المدرسة الوطنية للذكاء الاصطناعي والرقمنة - بركان
ÉCOLE NATIONALE DE L'INTELLIGENCE ARTIFICIELLE ET DU DIGITAL - BERKANE

MACHINE LEARNING 2

Pr. BOUTAHIR Mohamed Khalifa

2024 - 2025

- OBJECTIFS DU MODULE

1. Comprendre les fondamentaux de l'Apprentissage par Renforcement :
concepts clés et architectures principales.

2. Explorer les algorithmes de base :
méthodes basées sur la valeur (Q-Learning, SARSA), méthodes basées sur la politique (REINFORCE, PPO).

3. Appliquer les techniques sur des cas réels :
implémentation et entraînement d'agents RL sur des environnements simulés (OpenAI Gym, Atari, etc.).

4. Développer des compétences pratiques :
participation à des compétitions Kaggle et projets appliqués pour enrichir le CV et maîtriser l'utilisation de RL dans des applications concrètes.

- VOLUME HORAIRE

	Activités				
	Cours	TD	TP	Activités Pratiques(*)	Evaluation des connaissances et des compétences
VOLUME HORAIRE	18	0	30	0	4
Pourcentage %	35 %	0 %	58 %	0 %	8 %

- CONTENU DU MODULE

- **Chapitre 1 : Introduction à l'Apprentissage par Renforcement**
  - Définition et motivation
  - Différences entre RL, apprentissage supervisé et non supervisé
  - Applications réelles du RL (jeux, robotique, finance, etc.)
  - Défis et limites du RL
- **Chapitre 2 : Fondamentaux de l'Apprentissage par Renforcement**
  - Composants du RL
  - Environnements, états, actions, récompenses
  - Politiques, fonctions de valeur et Q-learning
  - Processus de prise de décision de Markov (MDP)
  - Notions de convergence et d'optimalité

- CONTENU DU MODULE

- **Chapitre 3 : Algorithmes de Base du RL**
  - Méthodes Monte Carlo : Concepts et applications
  - Apprentissage par différence temporelle (TD Learning): Algorithmes TD(0), SARSA, Q-learning
- **Chapitre 4 : Stratégies d'Exploration et Exploitation**
  - Balance exploration-exploitation : Stratégies epsilon-greedy
  - Méthodes de recherche de politique : Approximation de fonctions et gradients de politique; Algorithmes de recherche de politique basés sur des gradients

- CONTENU DU MODULE

- **Chapitre 5 : Apprentissage Profond par Renforcement (DRL)**
  - Introduction au Deep RL : Concepts de base du deep learning appliqué au RL
  - Réseaux de neurones pour l'approximation des fonctions de valeur
  - Deep Q-Networks (DQN)
  - Techniques d'optimisation : replay buffer, target network
  - Algorithmes avancés de DRL
  - Double DQN, Dueling DQN, Prioritized Experience Replay
  - Introduction aux Policy Gradient Methods : REINFORCE, Actor-Critic
  - Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO)

- CONTENU DU MODULE

- **Chapitre 6 : Outils et Bibliothèques pour le RL et DRL**
  - OpenAI Gym :Utilisation et création d'environnements simulés; Exemples d'implémentation de problèmes RL classiques
  - TensorFlow et PyTorch: Implémentation des réseaux de neurones pour le DRL; Entraînement et évaluation des agents RL
  - Stable Baselines et Rllib : Utilisation des bibliothèques avancées pour le RL ; Implémentation et optimisation des algorithmes

