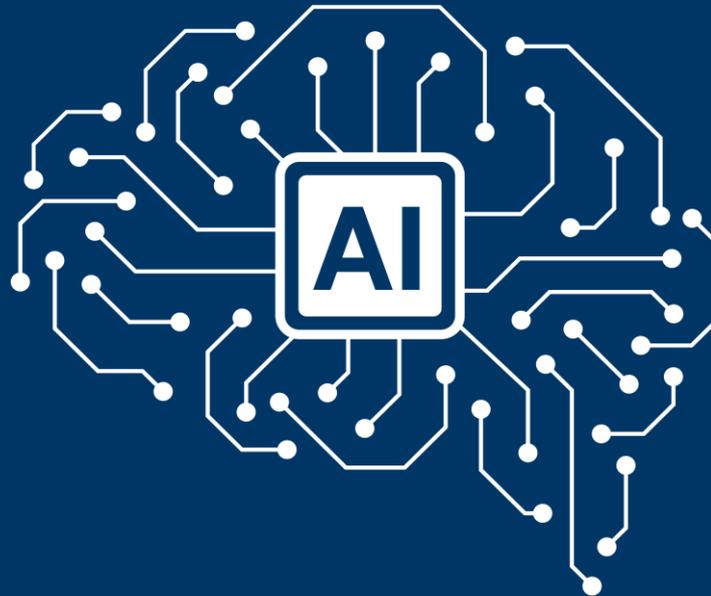


Intelligence Artificielle

Principes fondamentaux et utilisations

Mathieu Nierenberger

mathieu.nierenberger@ac-strasbourg.fr



Strasbourg - 2025



Plan

Introduction à l'IA

Notions de base de l'IA

Principaux algorithmes en IA

Et l'IA dans l'industrie ?

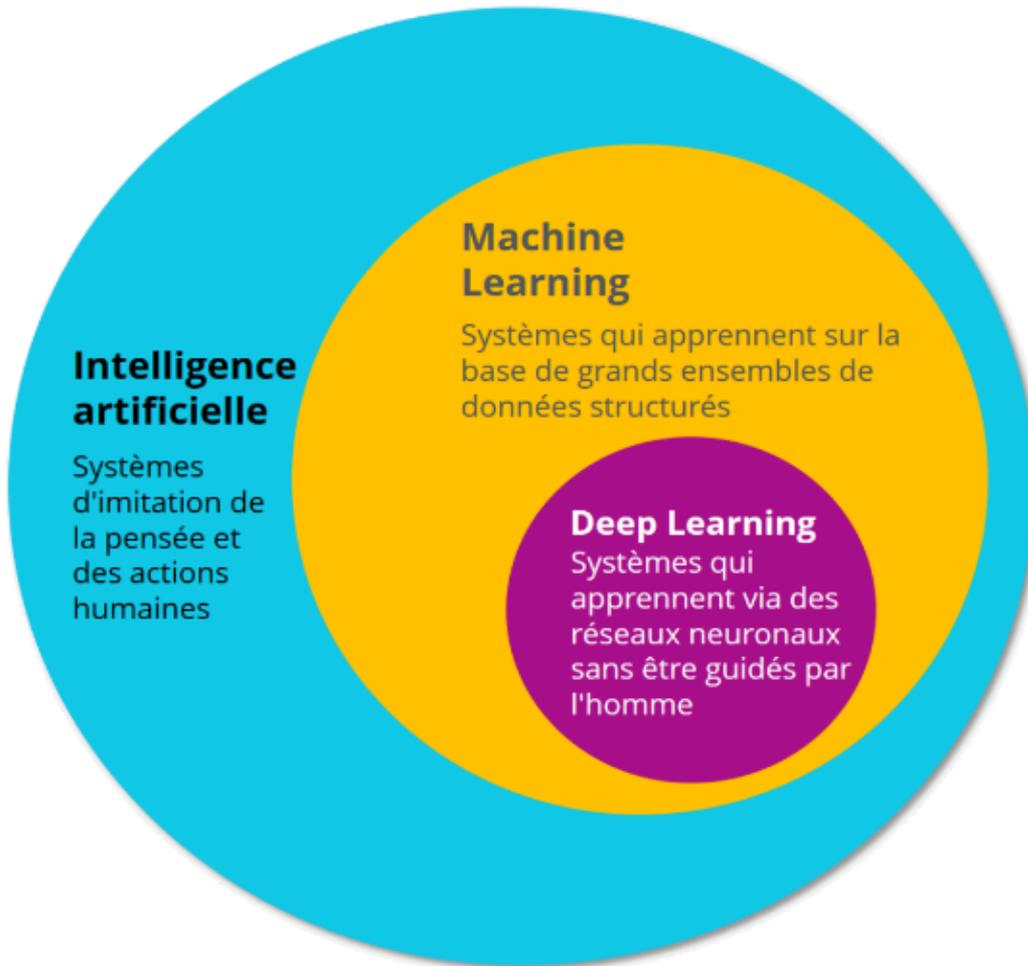


Introduction à l'IA

Définitions

Historique

Exemples d'utilisation de l'IA



L'intelligence artificielle (IA) est :

Définition Larousse :

L'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine.

Marvin Minsky (chercheur américain en IA) :

La construction de **programmes informatiques** qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique.

Historique



1950 - 1970

Naissance et développement de l'IA



1943

Premier réseau de neurones artificiel par Warren McCulloch et Walter Pitts

1950

Travaux de John Von Neumann et Alan Turing, création du Test de Turing

1956

Terme "intelligence artificielle" introduit à la conférence de Dartmouth.

1980 - 1990

"Hiver de l'IA" durant les années 1980 et début des années 1990



1996 - 1997

IBM Deep Blue bat Garry Kasparov aux échecs

2010

Essor de l'IA nouvelle génération



2011

IA Watson d'IBM remporte Jeopardy!

2012

Google X crée une IA de reconnaissance d'image



2016

Alpha Go de Google bat le champion du monde de go

HISTOIRE DE L'IA

Temps forts de 1940 à aujourd'hui



2023

Lancement de GPT-4, intégration de DALL-E dans ChatGPT Plus
Ouverture au public de Google Bard et de Bing Chat

2021

Introduction de Codex et DALL-E par OpenAI



2020

Démocratisation de l'IA
Lancement de GPT-3 par OpenAI

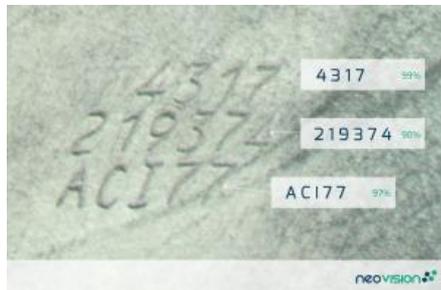
Source : Wizishop

Exemples d'utilisation de l'IA

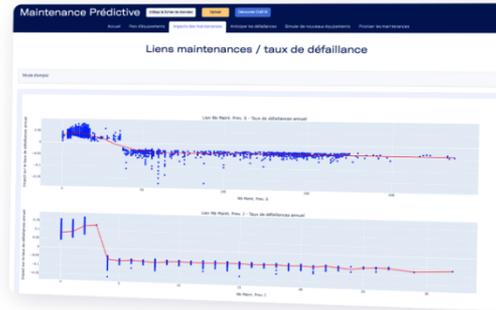


Utilisations de l'IA :

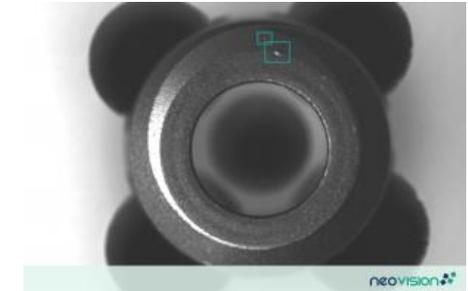
- assistants vocaux (Alexa, Google Home...)
- traduction automatique (DeepL)
- véhicules autonomes
- ChatGPT ...



Lecture automatique de références
(traçabilité de gravures sur cuir)



Maintenance prédictive



Détection automatique de défauts
(copeaux collés sur un implant chirurgical)



Aide au diagnostic en médecine :
Traitement de larges bases de
données multiparamétriques



Optimisation de
l'approvisionnement
d'une chaîne de magasins

Et encore bien d'autres !

Exemples d'utilisation de l'IA



IA génératives → Génération d'information

Exemples :

Prompt
= instruction (souvent
textuelle en 2024) destinée
à un algorithme d'IA



Gemini



Génération de texte / langage / code



Midjourney



Génération d'images



Génération de son / musique

+ Génération de vidéos

+ Génération de modèles 3D (ex : Leo – version alpha)

+ ...



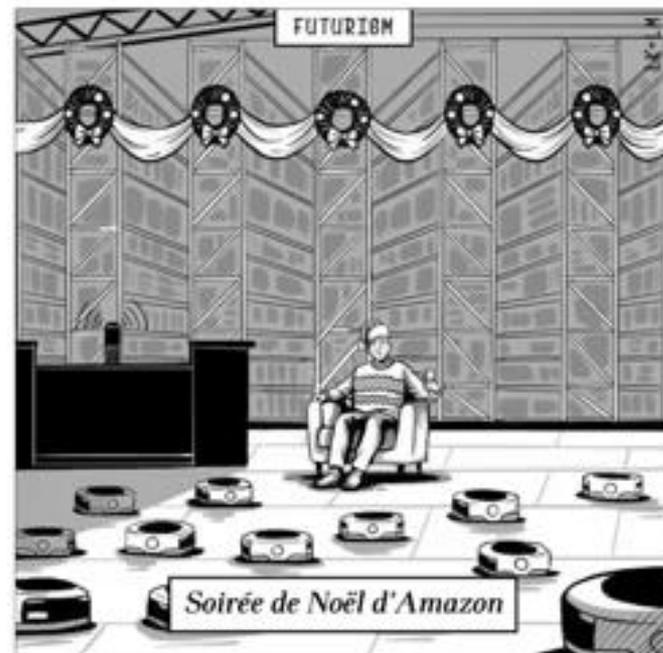
Extrait de la conférence de Laurent Haug – Digivisions 2024 – Université de Lausanne :

