

# Introduction à l'Apprentissage Automatique

Bassem Ben Hamed

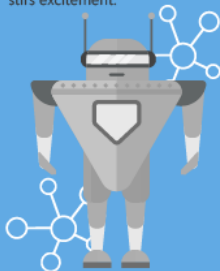
Control, Optimization and Model Reduction in ML - CIMPA 2025

18 février 2025

# IA vs ML vs DL

## ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



1950's 1960's 1970's

## MACHINE LEARNING

Machine learning begins to flourish.



1980's 1990's 2000's 2010's

## DEEP LEARNING

Deep learning breakthroughs drive AI boom.



- **Intelligence Artificielle (IA) :**

- Discipline visant à reproduire des processus cognitifs humains
- Englobe : raisonnement, apprentissage, perception, langage naturel
- Paradigmes : IA symbolique vs IA connexionniste

- **Machine Learning (ML) :**

- Sous-ensemble de l'IA focalisé sur l'apprentissage automatique
- Utilise des algorithmes pour apprendre des patterns à partir des données
- Formellement :  $f : X \rightarrow Y$  où  $f$  est apprise des données

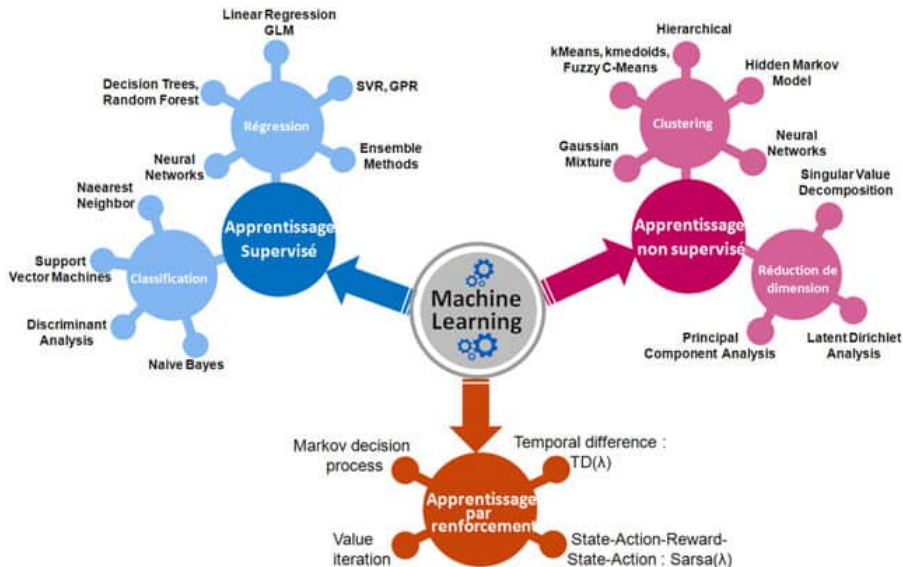
- **Deep Learning (DL) :**

- Sous-ensemble du ML basé sur les réseaux de neurones profonds
- Architecture en couches multiples :  $f(x) = f_n \circ f_{n-1} \circ \dots \circ f_1(x)$

- **Generative AI (GAI) :**

- Modèles capables de générer du nouveau contenu
- Apprentissage de la distribution  $P(X)$  des données

# Types d'Apprentissage



- **Définition formelle :**

- Données :  $\{(x_i, y_i)\}_{i=1}^n$  où  $x_i \in X$ ,  $y_i \in Y$
- Objectif : Trouver  $f : X \rightarrow Y$  minimisant l'erreur empirique

- **Classification :**

- $Y$  discret (catégories)
- Ex: Classification binaire :  $Y = \{0, 1\}$
- Algorithmes : SVM, Random Forest, Réseaux de neurones

- **Régression :**

- $Y$  continu ( $\mathbb{R}$  ou  $\mathbb{R}^n$ )
- Ex: Régression linéaire :  $f(x) = wx + b$
- Algorithmes : Ridge, Lasso, Réseaux de neurones