CHAPITRE 1

Introduction à l'Apprentissage par Renforcement

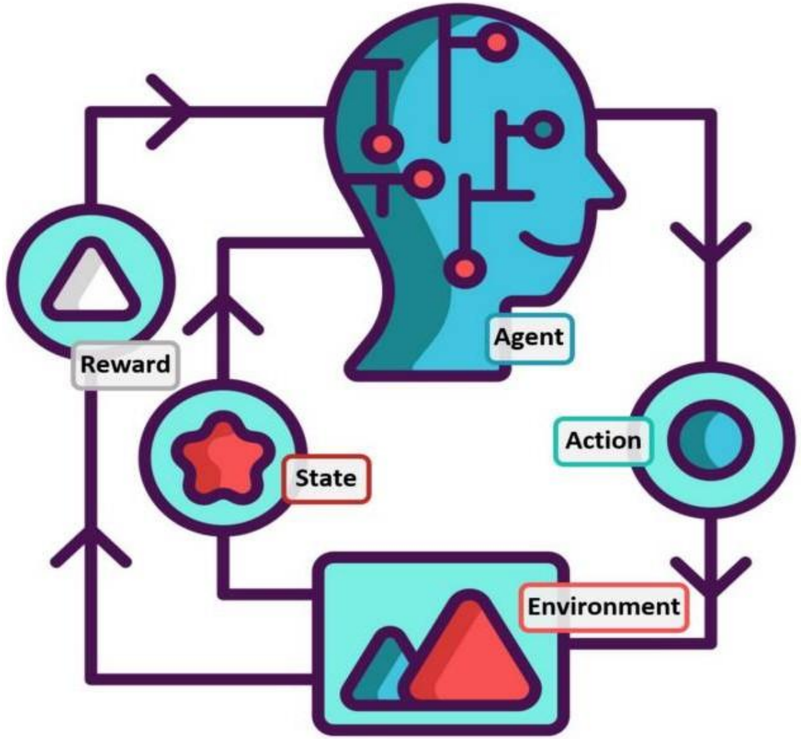
- Définition et motivation
- Différences entre RL, apprentissage supervisé et non supervisé
- Applications réelles du RL (jeux, robotique, finance, etc.)
- Défis et limites du RL

C’est quoi déjà l'Apprentissage par Renforcement?

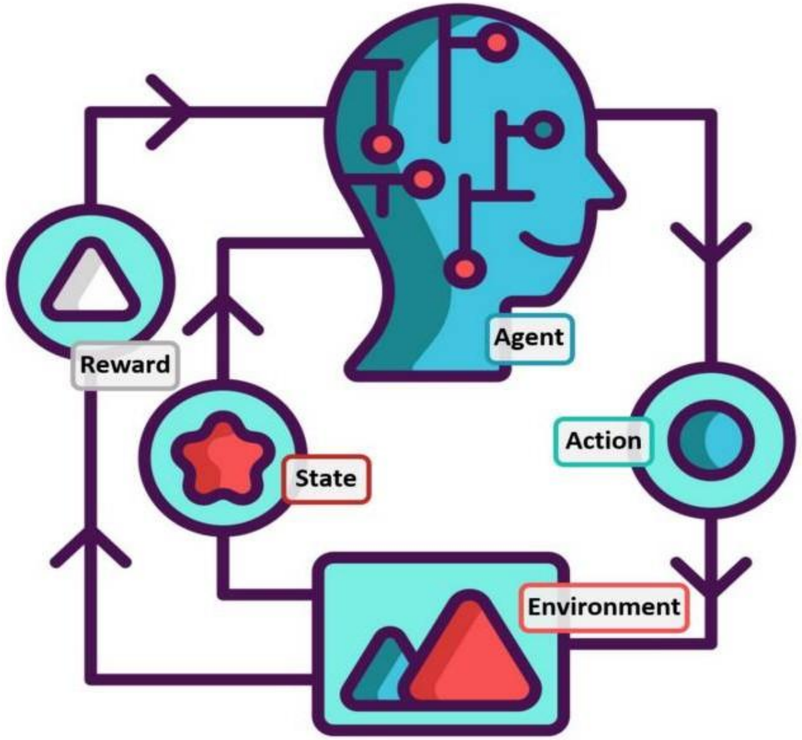
1. Définition de l'Apprentissage par Renforcement

L'Apprentissage par Renforcement

L’apprentissage par renforcement est une technique où un **agent** apprend à prendre des **décisions** en interagissant avec un **environnement**. Il reçoit une récompense pour les bonnes actions et une punition pour les mauvaises. Petit à petit, il découvre les meilleures actions à prendre pour obtenir le plus de **récompenses**.