IA en entreprise

Du fantôme à la réalité

DIGIVISIONS2024

Laurent Haug
6 Septembre 2024
Université de Lausanne





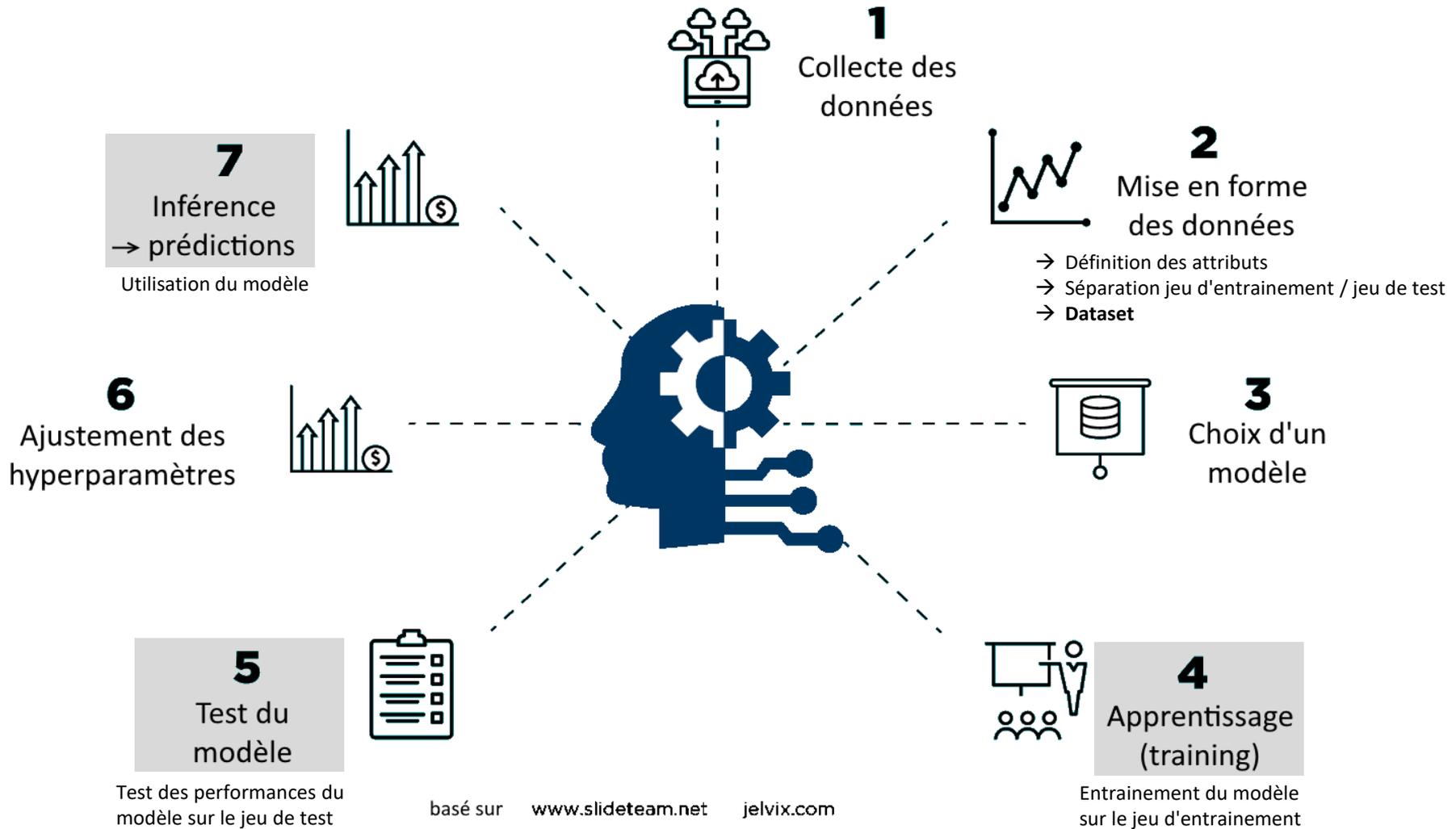
Notions de base de l'IA

Etapas clefs du machine learning

Problèmes de classification / régression

Apprentissage supervisé / non supervisé / renforcement

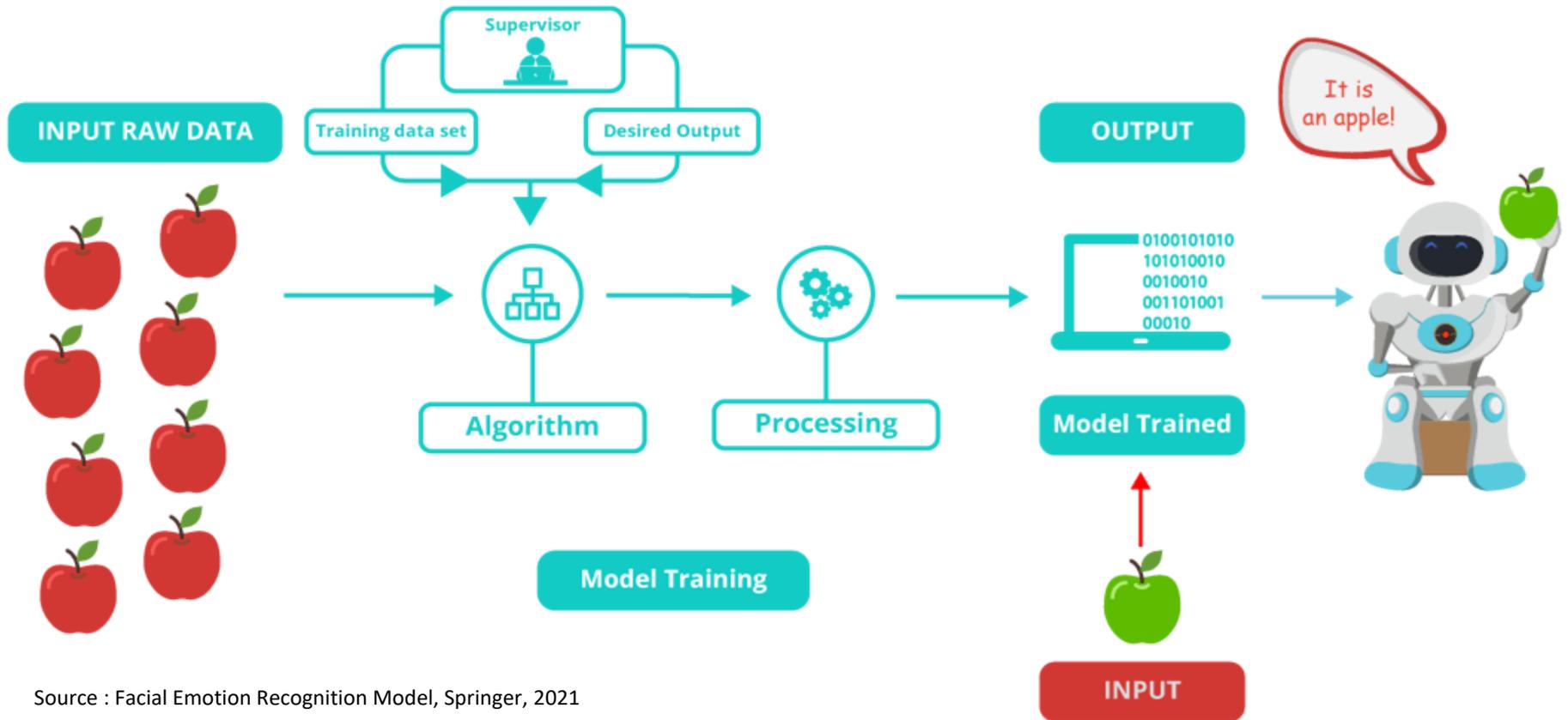
Etapes clés du machine learning



Etapes clefs du machine learning



Exemple :



Source : Facial Emotion Recognition Model, Springer, 2021

Problèmes de classification / régression



Problème de régression

→ On cherche à prédire une (plusieurs) valeur(s)
Températures, temps de réponse, tailles ...

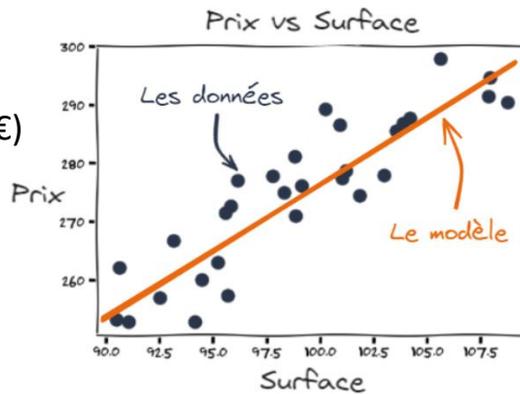


Exemple : Trouver le prix d'un logement en fonction de sa surface

Entrée : Surface du logement (m²)
Sortie : Prix de vente (k€)

	X	y
	92	259K
	95	237K
	98	272K
	93	234K
	100	252K
	110	300K

dataset



La machine va déterminer ("apprendre") une relation $y = f(X)$ entre l'entrée X et la sortie y .

→ Futures prédictions de prix

Problème de classification

→ On cherche à prédire l'appartenance à une classe
Animal, fleur, type de maladie/défaut ...

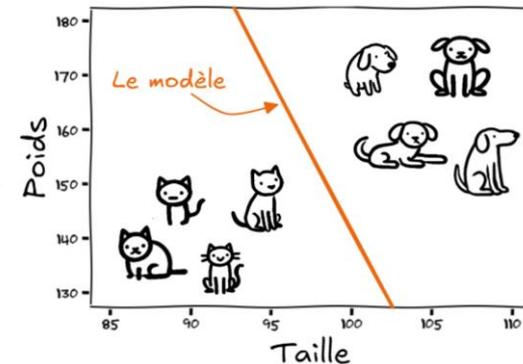
Exemple : Reconnaître un chat ou un chien en se basant sur sa taille et son poids



	Entrées X		Sortie y
	Taille	Poids	
	92	133	Chat
	105	170	Chien
	87	135	Chat
	103	155	Chien
	107	150	Chien

dataset

→ Futures prédictions de classe



La machine va déterminer une limite entre les classes de sortie y , basée sur les entrées X .

Nouvel Animal



Taille: 105
Poids: 140

Chat ou Chien?





Apprentissage supervisé

→ On donne à l'algorithme un dataset comportant les entrées X ET les sorties y associées

Les ex. précédents correspondent à de l'apprentissage supervisé

Apprentissage non supervisé

→ On laisse la machine trouver des similitudes entre les données d'entrée X pour les organiser en groupes. Pas de sorties y dans le dataset.

Adapté à des problèmes de classification uniquement

Exemple : on cherche à classer un ensemble d'objets

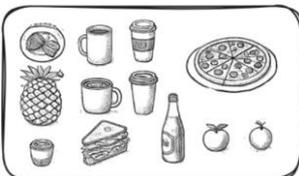
On impose les classes de sortie y :



y_1 = Livres



y_2 = Appareils électroniques



y_3 = Nourriture



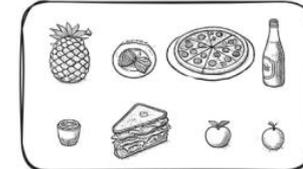
On laisse la machine trouver des similitudes entre les données d'entrée :



Regroupements décidés par la machine, sans donner de nom aux groupes.



