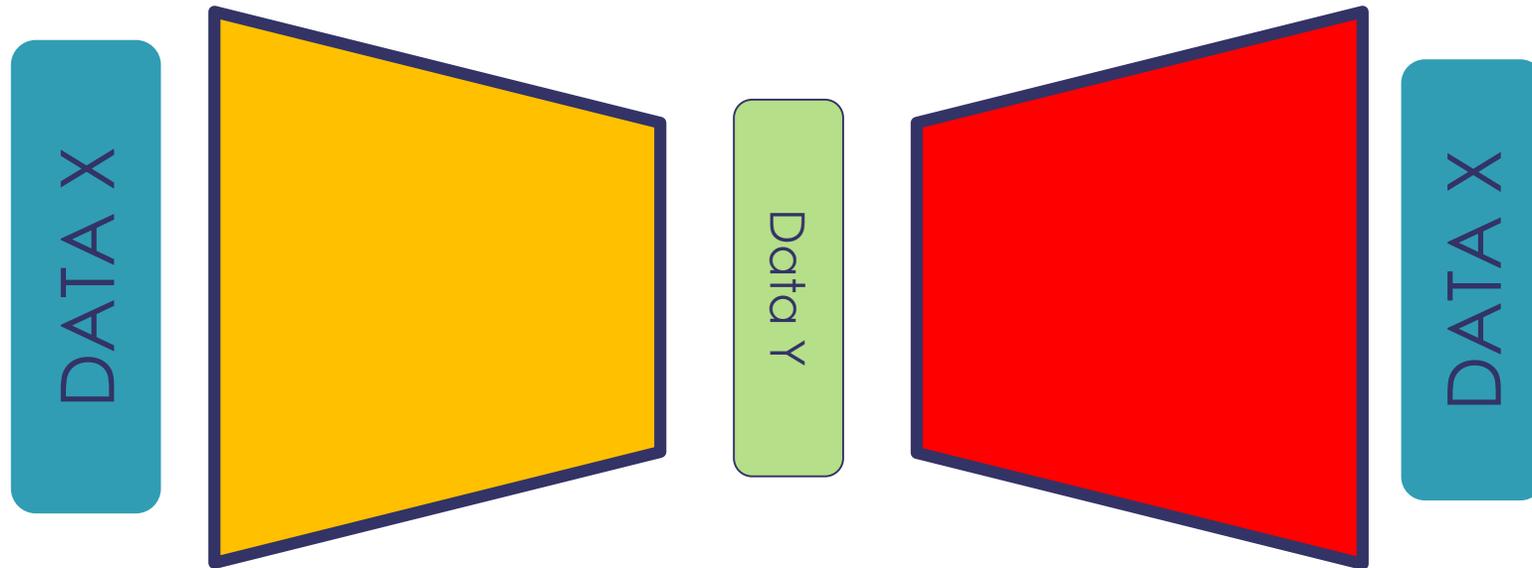
$$Y = X$$



➔ Et si l'IA avait de l'imagination

On ne connaît que les X

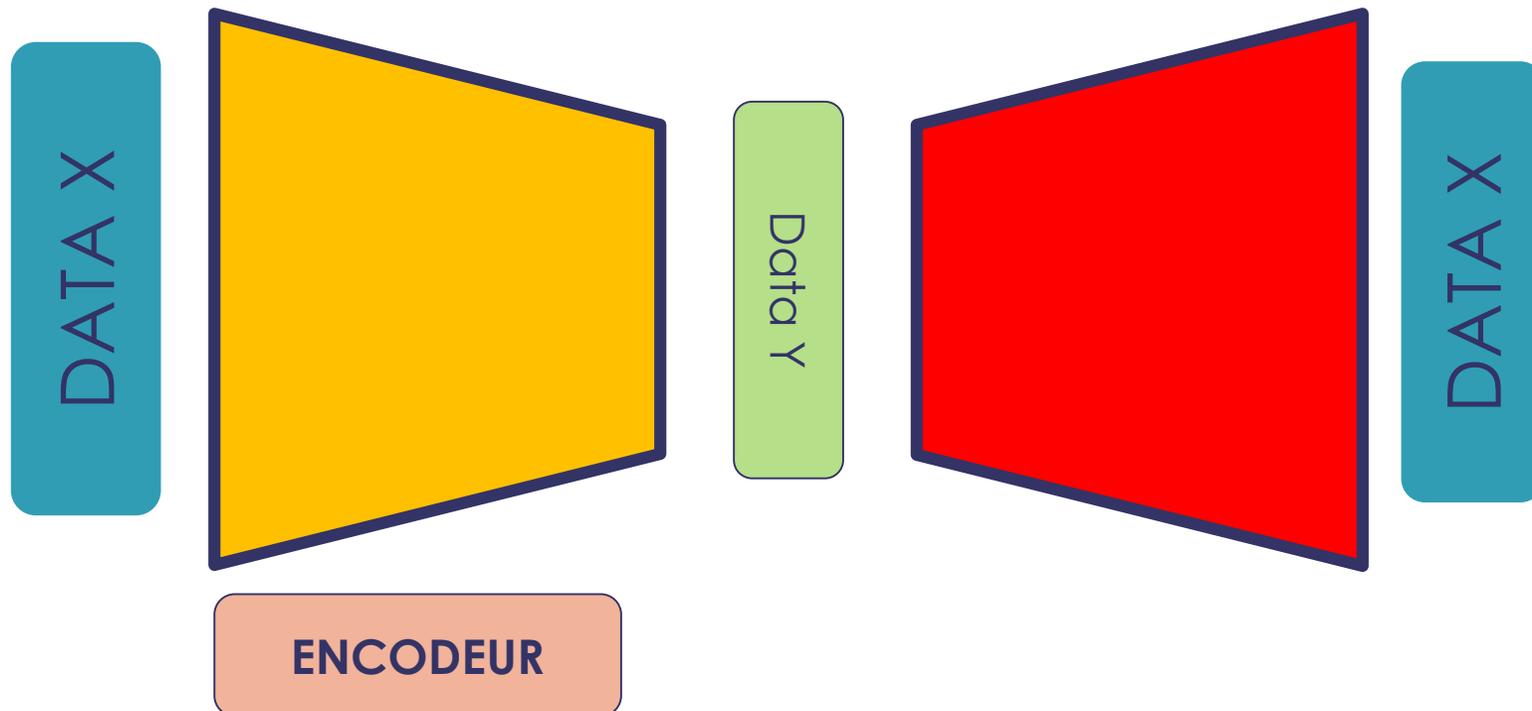
→  $Y = X$