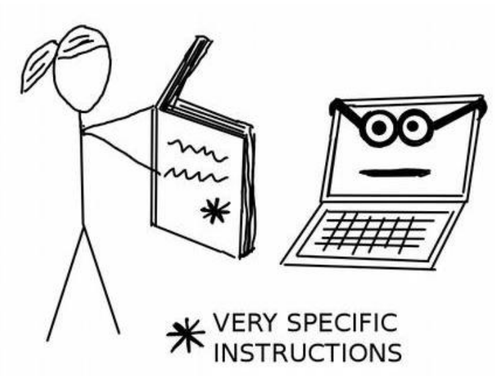
1. Définition de l'Apprentissage par Renforcement



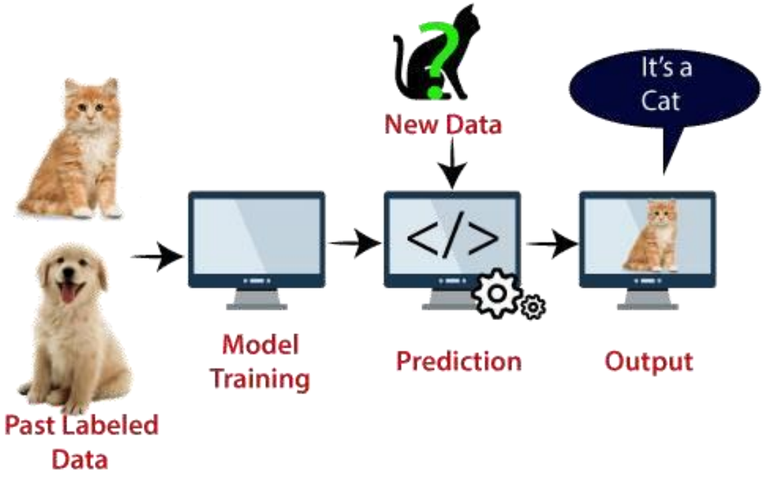
Dans ce processus, **l'état** représente la situation actuelle de **l'agent** dans **l'environnement**. À chaque instant, **l'agent** observe **l'état**, choisit une **action** en fonction de sa stratégie, puis reçoit une nouvelle **récompense** et un nouvel **état** en retour.
L'objectif est d'apprendre une **politique** qui permet de choisir la meilleure **action** en fonction de **l'état** actuel pour maximiser les **récompenses** à long terme.

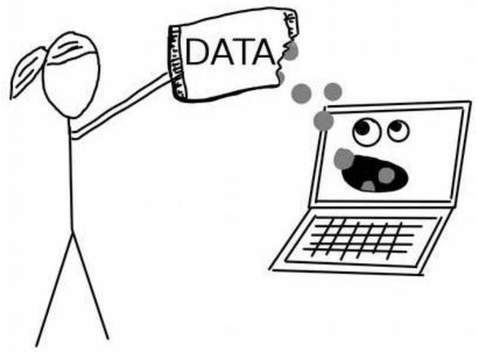
1. Définition de l'Apprentissage par Renforcement

- **Informatique classique :** les ordinateurs sont programmés pour chaque tâche qu'ils doivent accomplir



- **Concepts d'apprentissage automatique :** des échantillons sont fournis aux machines et celles ci découvrent / apprennent les tâches à accomplir sur la base de ces exemples





1. Définition de l'Apprentissage par Renforcement

- **Nature de l'apprentissage :** nous apprenons en interagissant avec notre environnement.
- Les **bébés** n'ont pas d'enseignant explicite mais une connexion neurosensorielle directe avec l'environnement.