Par ex. : les tablettes ont été regroupées avec les livres car elles permettent le même type d'activité



Source des illustrations : Guillaume Saint-Cirgue, 2024 un dataset comport



Exemples d'utilisations :

Apprentissage supervisé

Clustering in market segmentation



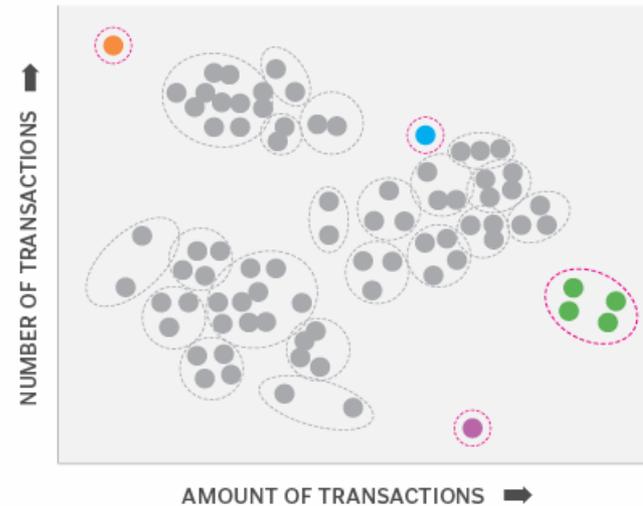
Cluster 1: High income/high property value

Cluster 2: Middle income/middle property value

Cluster 3: High income/low property value

Apprentissage non supervisé

Clustering in fraud detection



Outlying points that could be scrutinized for fraud

Source : Patrick Hall, TechTarget, 2023



Utilisation en maintenance dans l'industrie (voir TP) :

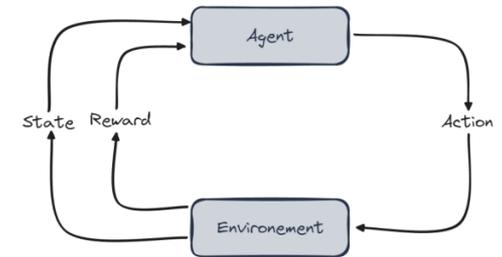
- Apprentissage supervisé → Détection des modes de marche standard
- Apprentissage non supervisé → Détection d'anomalies



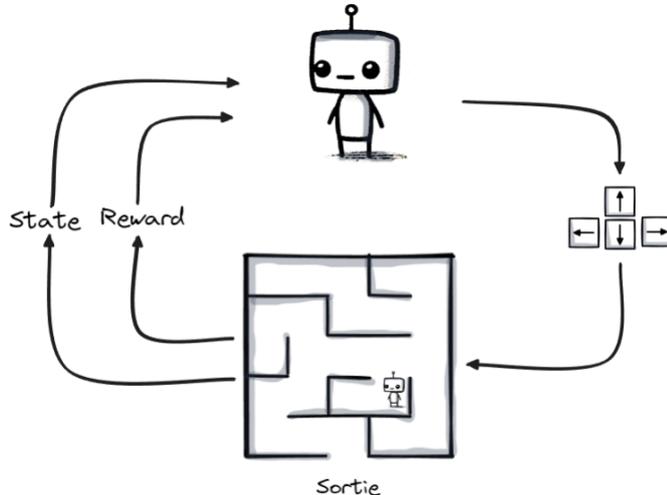
Apprentissage par renforcement

Troisième branche du machine learning → La machine (agent) génère ses propres données au sein d'un environnement imposé, et apprend de ses expériences via un système de récompenses

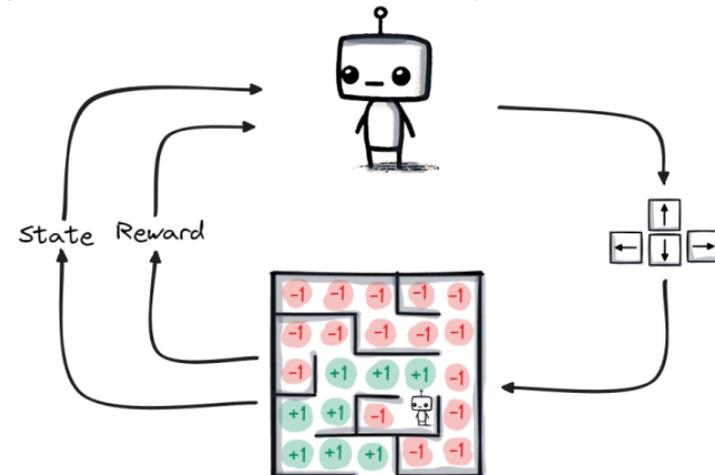
→ Utilité pour des problèmes avec séquences ordonnées : prise de décision, robotique, voitures autonomes, language processing (traitement et traduction)...



Exemple : un personnage (agent) doit sortir d'un labyrinthe (environnement imposé)



La machine peut déplacer le personnage dans toutes les directions qu'elle souhaite



A chaque déplacement, l'agent reçoit une récompense liée à la pertinence de son choix. Il apprend par essai / erreur.

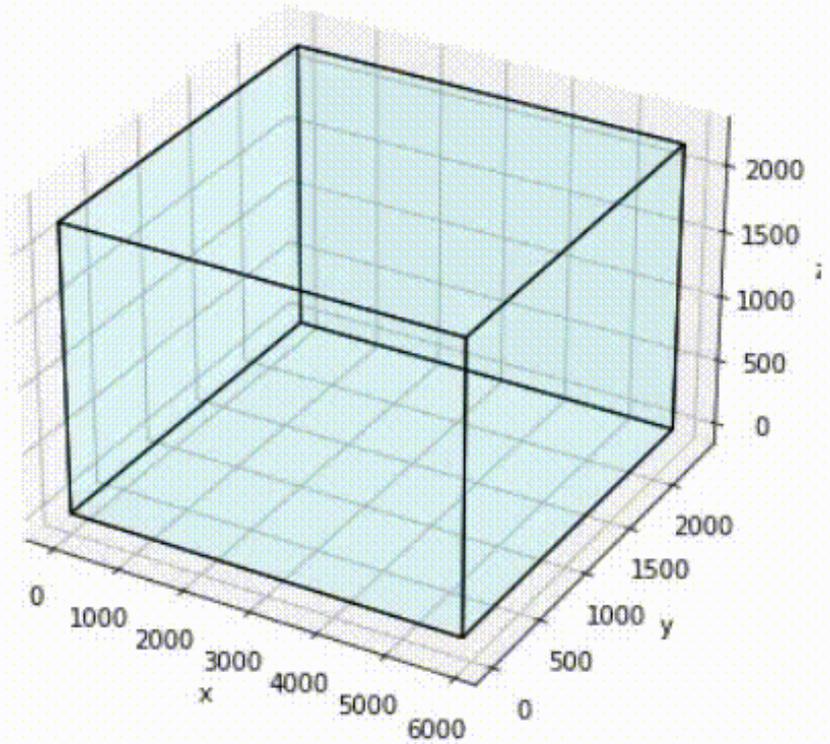


Exemple d'utilisation de l'apprentissage par renforcement :
optimisation du placement de cartons sur une palette (bin packing 3D)



→ Minimisation du volume à transporter

Placed: 0/20 | Used Volume: 0.0%



Source : LauraKarimova, Big_Data_Research_Project, 2023





Principaux algorithmes en IA

Régression (supervisée)

Classification (supervisée)

Classification (non supervisée)

Détection d'anomalie

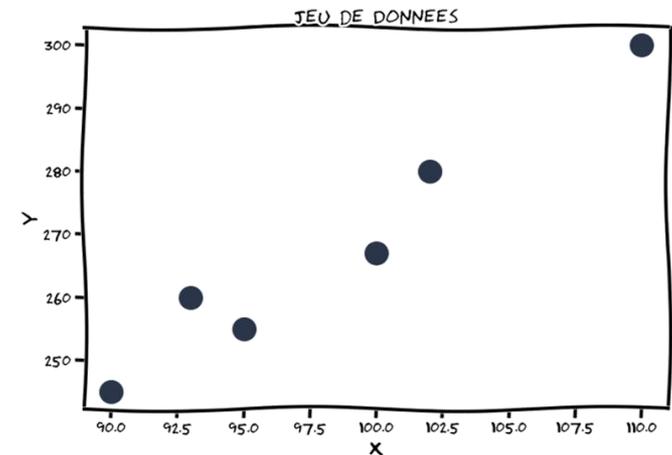
Deep Learning : Réseaux de neurones

Note : Il existe pour chaque type de problème une multitude d'algorithmes de résolution dans la littérature. Seuls les plus classiques sont présentés ici.



Remarque : On présente ici le cas d'une régression **linéaire**, mais on peut effectuer une régression avec d'autres modèles. On est également dans le cas d'un apprentissage supervisé.

Reprenons le problème précédent :
Trouver le prix d'un logement en fonction de sa surface



On dispose d'un jeu de données (dataset) qui à une surface X en m² associe un prix de vente y en k€.

On va ici chercher un modèle, caractérisé par les coefficients $\{a, b\}$ tel que :

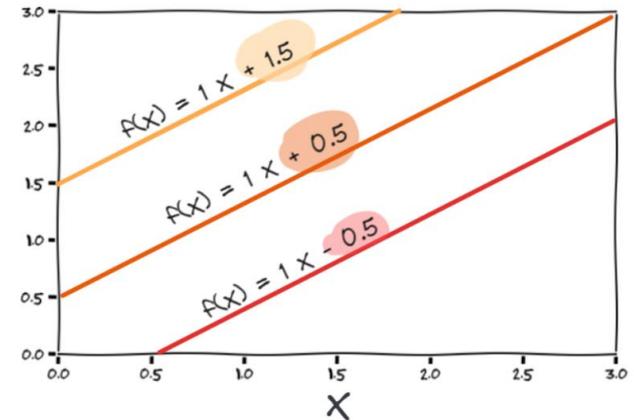
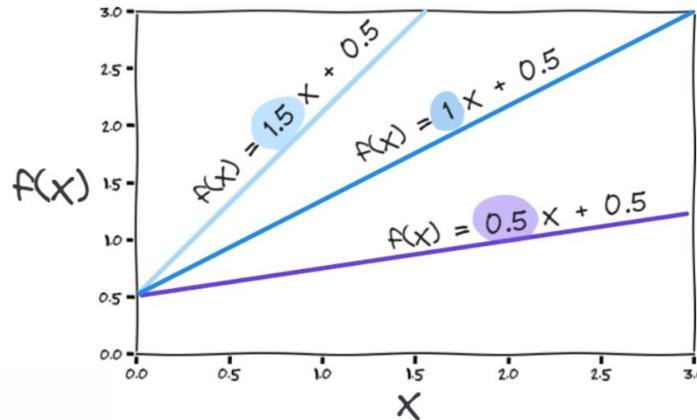
$$y = aX + b \text{ (si régression linéaire)}$$

Le problème consiste à trouver $\{a, b\}$ optimaux pour le dataset.