➔ Et si l'IA avait de l'imagination

On ne connaît que les X

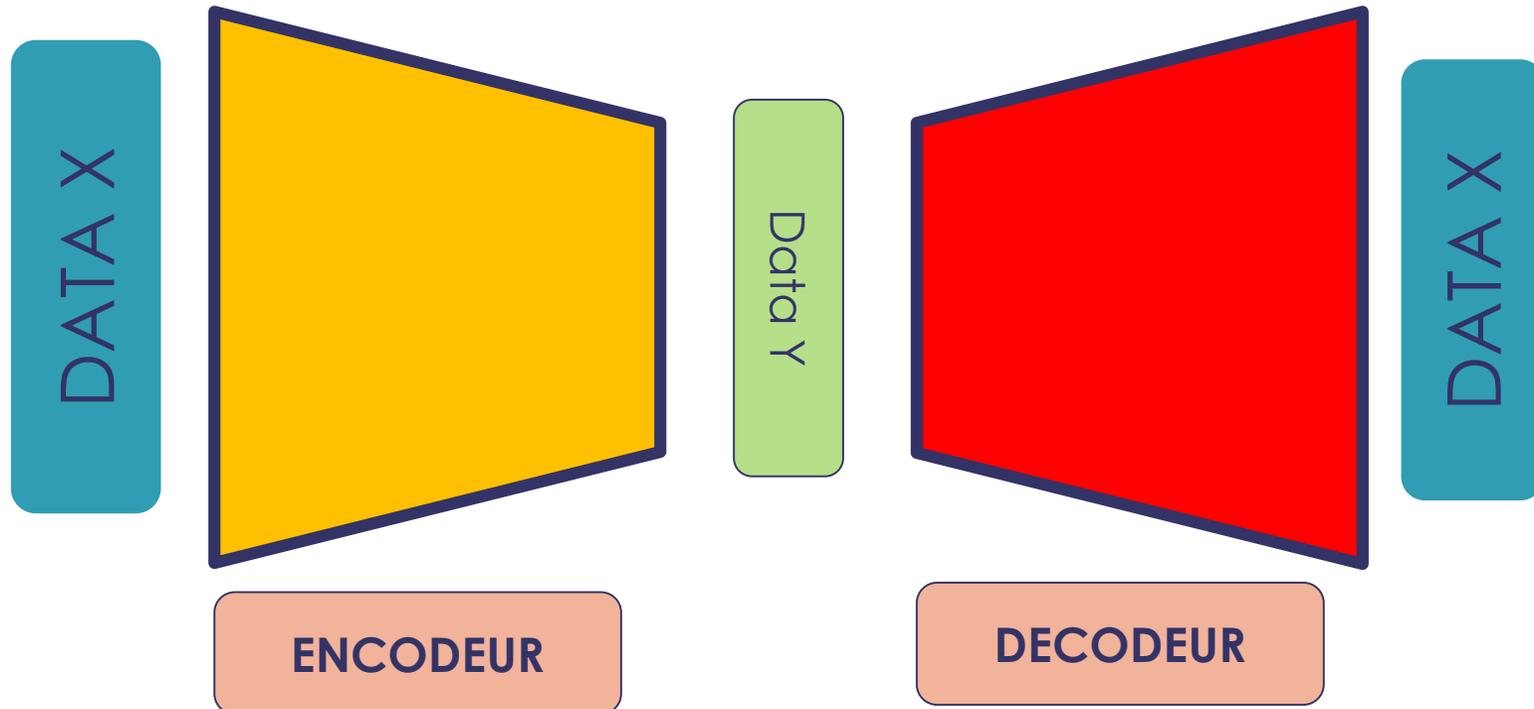
→  $Y = X$



➔ Et si l'IA avait de l'imagination

On ne connaît que les X

→  $Y = X$