1. Définition de l'Apprentissage par Renforcement

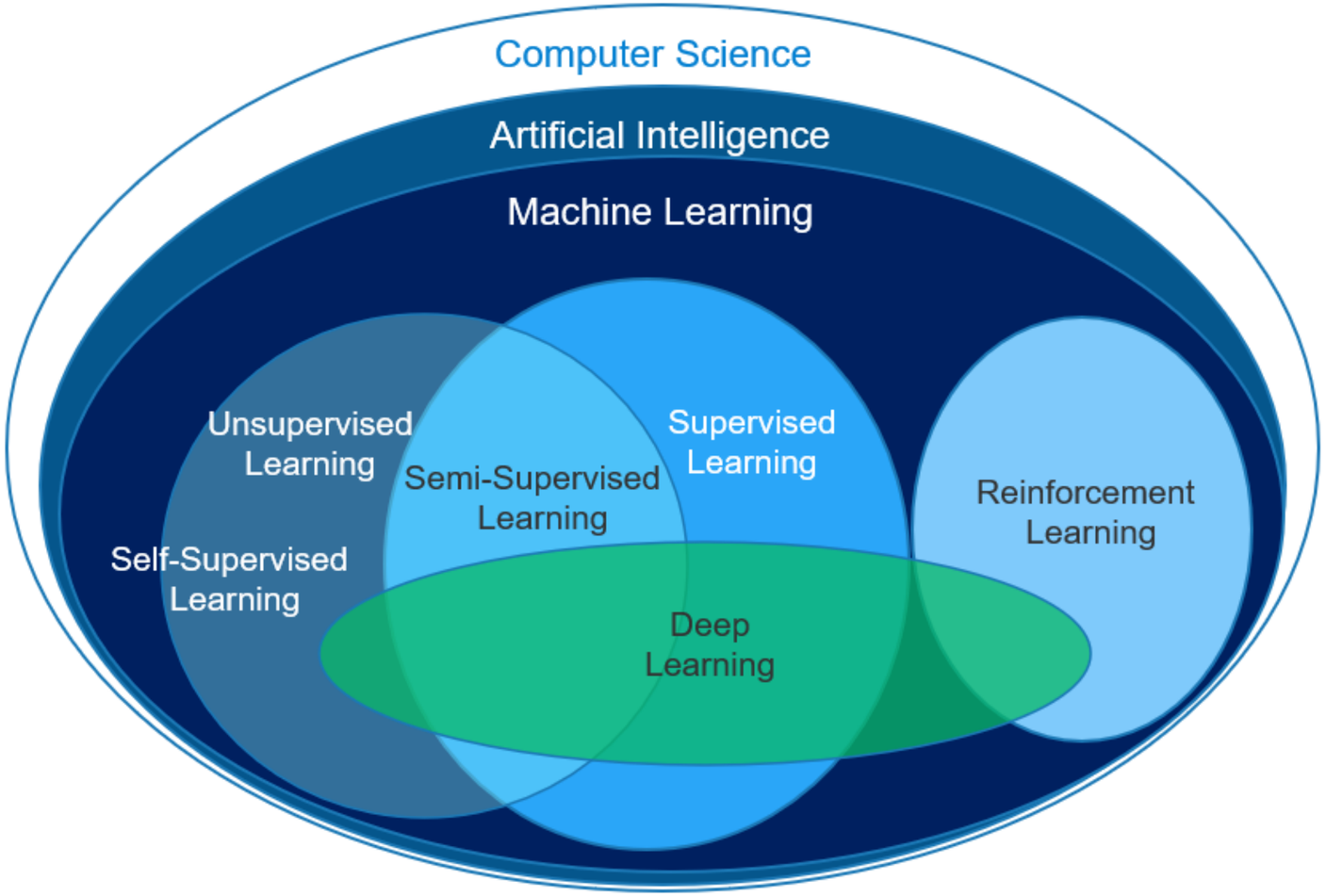
Apprendre à conduire une voiture

Apprendre à tenir une conversation

- Nous sommes conscients de la manière dont notre **environnement** réagit à ce que nous faisons et nous cherchons à influencer ce qui se passe par notre comportement.

- Apprendre par l'interaction : idée fondamentale qui sous-tend presque toutes les théories de l'apprentissage et de l'intelligence.