Régression (supervisée)



Pour rappel :



Trouver $\{a, b\}$ optimaux = minimiser l'erreur \rightarrow Fonction coût à minimiser

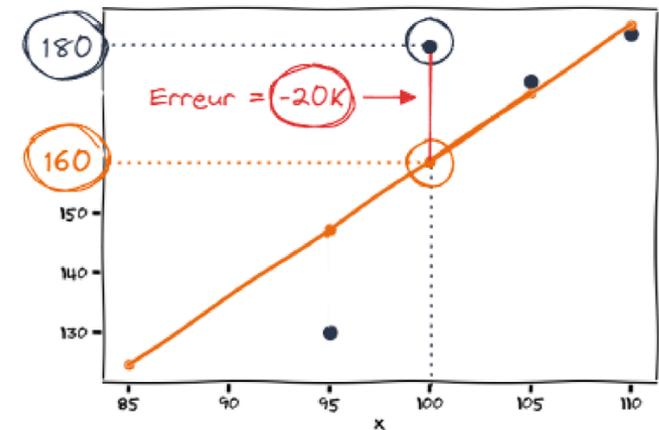
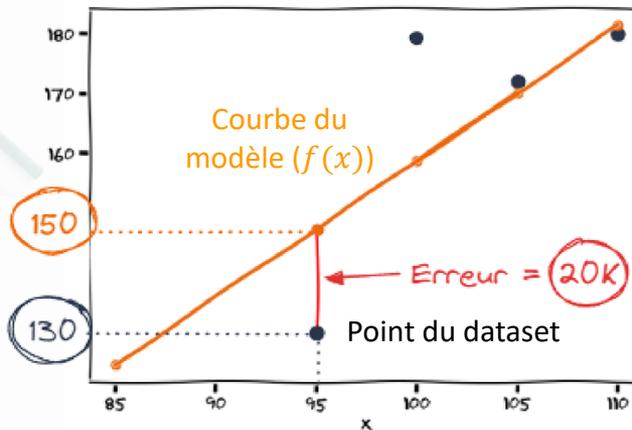
Erreur (résidu) :

$$R_i = f(x_i) - y_i$$

pour un point (x_i, y_i)
du dataset

Fonction coût = somme
des erreurs au carré
(carré \rightarrow perte du signe) :

$$L(a, b) = \sum_i (f(x_i) - y_i)^2$$
$$= \sum_i ((ax_i + b) - y_i)^2$$



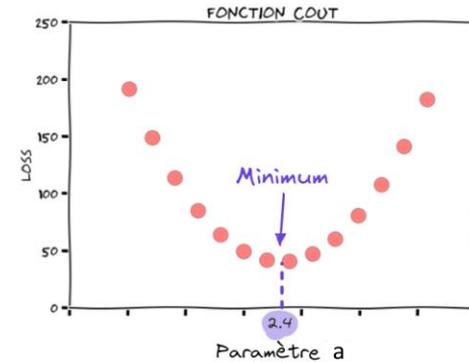
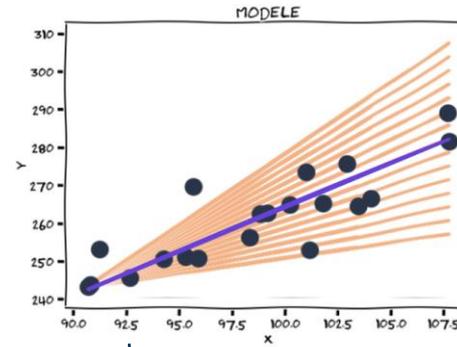
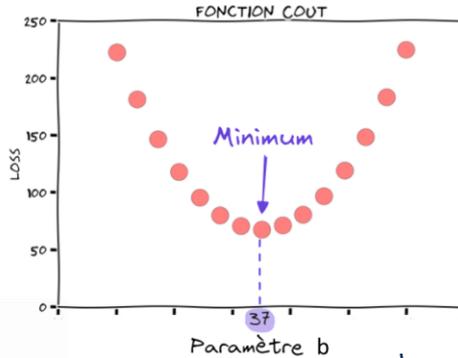
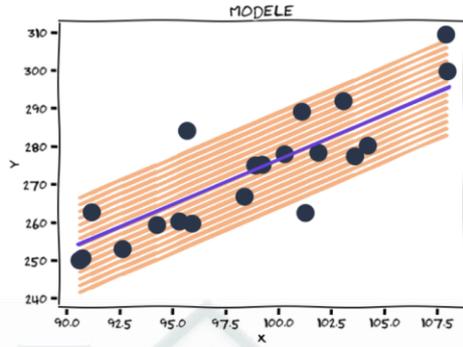
Trouver $\{a, b\}$ optimaux \leftrightarrow Minimiser la fonction coût $L(a, b)$

Régression (supervisée)



Influence de b sur la fonction coût $L(a, b)$

Influence de a sur la fonction coût $L(a, b)$



$L(a, b)$



Interdépendance des paramètres sur la fonction coût $L(a, b)$.

➔ Recherche du minimum global

Méthodes pour la recherche du minimum :

- Calcul de $L(a, b)$ pour un grand nombre de combinaisons (a, b)
trop grand nombre de calculs nécessaires si le problème est complexe
- Utilisation d'un algorithme de descente de gradient

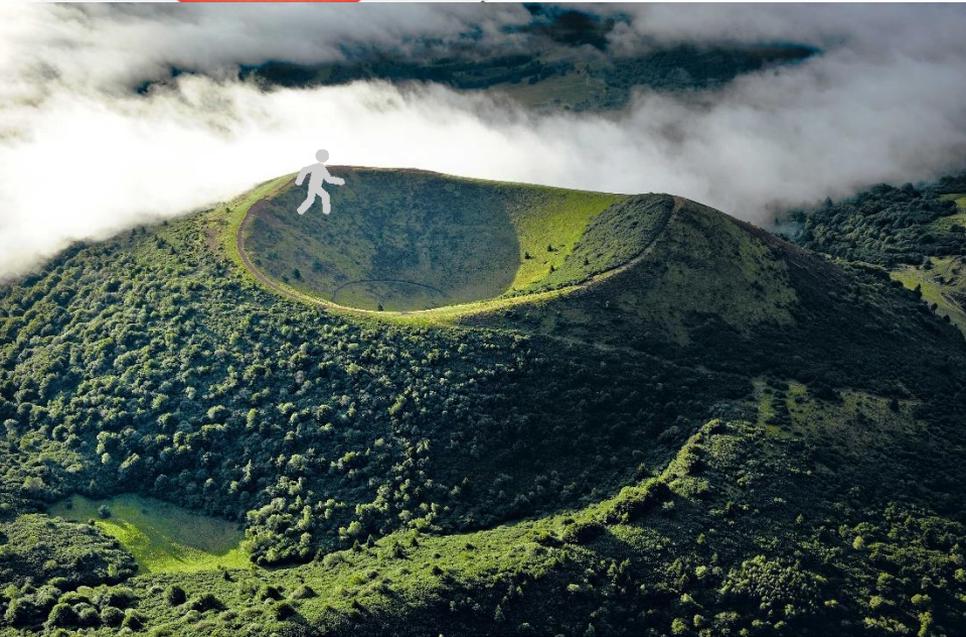
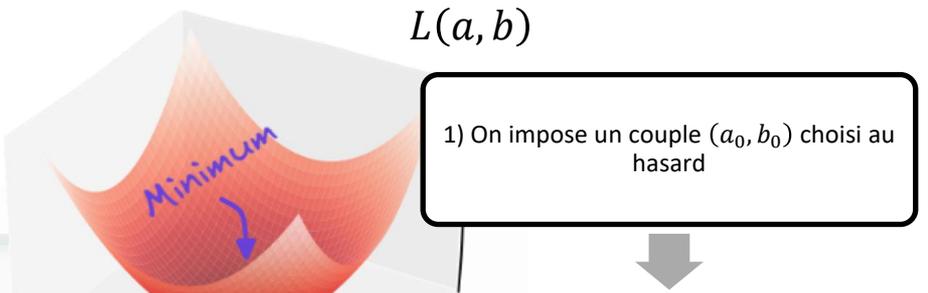
Rq : - Utilisation possible pour plus de 2 paramètres : $L(a_1, a_2, \dots, a_n)$
- Utilisation possible pour d'autres modèles (non linéaires)

Source des illustrations : Guillaume Saint-Cirgue, 2024

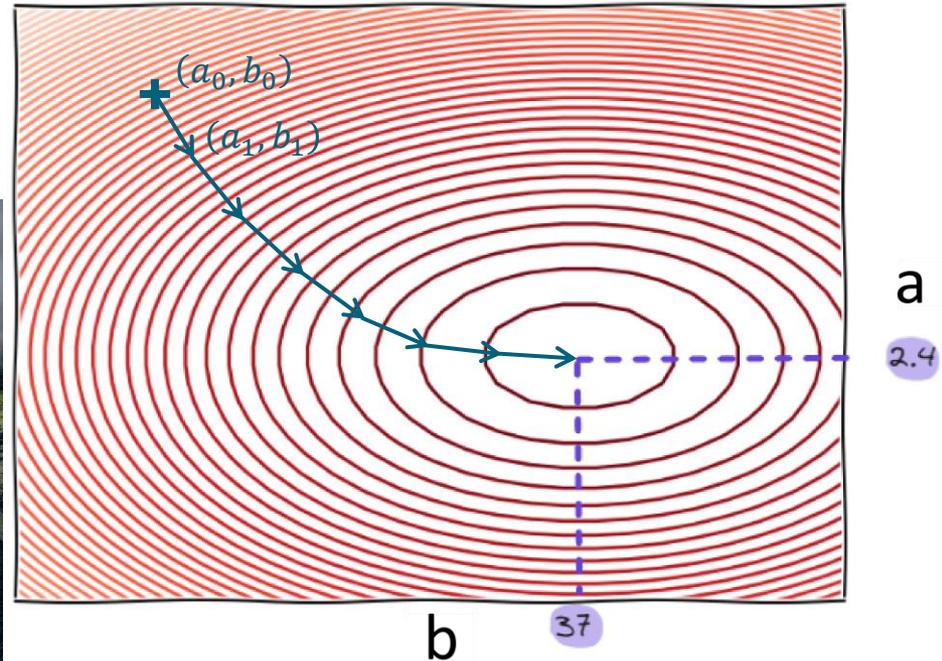


Algorithme de descente de gradient :

Plutôt que de tester un grand nombre de combinaisons (a, b) , utilisons une méthode plus intelligente.