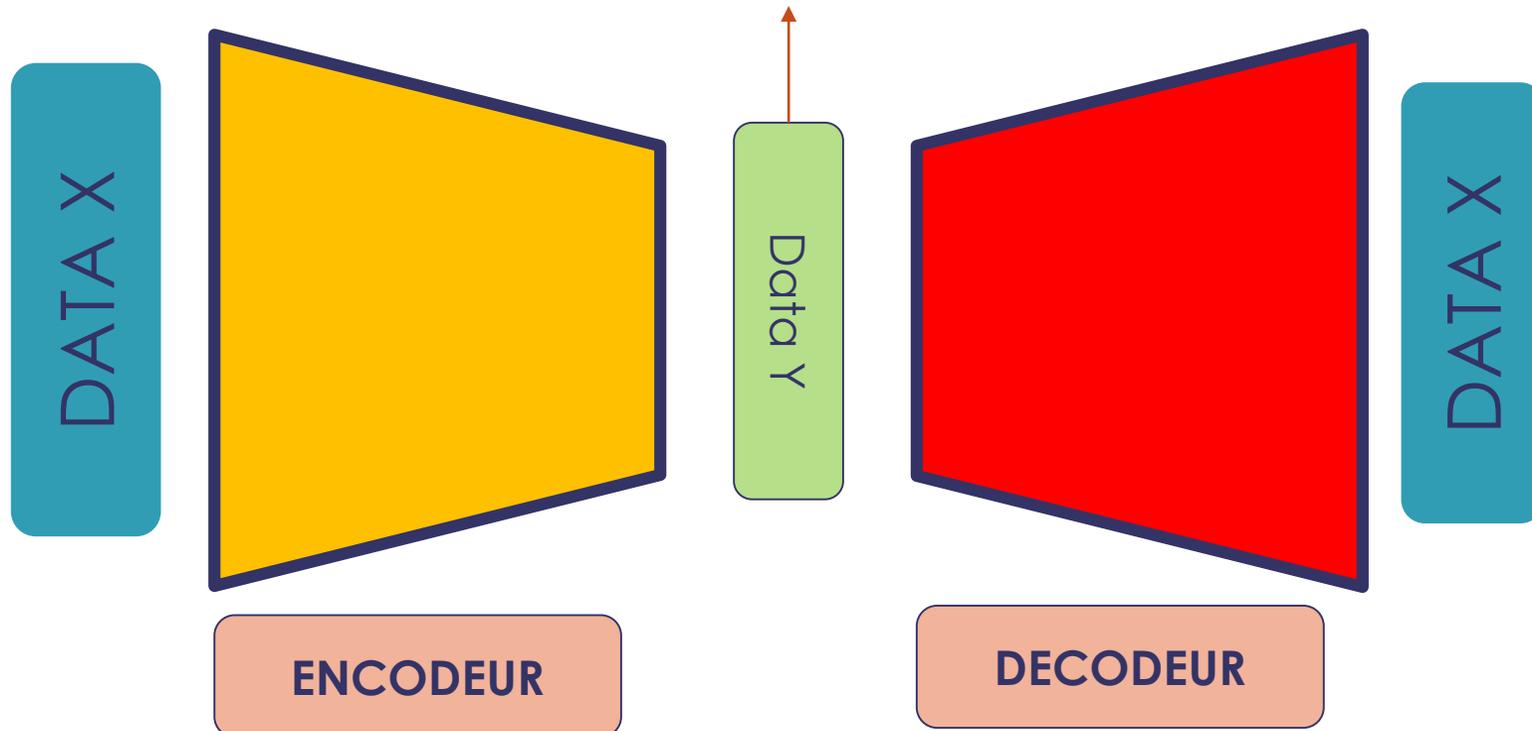
➔ Et si l'IA avait de l'imagination

On ne connaît que les X



$$Y = X$$

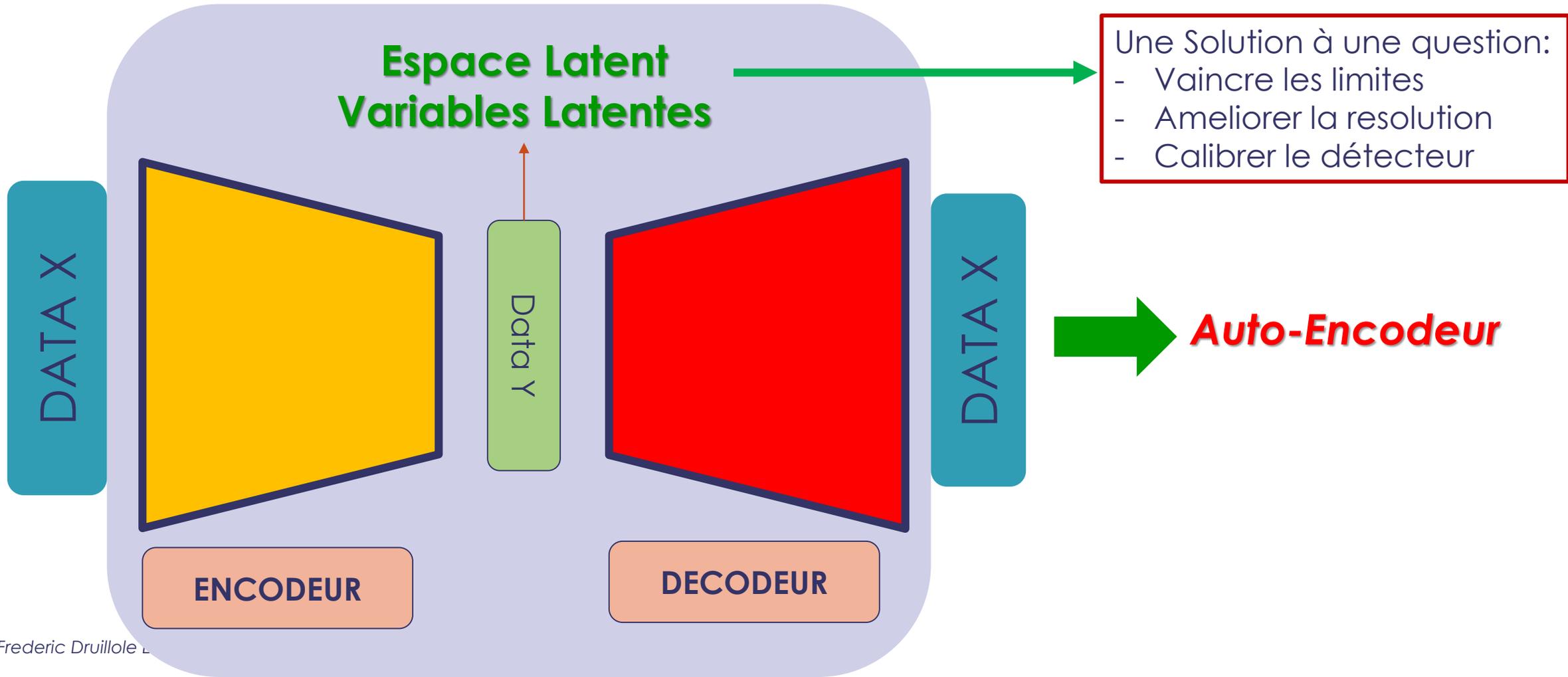
**Espace Latent**  
**Variables Latentes**