2. Différences entre RL, apprentissage supervisé et non supervisé



2. Différences entre RL, apprentissage supervisé et non supervisé

Apprentissage Supervisé

L'apprentissage supervisé consiste à entraîner un modèle à partir de données étiquetées, c'est-à-dire des données où l'entrée et la sortie attendue sont connues. Le modèle apprend à prédire la sortie pour de nouvelles entrées.

Apprentissage Non Supervisé

L'apprentissage non supervisé traite des données non étiquetées. Le modèle découvre des structures, des modèles ou des regroupements cachés dans les données sans l'aide de labels pré-définis.

2. Différences entre RL, apprentissage supervisé et non supervisé

Supervised Learning

Imaginez que vous enseignez à un enfant à identifier les fruits. Vous lui montrez des images de pommes, d'oranges et de bananes, en lui disant le nom de chaque fruit. Le modèle apprend à prédire le type de fruit en fonction de son apparence.

Unsupervised Learning

Imaginez que vous donnez à l'enfant une boîte de fruits mélangés sans aucune indication. L'enfant doit trier les fruits en fonction de leurs similitudes, comme la forme, la couleur ou la taille. Le modèle découvre des groupes de fruits similaires sans l'aide de labels.

Reinforcement Learning

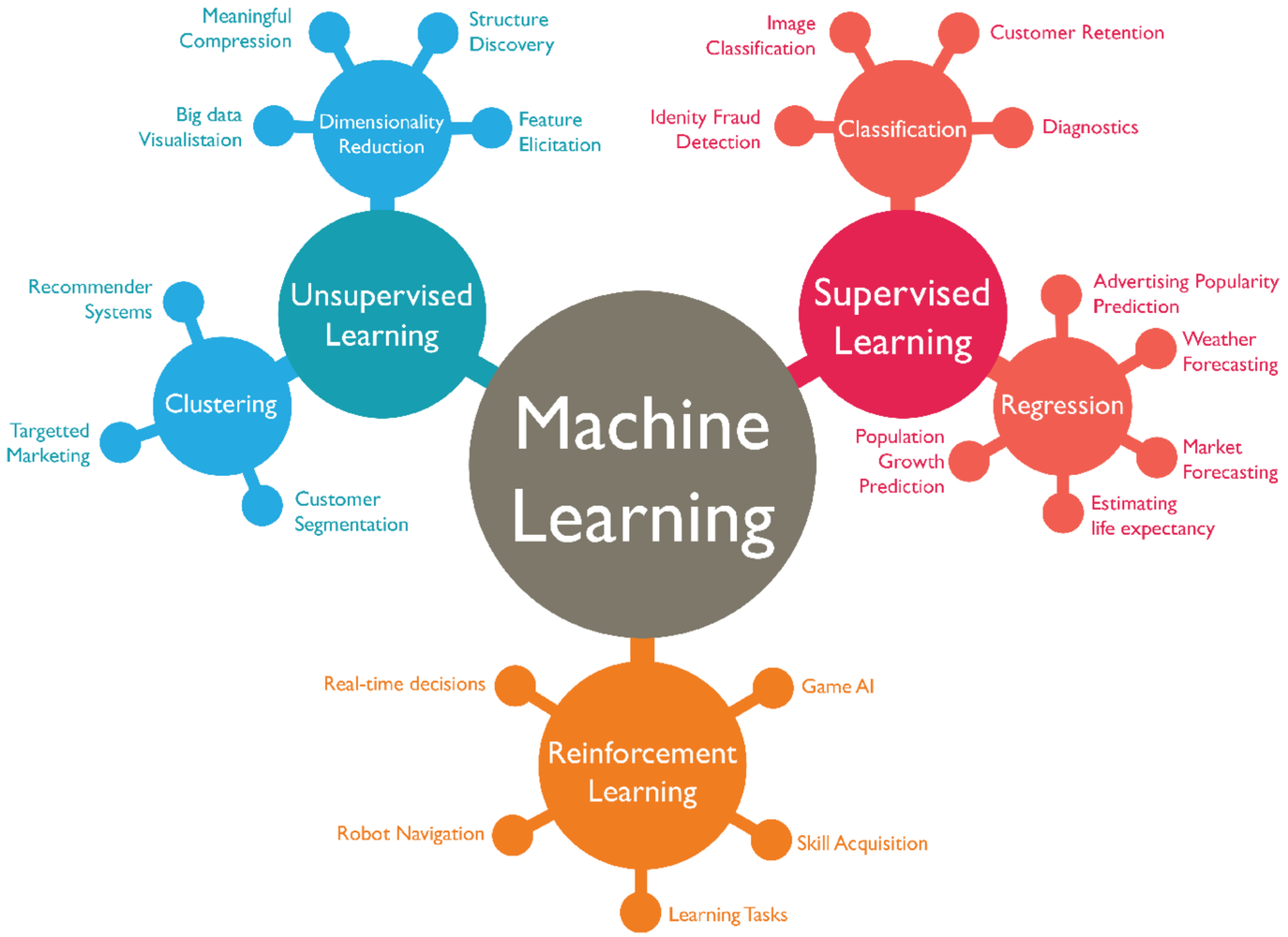
Imaginez que vous donnez à l'enfant un jeu vidéo et qu'il doit apprendre à y jouer. L'enfant explore le jeu, effectue des actions et reçoit des récompenses (comme des points) ou des pénalités (comme la perte de vies). L'enfant apprend à maximiser ses récompenses en ajustant sa stratégie au fil du temps.