- **Définition formelle :**

- Données :  $\{x_i\}_{i=1}^n$  où  $x_i \in X$
- Objectif : Découvrir la structure sous-jacente des données

- **Clustering :**

- K-means :  $\min_{\{C_k\}_{k=1}^K} \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$
- DBSCAN : Basé sur la densité des points
- Clustering hiérarchique : Construction d'un dendrogramme

- **Réduction de dimension :**

- PCA :  $X = U\Sigma V^T$  (décomposition SVD)
- t-SNE : Préserve les similarités locales
- Autoencodeurs :  $f(x) \approx x$  avec compression

- **Contexte :**

- Données étiquetées :  $\{(x_i, y_i)\}_{i=1}^l$
- Données non étiquetées :  $\{x_j\}_{j=l+1}^n$
- Typiquement  $l \ll n$

- **Hypothèses fondamentales :**

- Continuité : Points proches  $\rightarrow$  étiquettes similaires
- Clustering : Points d'une même classe forment des clusters
- Manifold : Données sur une variété de dimension inférieure

- **Méthodes principales :**

- Self-training :  $\hat{y} = \arg \max_y P(y|x)$
- Co-training : Utilisation de vues multiples
- Graph-based : Propagation des étiquettes sur un graphe

# Types d'Apprentissage - Apprentissage par Renforcement

- **Composants principaux :**

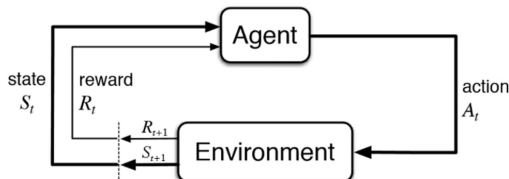
- Agent et Environnement
- États  $S$ , Actions  $A$ , Récompenses  $R$
- Politique  $\pi : S \rightarrow A$

- **Équation de Bellman :**

$$V^\pi(s) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s \right] \quad (1)$$

- **Algorithmes clés :**

- Q-Learning :  $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
- Policy Gradient :  $\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a)]$
- Actor-Critic : Combine valeur et politique





- **Nettoyage des données :**

- Gestion des valeurs manquantes : imputation ou suppression
- Détection et traitement des outliers :  $z\text{-score} = \frac{x-\mu}{\sigma}$
- Correction des erreurs de mesure

- **Feature Engineering :**

- Normalisation :  $x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$
- Standardisation :  $z = \frac{x - \mu}{\sigma}$
- One-hot encoding :  $\{cat_1, cat_2, cat_3\} \rightarrow [1, 0, 0], [0, 1, 0], [0, 0, 1]$

- **Sélection des features :**

- Méthodes filtres : corrélation, information mutuelle
- Méthodes wrapper : recursive feature elimination
- Méthodes embedded : Lasso, Elastic Net

- **Division des données :**

$$\text{Train}(60\%)/\text{Validation}(20\%)/\text{Test}(20\%) \quad (2)$$

- **Validation croisée k-fold :**

- Division en k sous-ensembles
- Score moyen :  $CV_{\text{score}} = \frac{1}{k} \sum_{i=1}^k \text{score}_i$

- **Optimisation des hyperparamètres :**

- Grid Search :  $\{\theta_i\} \times \{\lambda_j\} \times \{\eta_k\}$
- Random Search : échantillonnage aléatoire
- Bayesian Optimization :  $\arg \max_x \text{EI}(x)$

- **1. Définition du problème :**
  - Objectifs métier
  - Métriques de succès
  - Contraintes (temps, ressources, etc.)
- **2. Collecte et préparation :**
  - Sources de données
  - Pipeline ETL
  - Validation de la qualité
- **3. Modélisation :**
  - Sélection d'algorithmes
  - Entraînement et validation
  - Optimisation des performances
- **4. Déploiement :**
  - Infrastructure (cloud, edge)
  - Monitoring
  - Maintenance et mise à jour