➔ Et si l'IA avait de l'imagination

On ne connaît que les X

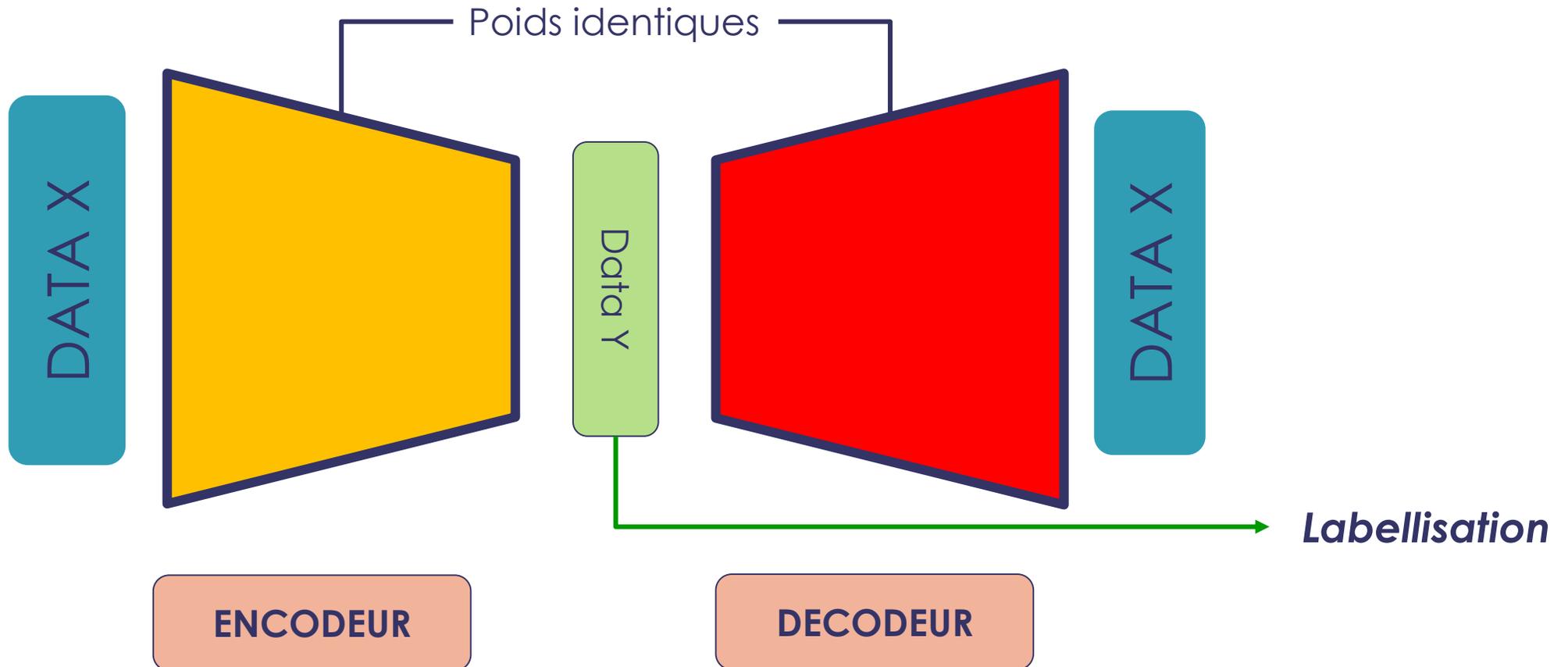
$$Y = X$$



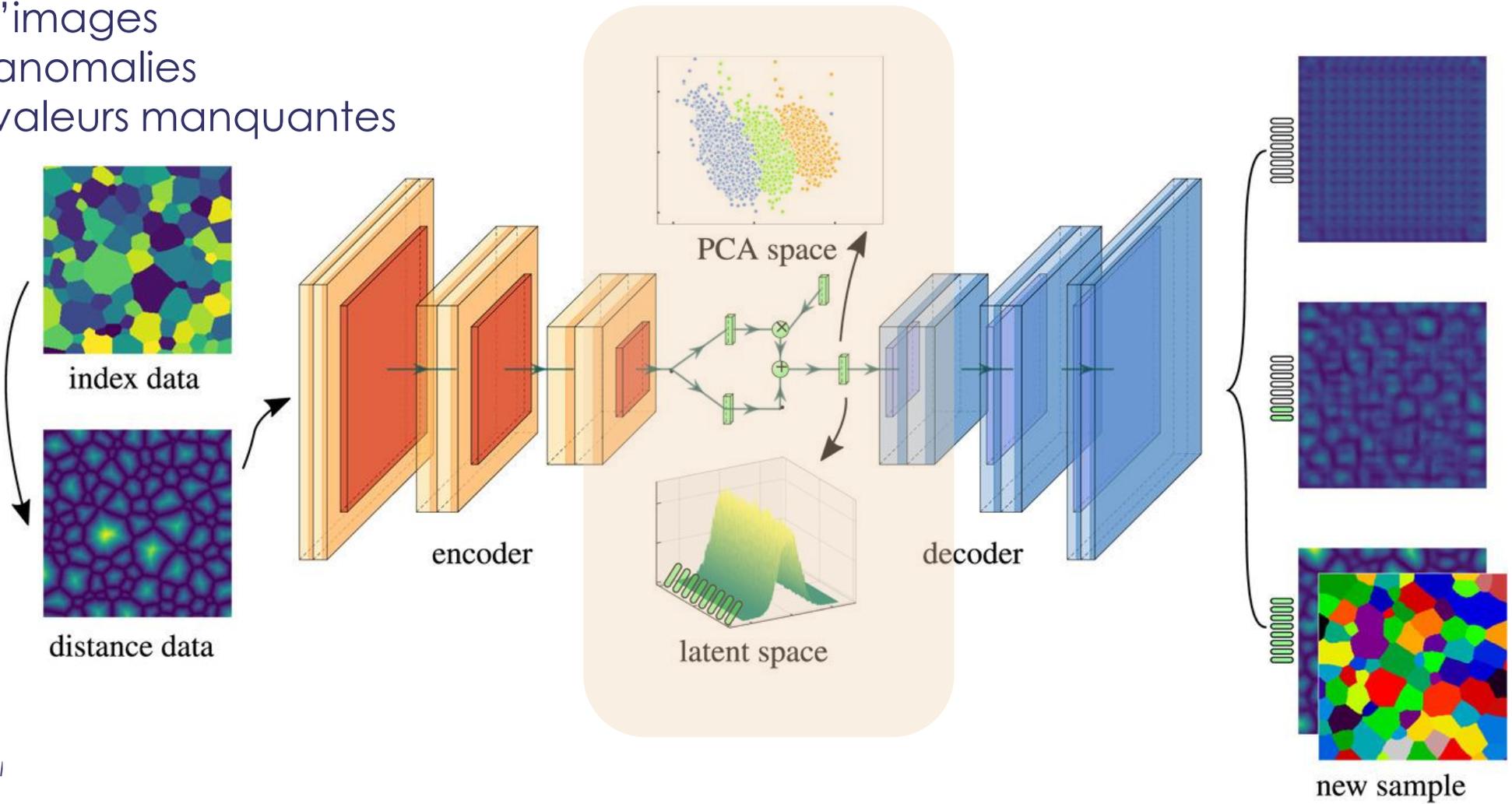
➔ Et si l'IA avait de l'imagination

On ne connaît que les X

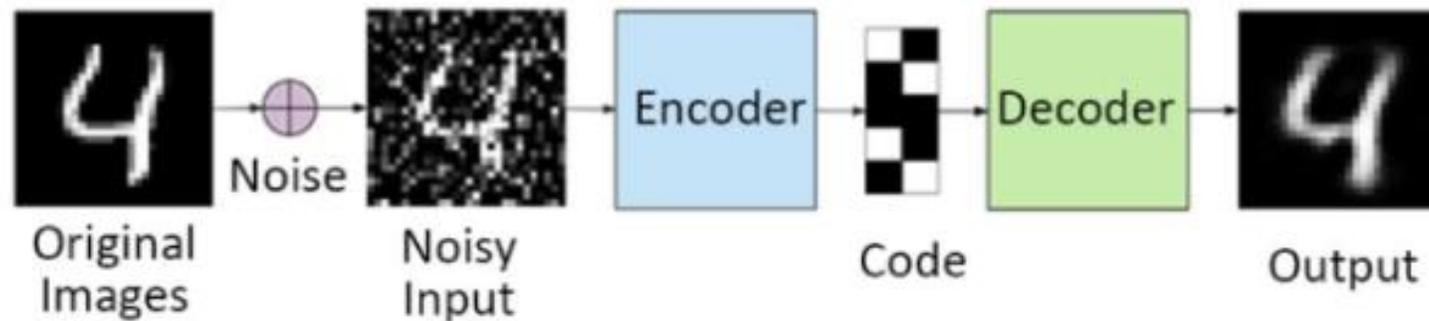
➔  $Y = X$



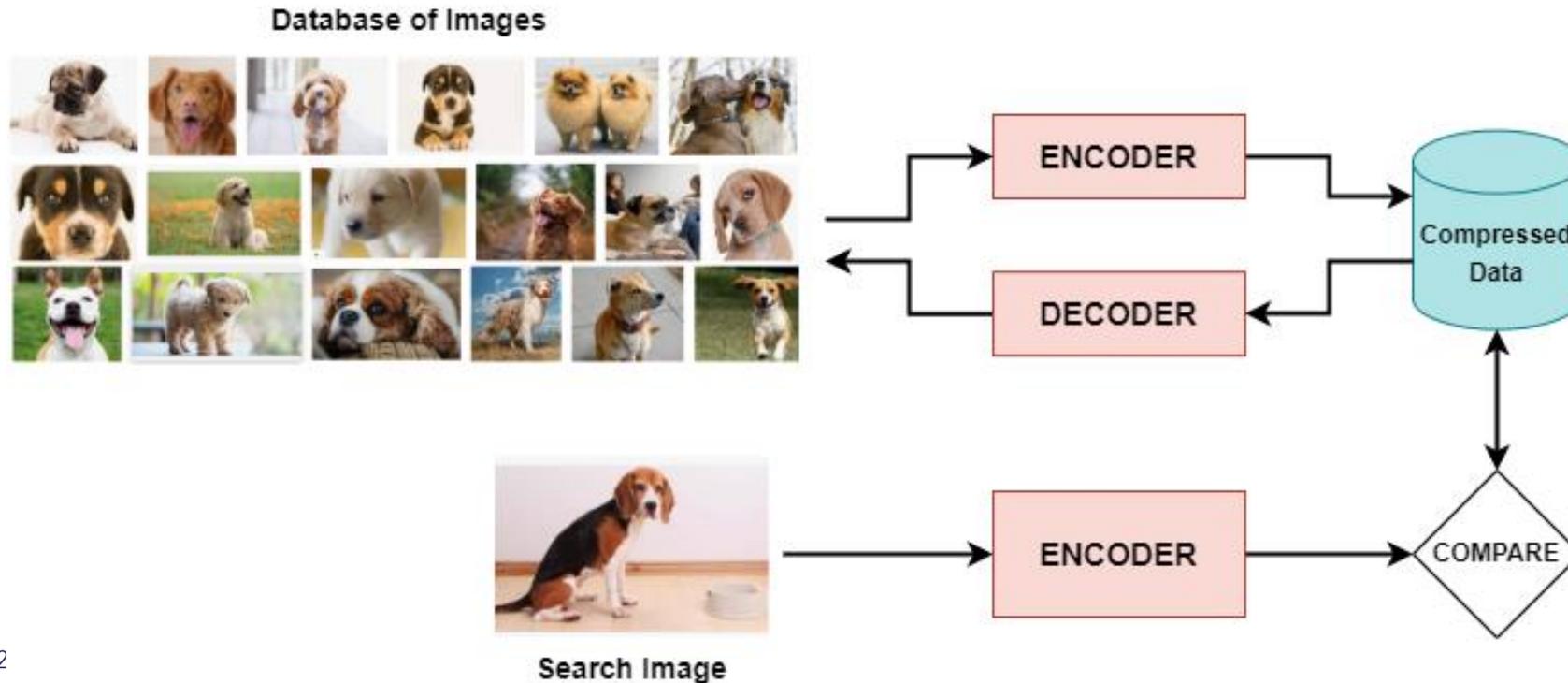
1. Réduction de dimensionnalité