Task Driven
(Classification/Regression)

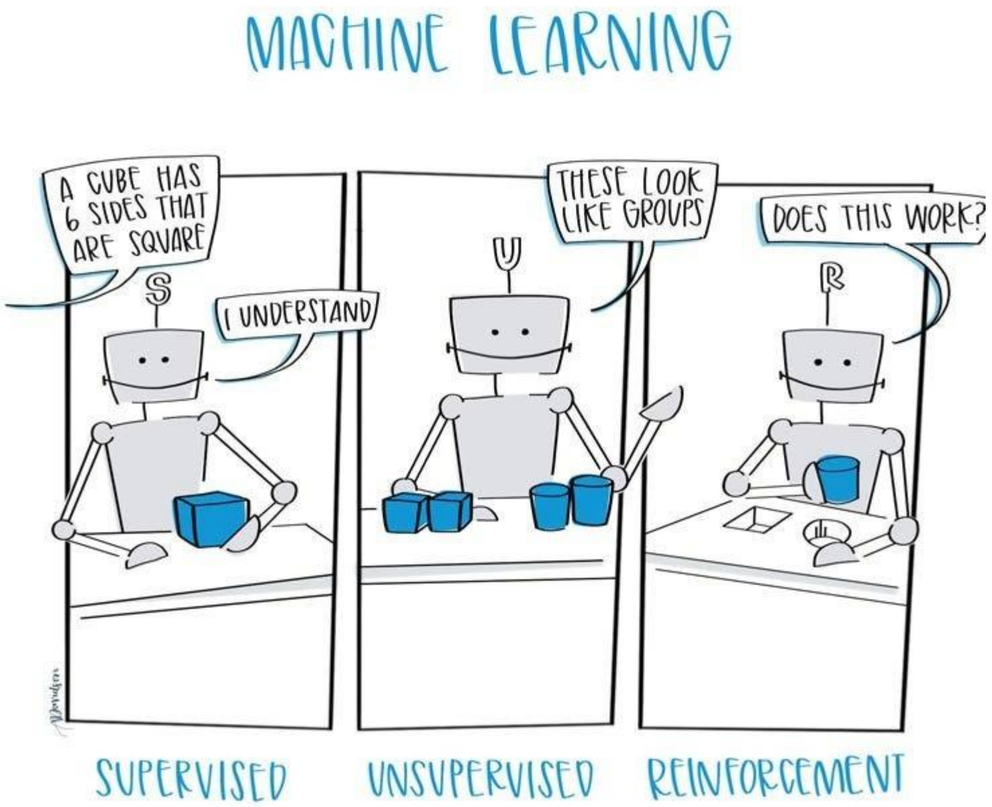
Data Driven
(Clustering)