Fonction coût, isolignes en fonction de (a, b)
(= vue de dessus du graphique 3D ci-contre)



- ➔ Détermination "directe" des paramètres optimaux
- ➔ Nécessité d'avoir un seul minimum global (1 seul creux, pas de minimums locaux de $L(a, b)$)

Source des illustrations : Guillaume Saint-Cirgue, 2024



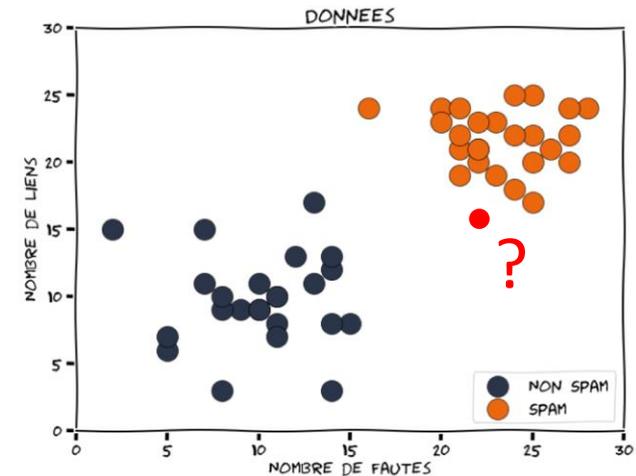
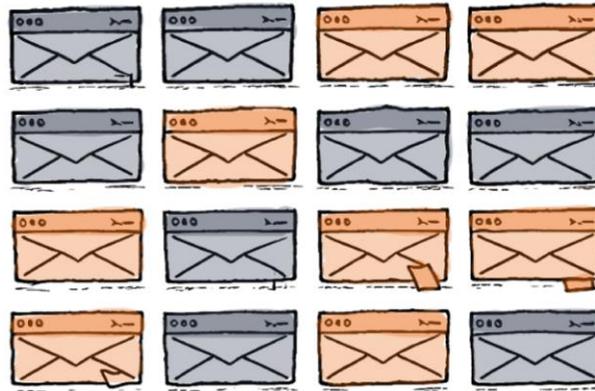


Remarque : On présente ici une classification avec 2 classes mais on pourrait l'étendre à d'avantage de classes. On est également dans le cas d'un apprentissage supervisé.

Considérons le problème suivant :

Classer des emails en spam / non-spam en se basant sur :

- Le nombre de liens qu'il contient
- Le nombre de fautes d'orthographe dans le texte



On dispose d'un dataset de donnée déjà étiquetées ("labels" en anglais) qui aux données d'entrée X (nb. liens, nb. fautes) fait correspondre une classe de sortie y (spam ou non-spam).

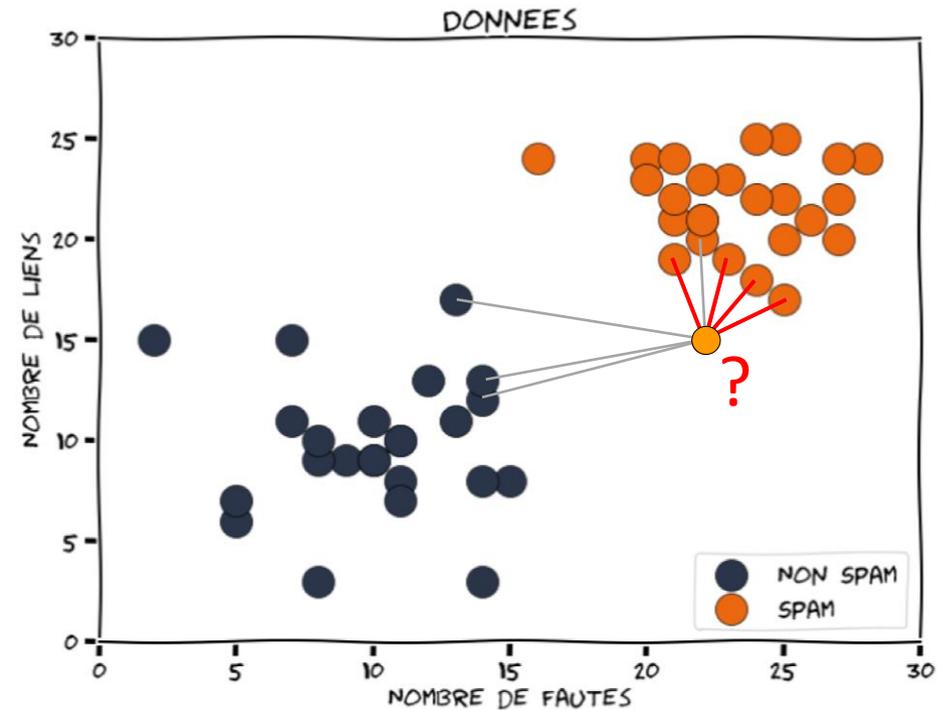
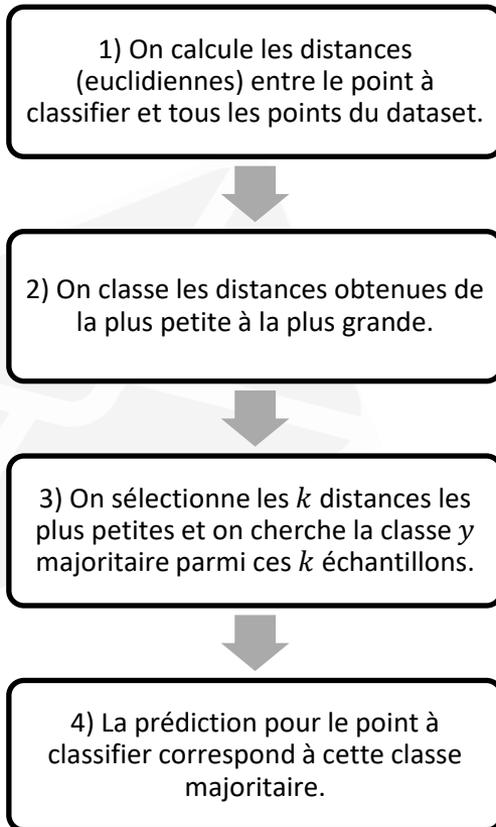
L'algorithme de classification devra être capable de prédire l'appartenance d'un nouvel email à une des classes de sortie.

Source des illustrations : Guillaume Saint-Cirgue, 2024

Classification (supervisée)



La méthode la plus utilisée pour résoudre ce problème est celle des **plus proches voisins** :
kNN = k Nearest Neighbors



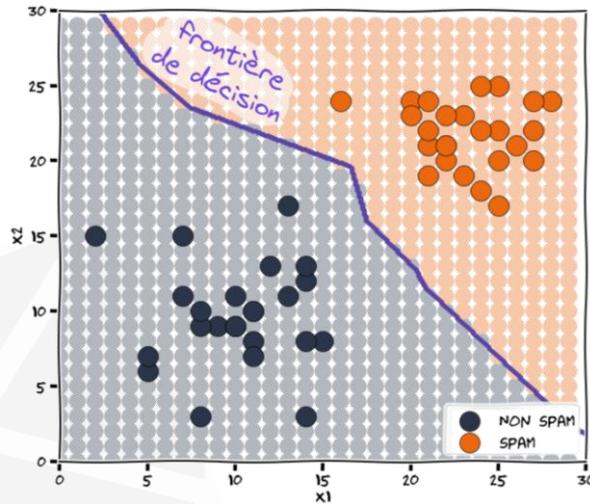
Calcul d'une distance euclidienne entre un point i et un point j :

$$d_{ij}^2 = (x_j - x_i)^2 + (y_j - y_i)^2$$

Classification (supervisée)

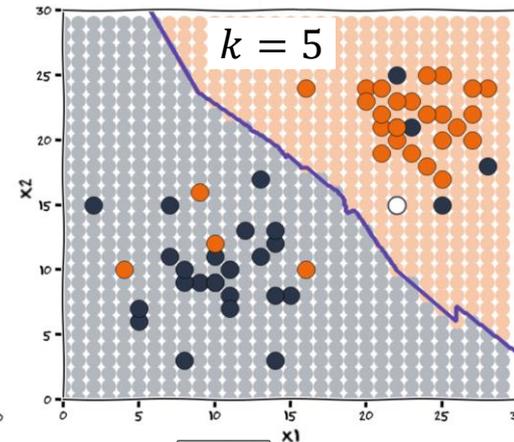
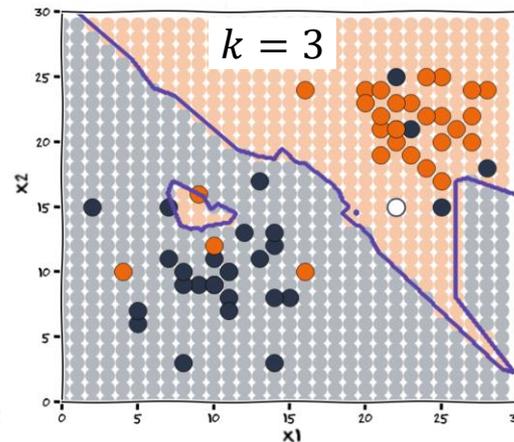
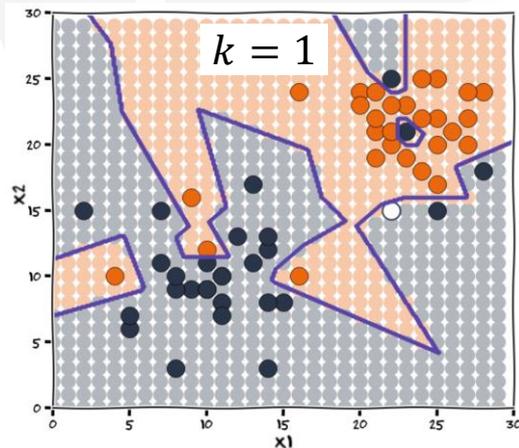


En appliquant ce raisonnement à tous les points de l'espace (étape très gourmande en calculs), on peut définir une **frontière de décision** : limite entre 2 classes.



→ L'algorithme de prédiction est ensuite très rapide à utiliser, et demande peu de ressources.

La frontière a ici été obtenue avec $k = 1$ car le dataset est très peu bruité. L'intérêt de choisir judicieusement $k > 1$ est montré ci-dessous.



$k \nearrow \rightarrow$ Lissage de la frontière de décision.

Attention à ne pas "gommer" les détails du dataset → Choisir judicieusement k en fonction du dataset

Source des illustrations : Guillaume Saint-Cirgue, 2024





Il peut être intéressant d'effectuer une classification sans avoir à étiqueter des classes de sortie y :
On laisse l'algorithme trouver les similitudes entre les données du dataset d'entrée $X \rightarrow$ **Clusters**

Méthode de k -means clustering

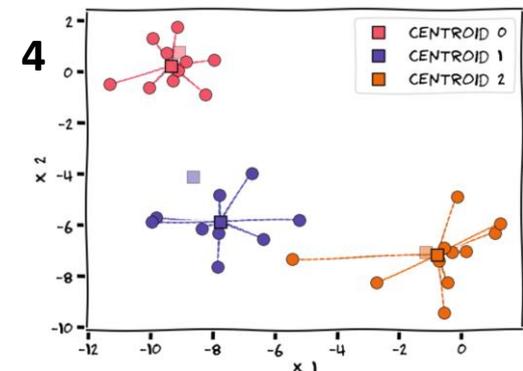
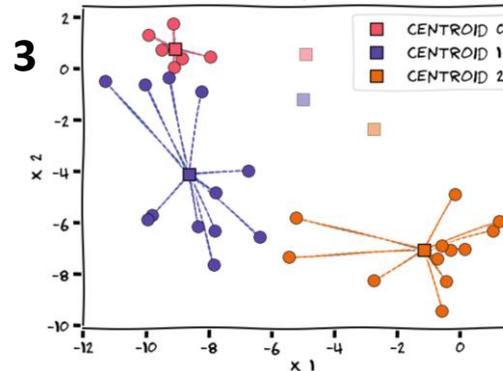
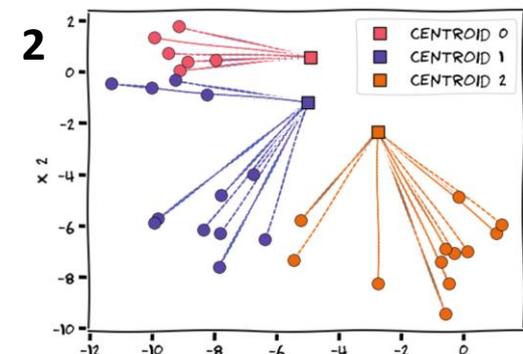
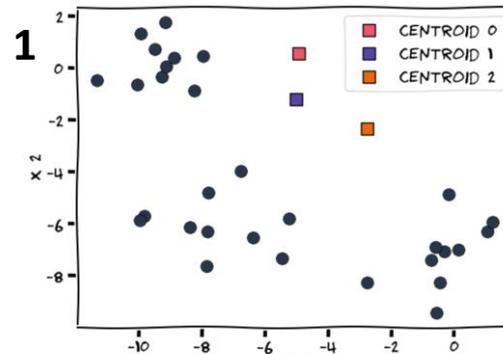
= algorithme capable de déterminer des frontières de décision entre k clusters

1) On place $k = 3$ (choisi) points aléatoires ("centroids") qui deviendront les barycentres des 3 clusters.