2. Extraction de caractéristiques
3. Débruitage d'images
4. Recherche d'images
5. Détection d'anomalies
6. Création de valeurs manquantes



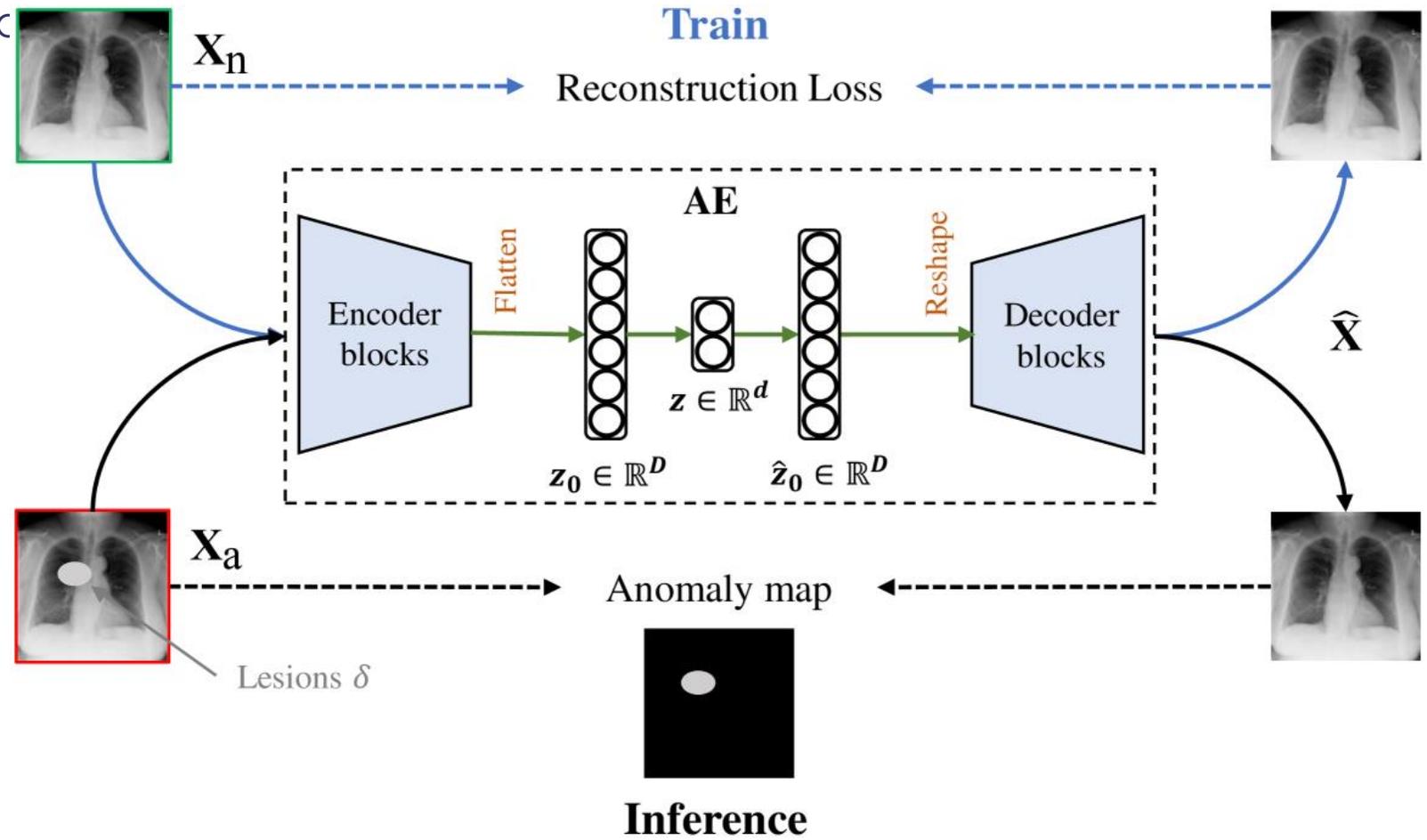
1. Reduction de dimensionalité
2. Extraction de caractéristiques
- 3. Débruitage d'images**
4. Recherche d'images
5. Détection d'anomalies
6. Création de valeurs manquantes