Learning from
mistakes
(Playing Games)

2. Différences entre RL, apprentissage supervisé et non supervisé



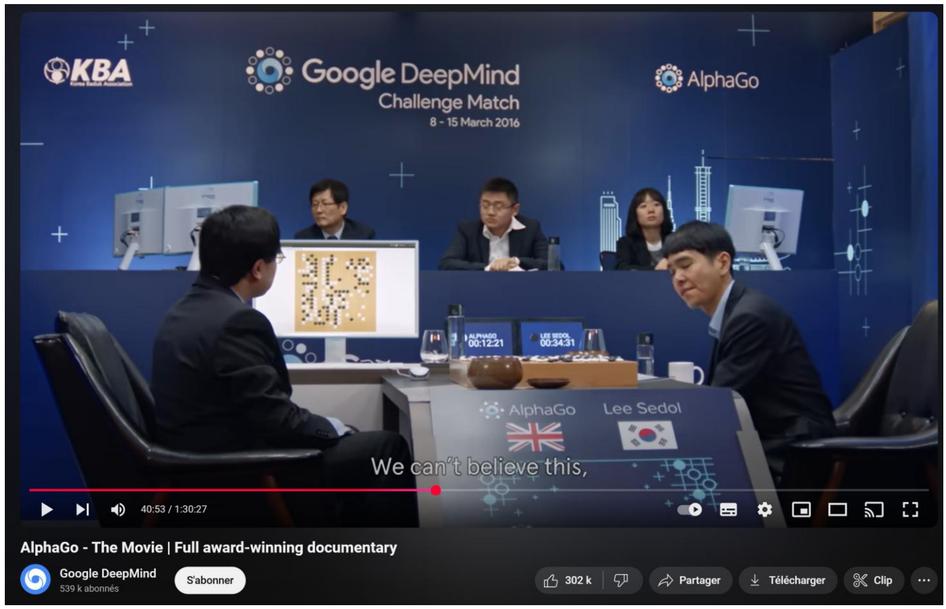
2. Différences entre RL, apprentissage supervisé et non supervisé



3. Applications réelles du RL

L'histoire d'AlphaGo :

En 2016, le programme informatique AlphaGo a capté l'attention du monde entier lorsqu'il a vaincu le légendaire joueur de Go Lee Sedol. Le jeu de Go est l'un des jeux les plus complexes jamais conçus, avec plus de configurations possibles que d'atomes dans l'univers. Il s'agissait depuis longtemps d'un grand défi pour l'intelligence artificielle et la victoire 4-1 d'AlphaGo a été considérée par beaucoup comme une décennie en avance sur son temps. Le système a été inventé par DeepMind, cofondé par le scientifique Demis Hassabis. Cinq mois plus tôt, AlphaGo avait battu le champion d'Europe Fan Hui, devenant ainsi le premier programme à vaincre un joueur professionnel.



3. Applications réelles du RL

AlphaZero: Le leader des échecs:

AlphaZero, développé par DeepMind, est un programme d'IA basé uniquement sur l'apprentissage par renforcement. Il apprend en jouant contre lui-même, sans connaissances humaines préprogrammées, et optimise ses stratégies via des réseaux neuronaux profonds. En seulement 9 heures, il a joué 44 millions de parties et surpassé Stockfish après 4 heures d'entraînement. Contrairement aux moteurs traditionnels comme Stockfish, qui évaluent jusqu'à 70M de positions/s par recherche exhaustive, AlphaZero utilise une approche plus efficace avec Monte Carlo Tree Search (MCTS) et un réseau neuronal pour une évaluation plus précise des positions.