2) On associe chaque point du dataset à un des "centroids" en calculant les distances euclidiennes.

3) On déplace les k points au barycentre de chaque ensemble déterminé.

4) On réitère les étapes 1 à 3 car les points les plus proches donc les clusters ont changé. On s'arrête à convergence



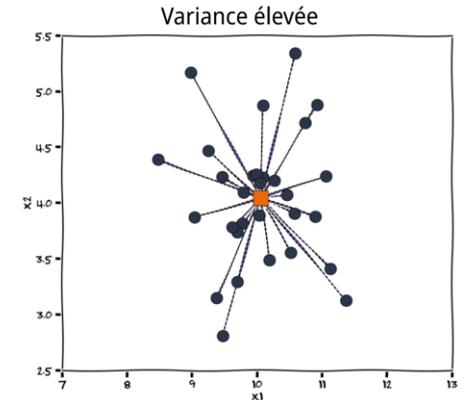
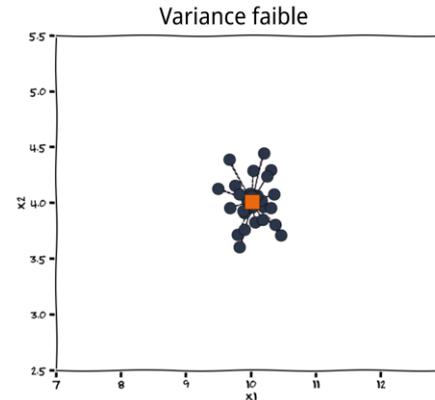
Reste à voir comment déterminer le nombre de clusters.

Classification (non supervisée)

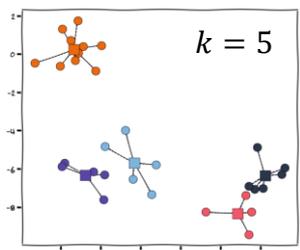
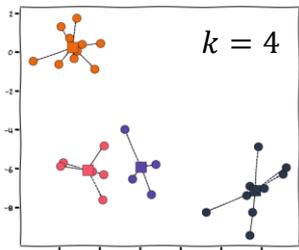
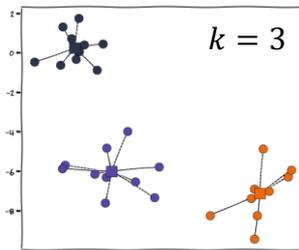
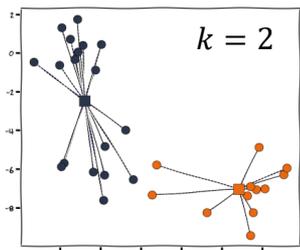
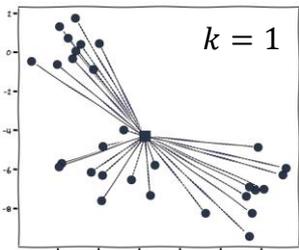


Détermination du nombre de clusters – méthode du coude :

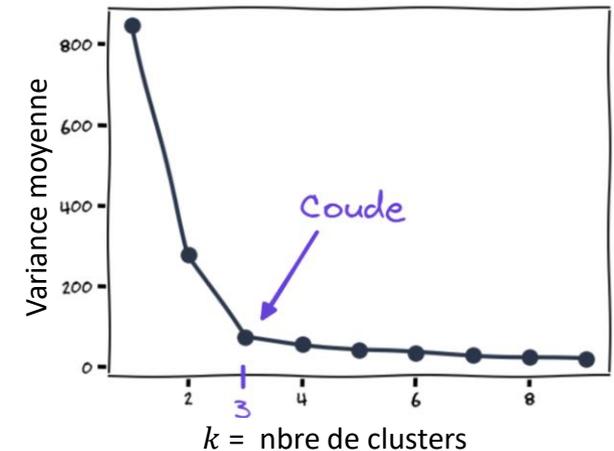
Rappel : notion de variance = étalement d'une variable autour de sa valeur moyenne



On détermine la variance moyenne entre les points des clusters et leurs barycentres pour différents k :



La courbe ci-contre présente un "coude" qui correspond au nombre de clusters k optimal.



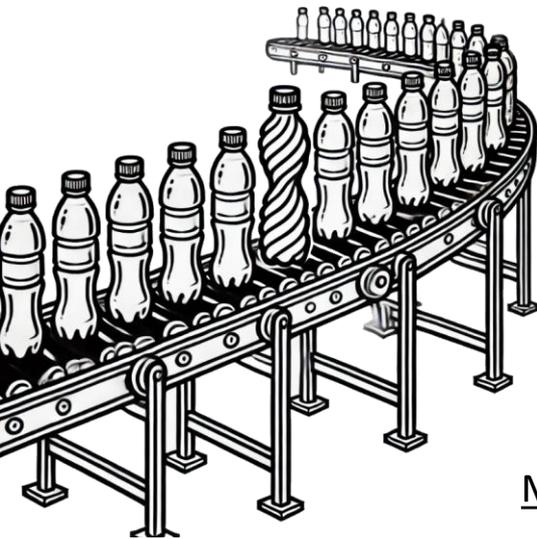
Source des illustrations : Guillaume Saint-Cirgue, 2024



Détection d'anomalie

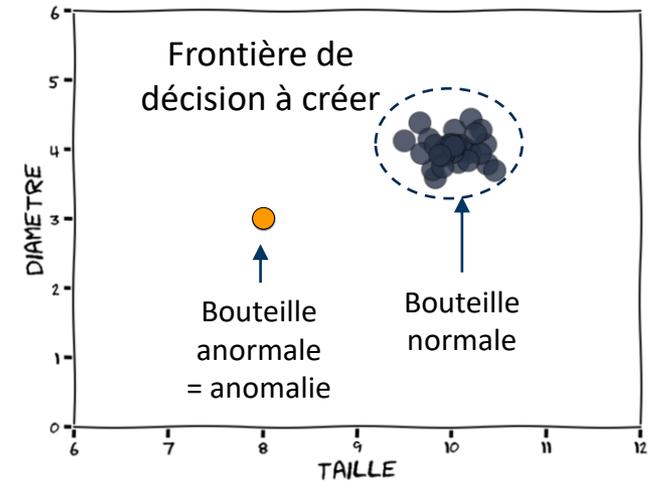


Détecter des anomalies revient à détecter des différences notables par rapport à un ensemble de clusters dit "normal". On peut l'utiliser en entreprise pour détecter un dysfonctionnement par exemple.



Exemple : une ligne de nettoyage de bouteilles consignées. On vérifie que l'on a bien affaire au bon modèle de bouteilles.

- Mesure de diamètre
- Mesure de hauteur (taille)
- ...

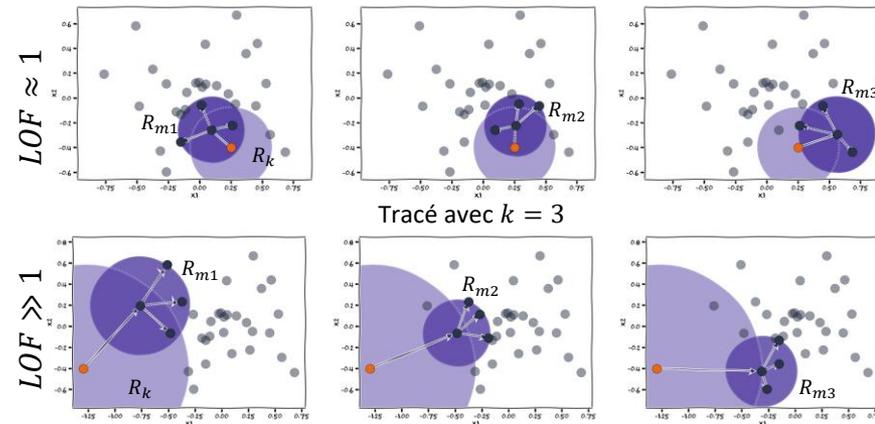


Méthode du Local Outlier Factor (LOF) :

1) On considère un point : on cherche les distances avec ses k (ici 3) plus proches voisins. La plus grande des distances est appelée distance- k , notée R_k .

2) Pour chacun des k -voisins, on réitère l'étape 1, et on cherche la moyenne des distances- k obtenues, notée R_m .

3) On détermine le LOF : $LOF = \frac{R_k}{R_m}$.
 $LOF \approx 1$: le point n'est pas une anomalie. $LOF \gg 1$: le point est une anomalie.