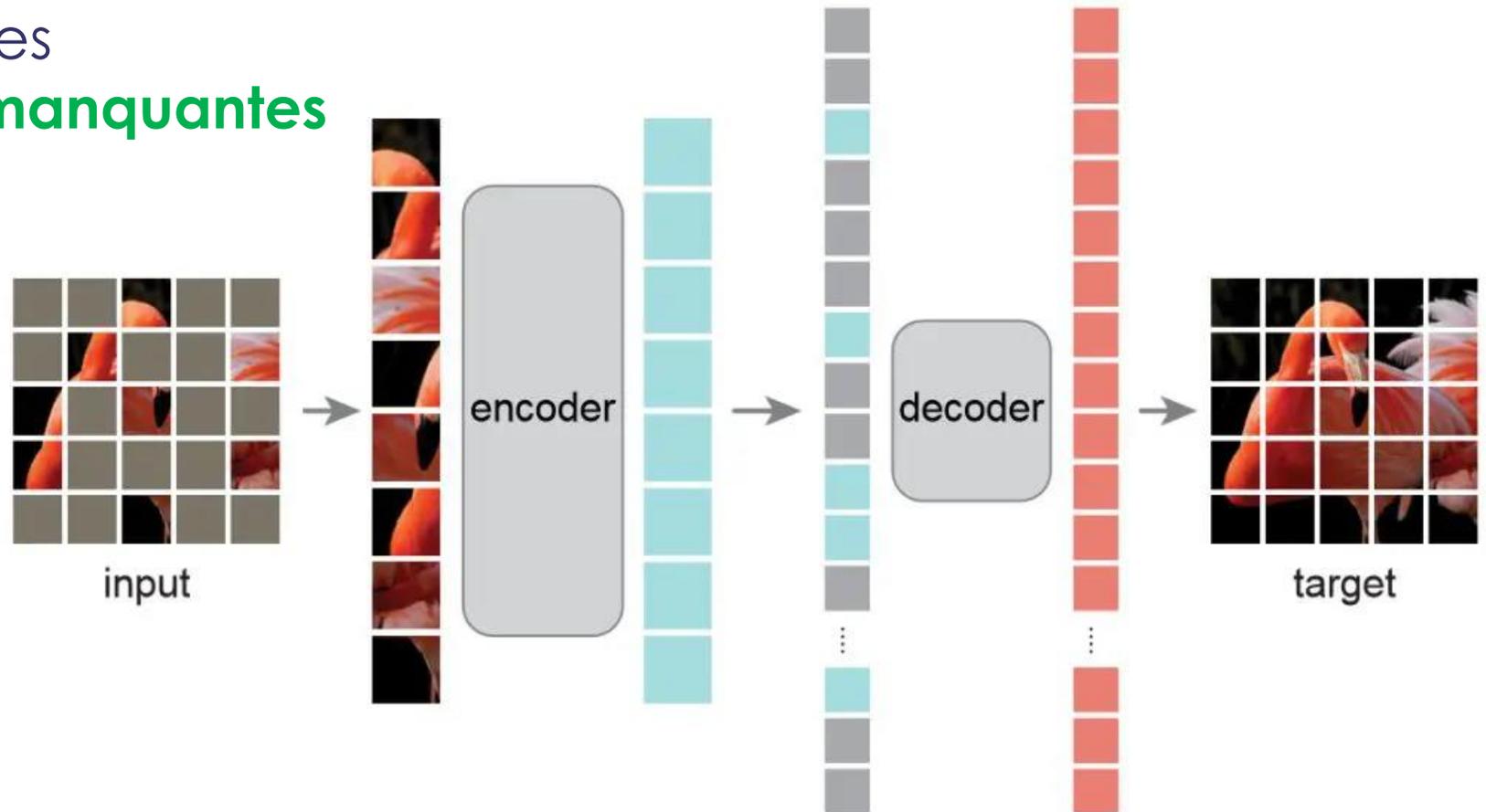
1. Reduction de dimensionalité
2. Extraction de caractéristiques
3. Débruitage d'images
- 4. Recherche d'images**
5. Détection d'anomalies
6. Création de valeurs manquantes