Source des graphiques : Guillaume Saint-Cirgue, 2024

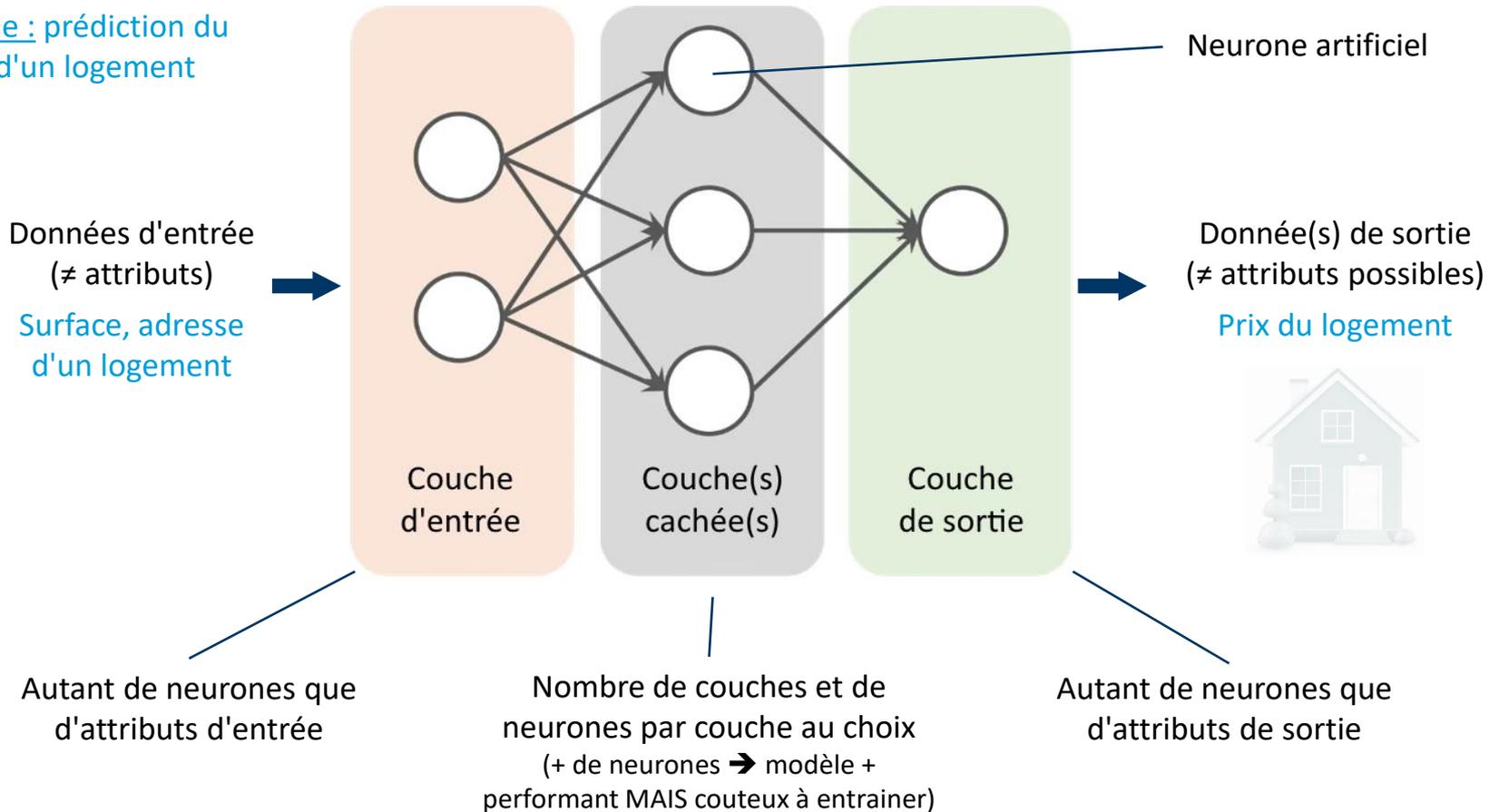


Deep Learning : Réseaux de neurones



Les réseaux de neurones peuvent être utilisés pour résoudre des problèmes variés (régression, classification, clustering, apprentissage par renforcement). **Ils interviennent dans toutes les IA génératives.**

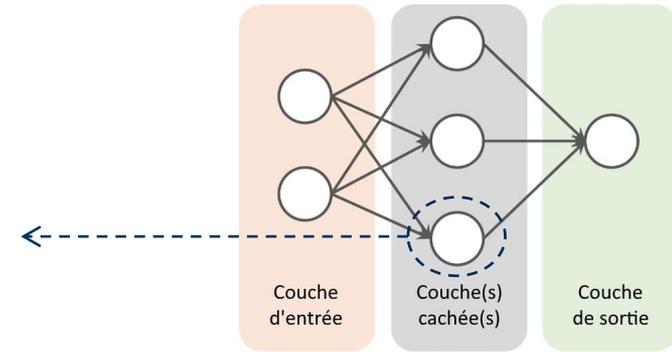
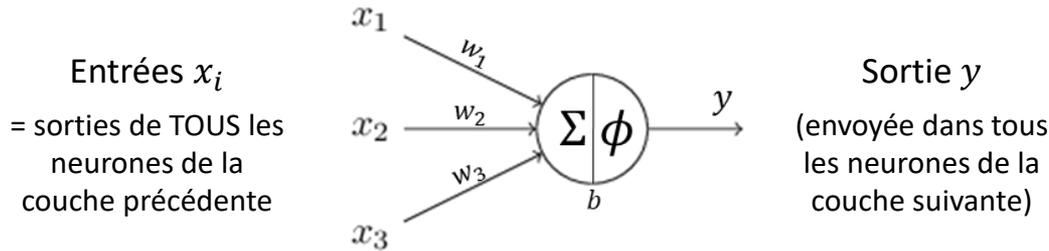
Exemple : prédiction du prix d'un logement



Deep Learning : Réseaux de neurones

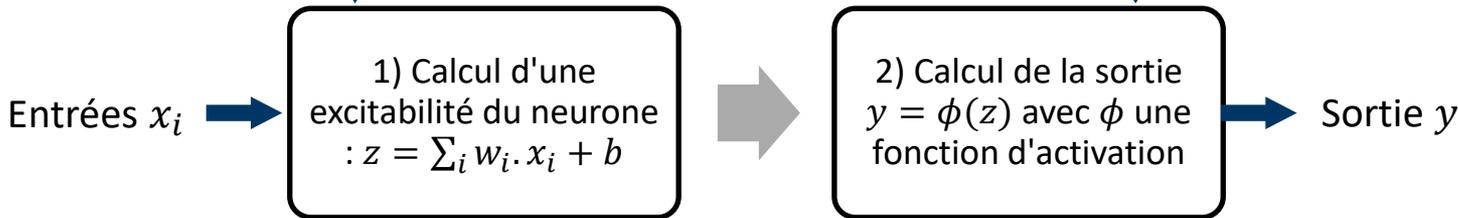


Qu'est-ce qu'un neurone artificiel ? Modèle du perceptron

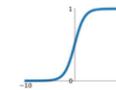


Le perceptron comprend des paramètres réglables :

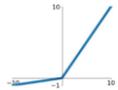
- Les poids w_i (autant que d'entrées x_i)
- Le biais b
- Le choix du type de fonction d'activation



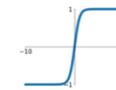
Sigmoid
 $\sigma(x) = \frac{1}{1+e^{-x}}$



Leaky ReLU
 $\max(0.1x, x)$

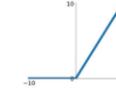


tanh
 $\tanh(x)$

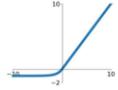


Maxout
 $\max(w_1^T x + b_1, w_2^T x + b_2)$

ReLU
 $\max(0, x)$



ELU
 $\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



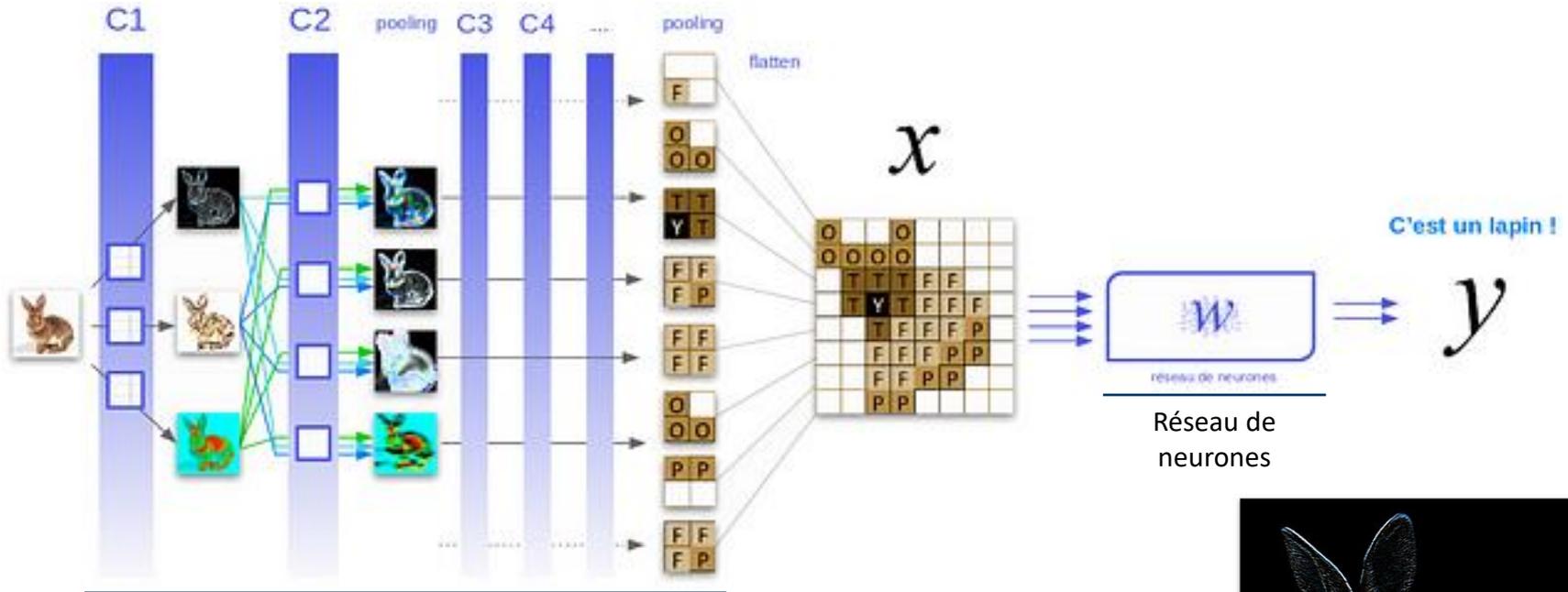
Les poids w_i et biais b des neurones sont **déterminés lors de la phase d'apprentissage** en utilisant une méthode de descente de gradient démarrant par la dernière couche (back-propagation).

Deep Learning : Réseaux de neurones



Différentes architectures de réseaux de neurones :

- Réseaux de neurones de convolution (CNN) → Analyse et/ou tracking sur images



Filtres de convolution pour faire ressortir différents aspects de l'image + pooling (= simplification)



Exemple de l'application d'un filtre de convolution "vers le haut" sur une image

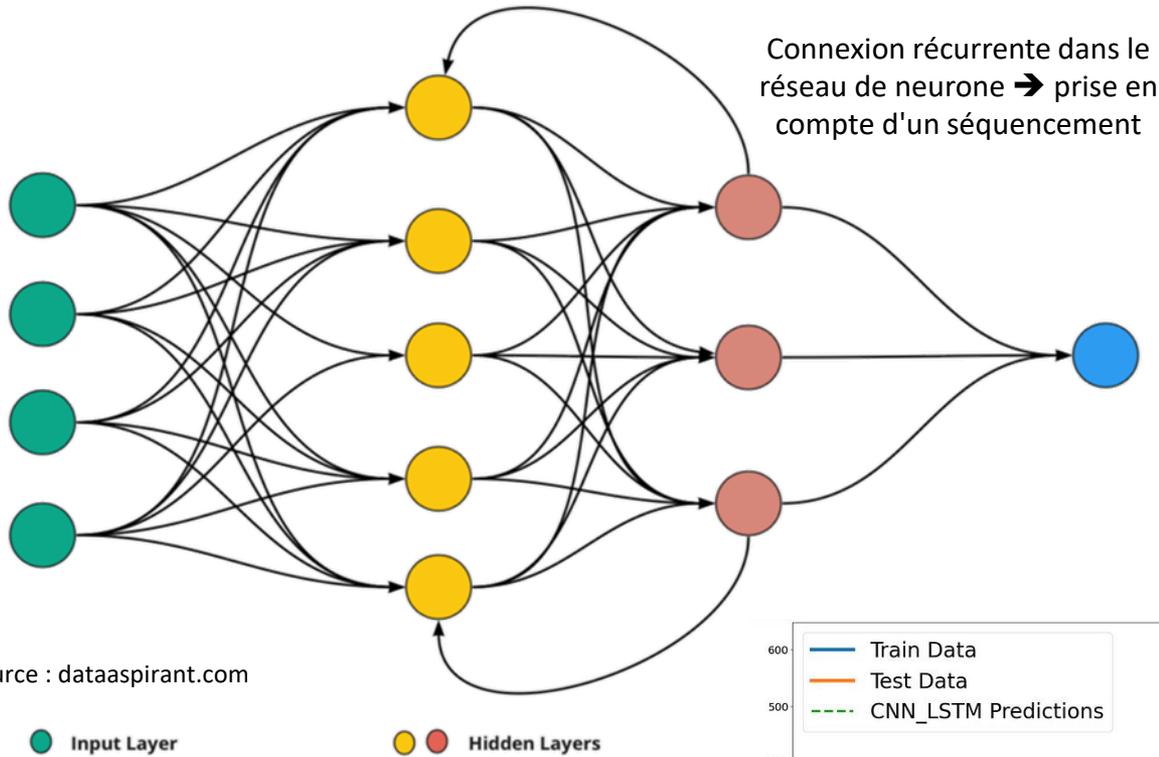


Source : medium.com - Les réseaux de neurones de convolution pour les néophytes



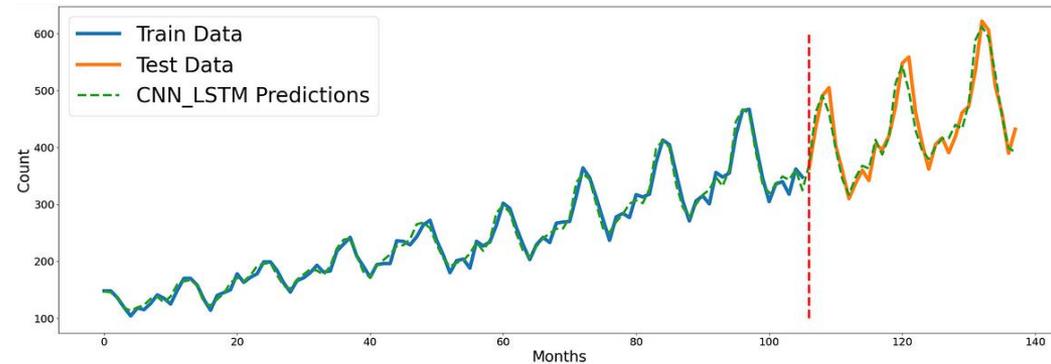
Différentes architectures de réseaux de neurones :

- Réseaux de neurones récurrents (RNN) → Analyse de données séquencées (série temporelle, texte...)



Source : dataaspirant.com

Source : medium.com

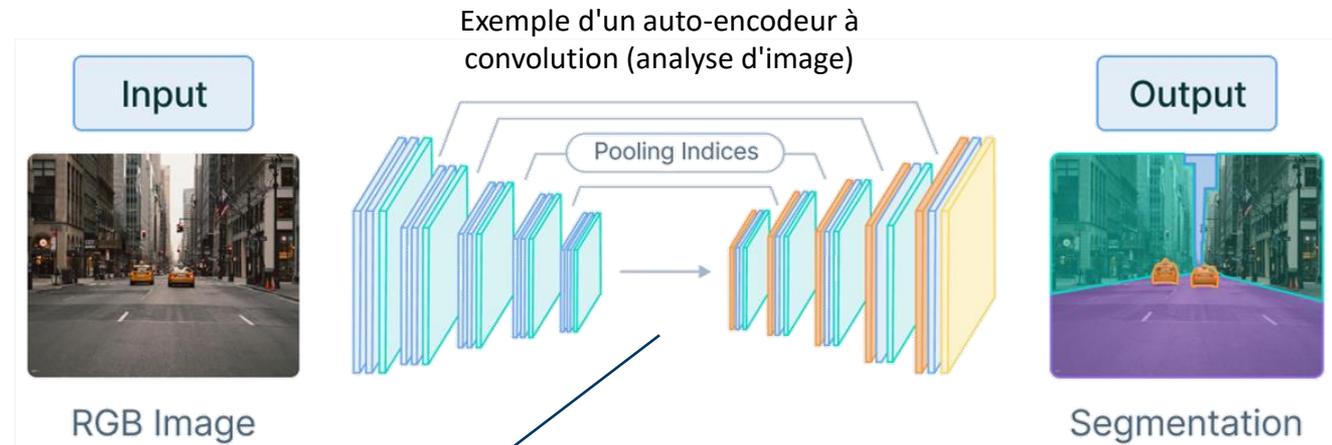


Deep Learning : Réseaux de neurones

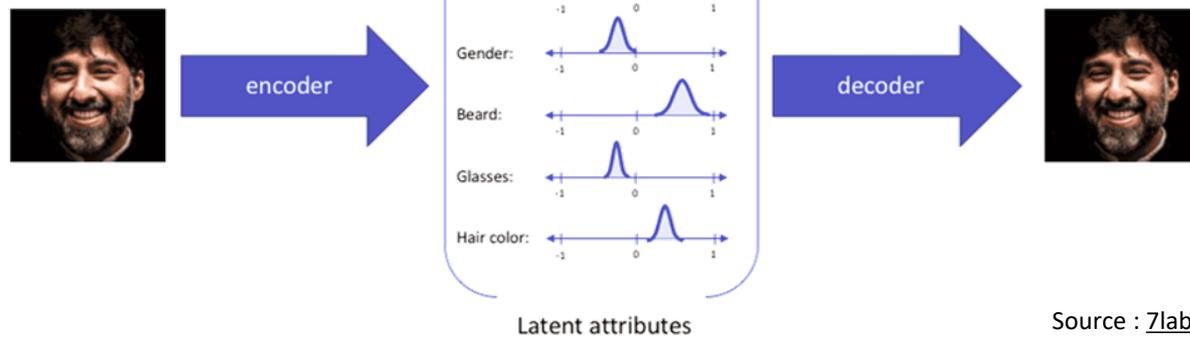


Différentes architectures de réseaux de neurones :

- Auto-encodeurs → Clustering / segmentation, réduction de dimension, deepfakes



Encodeur → Réduction de dimension



Source : 7labs.com - Autoencoders in Deep Learning: Tutorial & Use Cases

Deep Learning : Réseaux de neurones



Et ChatGPT ? → Architecture nommée **transformer**, proche des auto-encodeurs

Inventé en 2017 par Google – architecture similaire pour tous les Large Language Models actuels

Exemple : traduction

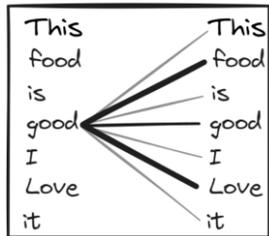
This food is good. I Love it



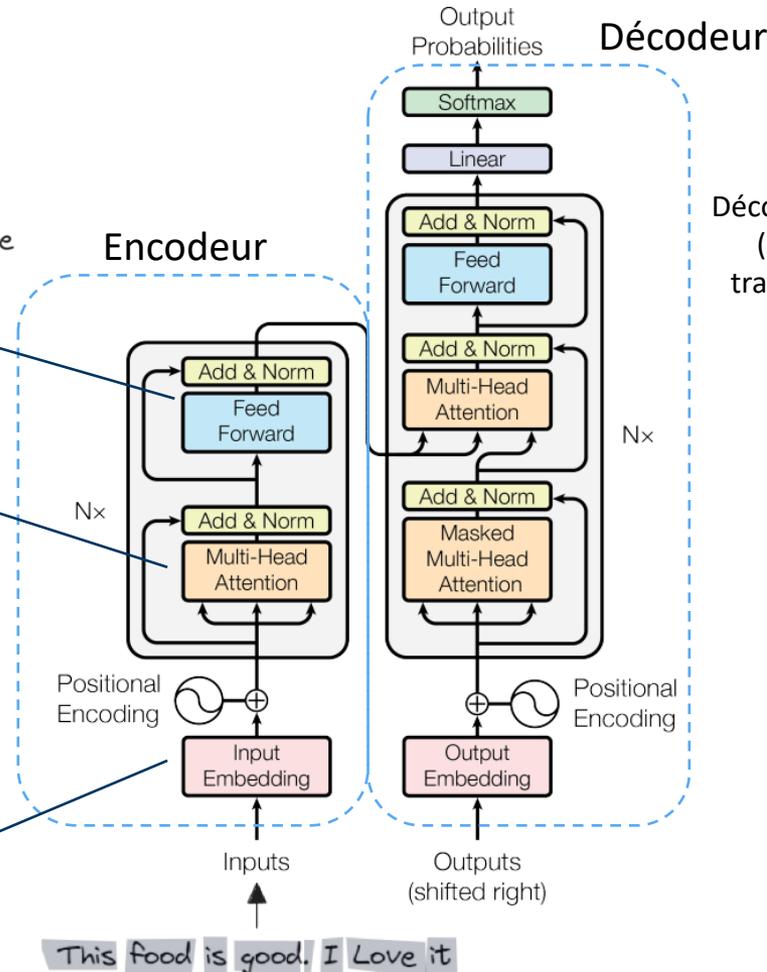
Cette nourriture est bonne. Je l'adore

Encodage de la phrase
(dimension réduite) : sens,
connexions des mots...

Mise en évidence des
liens entre les mots :



Conversion en vecteurs =
représentation numérique des mots



Décodage = générateur de texte
(dans une autre langue si
traduction) : fonctionnement
similaire à l'encodeur

Nombres de paramètres (w, b) :

- GPT-1 : 117 millions
- GPT-2 : 1,5 milliards
- GPT-3 : 175 milliards
- GPT-4 : 1,7 trilliards



Sources principales :

Principales illustrations : Apprendre le Machine Learning en une semaine – Guillaume Saint-Cirgue, 2024 – www.machinelearnia.com



Prépas scientifiques - Sciences Industrielles pour l'ingénieur, tout en un – Alain Caignot, Vincent Crespel, David Violeau, Vuibert 2022

Articles scientifiques vulgarisés : medium.com