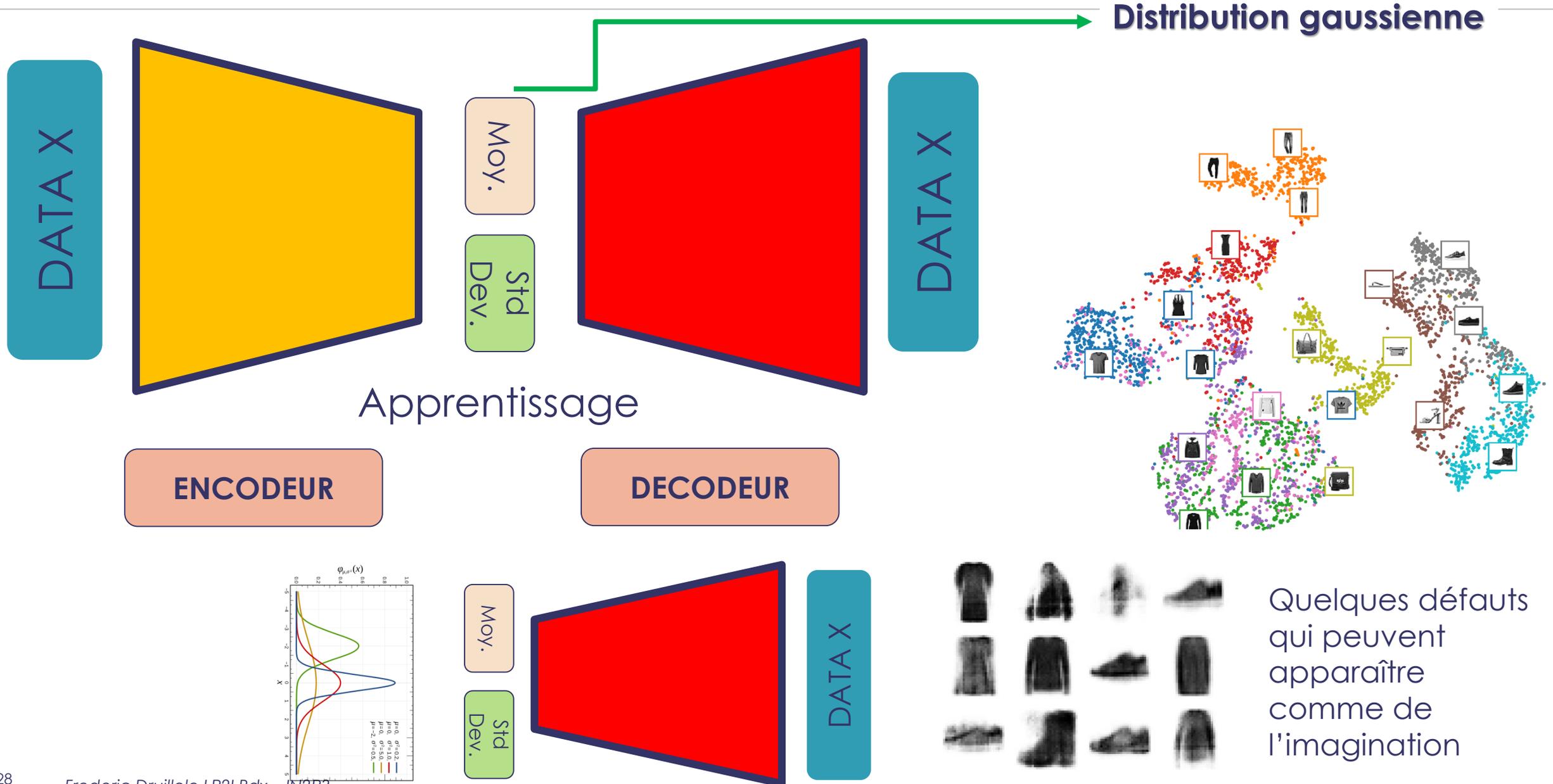


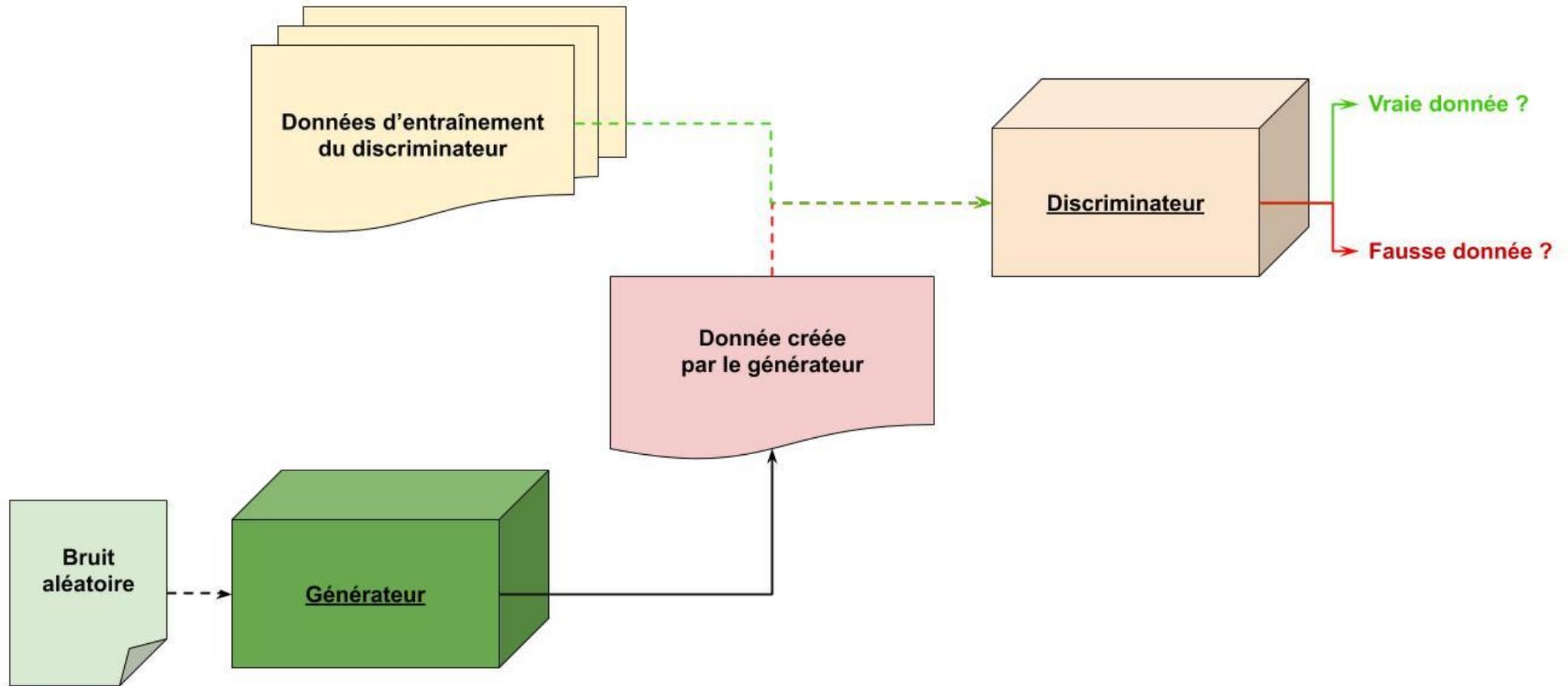
1. Reduction de dimensionalité
2. Extraction de caractéristiques
3. Débruitage d'images
4. Recherche d'images
- 5. Détection d'anomalies**
6. Création de valeurs manquantes



1. Réduction de dimensionnalité
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 »

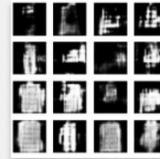
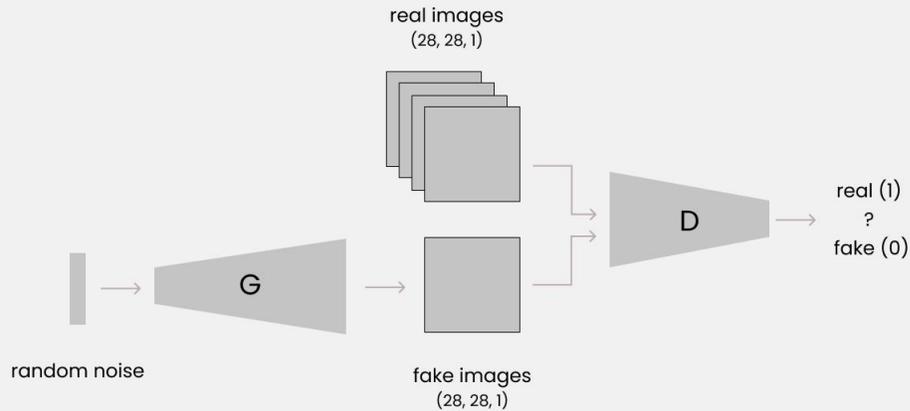


## DCGAN for Fashion MNIST

Get Started: Deep Convolutional  
Generative Adversarial Networks

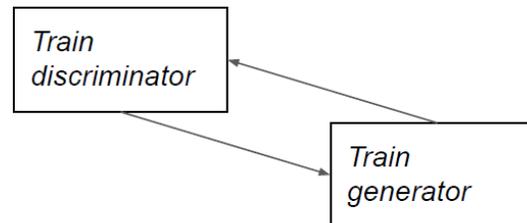
pyimagesearch

GANs 101



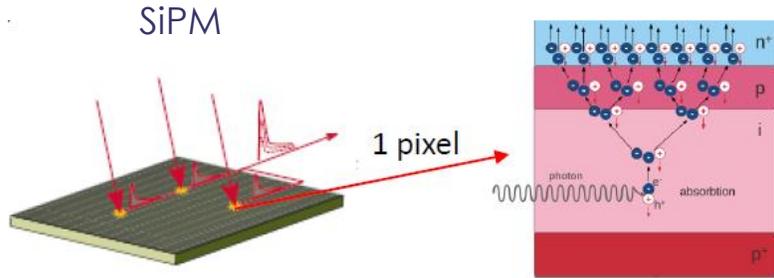
- ❑ Entraînement du générateur
- ❑ + Entraînement du discriminateur
- ❑ Entraînement du générateur sans discriminateur

### Training process



### GAN COST FUNCTION

$$\begin{aligned} \min_G \max_D L(D, G) &= \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \\ &= \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{x \sim p_g(x)} [\log(1 - D(x))] \end{aligned}$$



## Imperfections:

### Température :

- augmente le courant d'obscurité, ce qui peut entraîner un bruit de fond plus élevé.

### Tension de polarisation :

- Une tension trop élevée peut augmenter le bruit et endommager le détecteur, tandis qu'une tension trop basse peut réduire le gain et la sensibilité.

### Photons parasites :

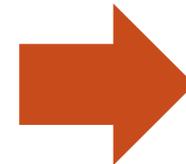
- Le crosstalk : peut déclencher une avalanche dans une cellule voisine.
- La lumière ambiante ou les photons parasites provenant d'autres sources peuvent également déclencher des signaux indésirables, augmentant ainsi le bruit de fond.

### Radiation :

L'exposition à des rayonnements ionisants peut endommager les SiPM, réduisant leur efficacité et augmentant le courant d'obscurité.

### Bruit électronique :

- Les interférences électromagnétiques (EMI) et le bruit thermique dans les circuits de lecture peuvent affecter la qualité du signal détecté



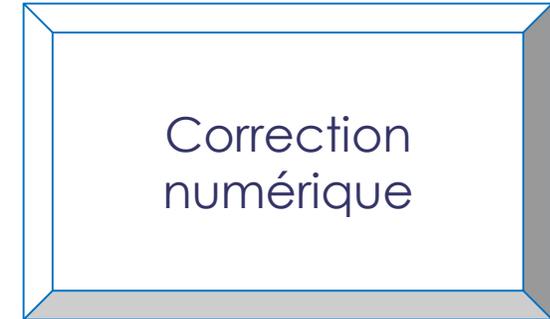
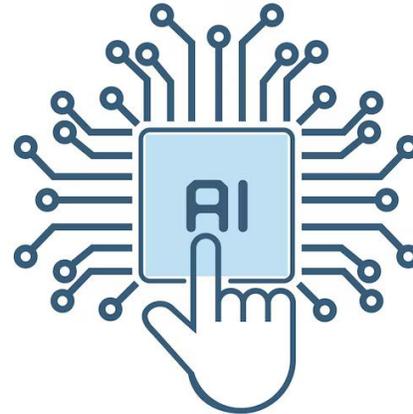
**Peut-on faire une correction automatique par IA ?**

# Modélisation du Comportement en Fonction de la Température

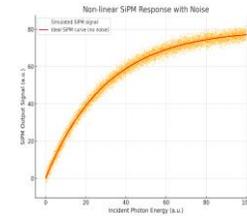
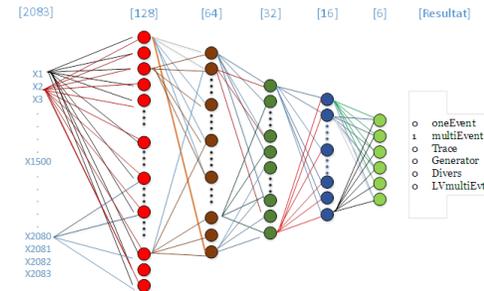
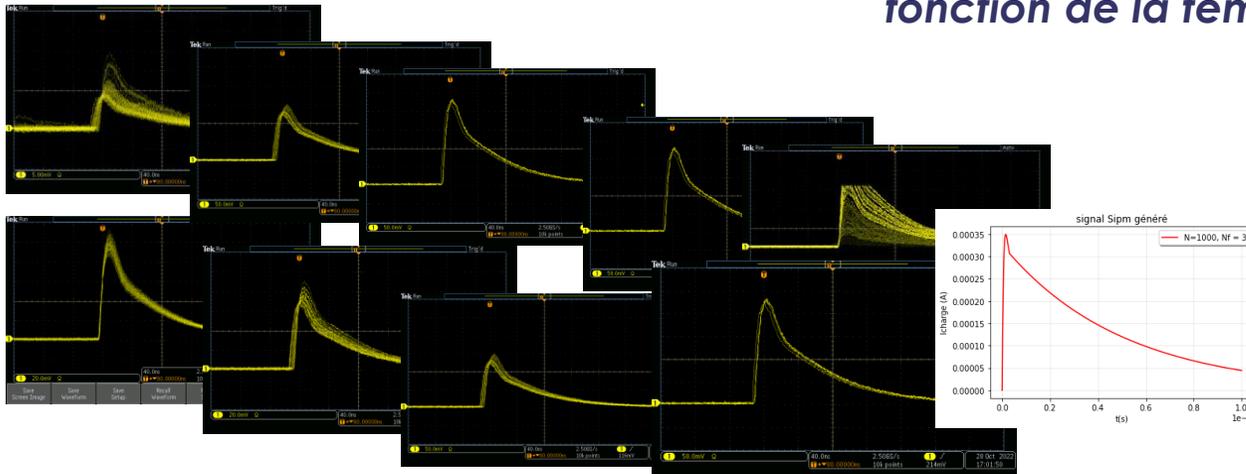
Mesures à différentes températures:  
gain,  
courant d'obscurité,  
bruit



Modélisation et simulation à différentes températures:  
gain,  
courant d'obscurité,  
bruit



*Un modèle d'IA pour prédire et corriger les variations de réponse en fonction de la température.*



Conditions environnementaux

# Conclusion

- ❑ Un modèle est une fonction dont les paramètres sont déterminés par les X et les Y ou uniquement les X
- ❑ On peut déduire les paramètres cachés des détecteurs
- ❑ On peut améliorer la résolution en Energie et en Temps en éliminant les parasites (crosstalk et bruit électronique)
- ❑ On peut réaliser de la déconvolution en ligne
- ❑ On peut classifier les signaux de façon automatique
- ❑ On peut simplifier l'étape de calibration

 **Il faut essayer pour améliorer l'efficacité des détecteurs**

Mesures – Simulation – Apprentissage - inférence