3. Applications réelles du RL

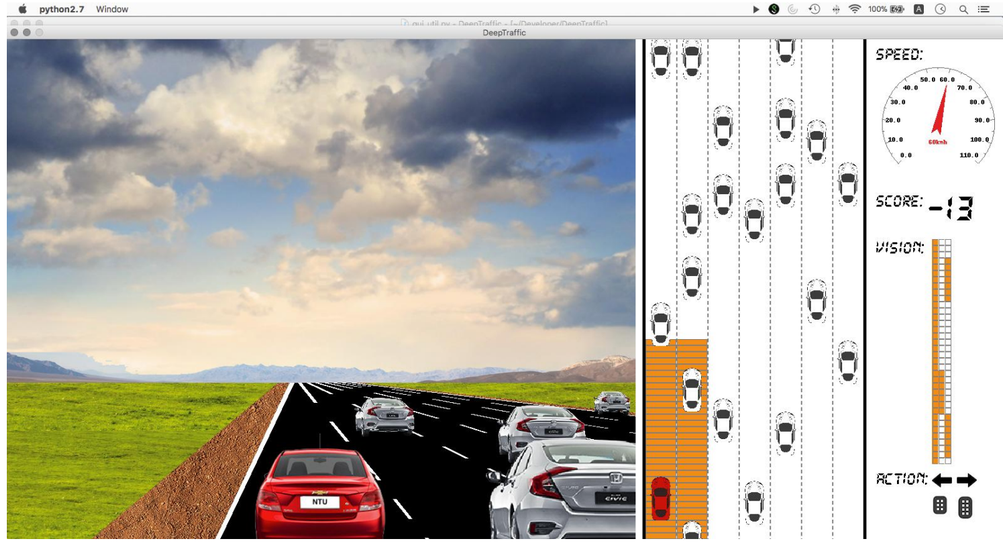
Open AI sur l'apprentissage par Renforcement:

En avril 2019, OpenAI Five, une intelligence artificielle développée par OpenAI, a marqué l'histoire en devenant la première IA à battre les champions du monde du jeu vidéo Dota 2. Elle a remporté deux victoires consécutives contre l'équipe OG, alors championne en titre. OpenAI Five a été entraînée en jouant des millions de parties contre elle-même, utilisant des techniques d'apprentissage par renforcement profond pour maîtriser les stratégies complexes du jeu. Cette réalisation démontre le potentiel des IA à atteindre des performances surhumaines dans des environnements complexes et dynamiques.

3. Applications réelles du RL

Apprentissage par renforcement pour les voitures autonomes:

Ce projet utilise l'apprentissage par renforcement pour entraîner une voiture autonome à maximiser sa vitesse. Un réseau de neurones convolutionnel extrait les caractéristiques d'une matrice représentant l'environnement de conduite.



- 5 actions évaluées pour estimer les récompenses futures
- Algorithme Q-learning pour l'entraînement
- Simulation sur une autoroute à 7 voies avec conditions de trafic
- Après 2340 min d'entraînement : la voiture atteint 94 km/h en moyenne (vs 110 km/h max)

✓ Le modèle apprend des politiques de contrôle optimisées selon le trafic !

3. Applications réelles du RL

Apprentissage par renforcement pour les voitures autonomes:

Tesla applique l'apprentissage par renforcement (Reinforcement Learning, RL) pour améliorer les capacités de conduite autonome de son système Autopilot.

L'IA apprend en interagissant avec son environnement, en recevant des récompenses basées sur la sécurité et l'efficacité des décisions prises sur la route.

Grâce à des algorithmes de Deep Reinforcement Learning, Tesla entraîne ses modèles dans des simulations et exploite les données réelles collectées par sa flotte de véhicules. Cette approche permet d'optimiser la prise de décision en temps réel, notamment pour la gestion des intersections, le dépassement ou l'évitement d'obstacles, en améliorant continuellement le comportement du véhicule sans intervention humaine directe.

3. Applications réelles du RL

L'Apprentissage par Renforcement dans le Trading

L'Apprentissage par Renforcement (RL) est utilisé en trading algorithmique pour entraîner des agents capables de prendre des décisions d'achat et de vente de manière autonome. L'agent interagit avec le marché financier comme un environnement, où chaque état représente des indicateurs économiques (prix, tendances, volumes d'échange, etc.), et chaque action correspond à acheter, vendre ou conserver un actif. L'agent reçoit une récompense lorsqu'il réalise un profit et une pénalité lorsqu'il subit une perte.

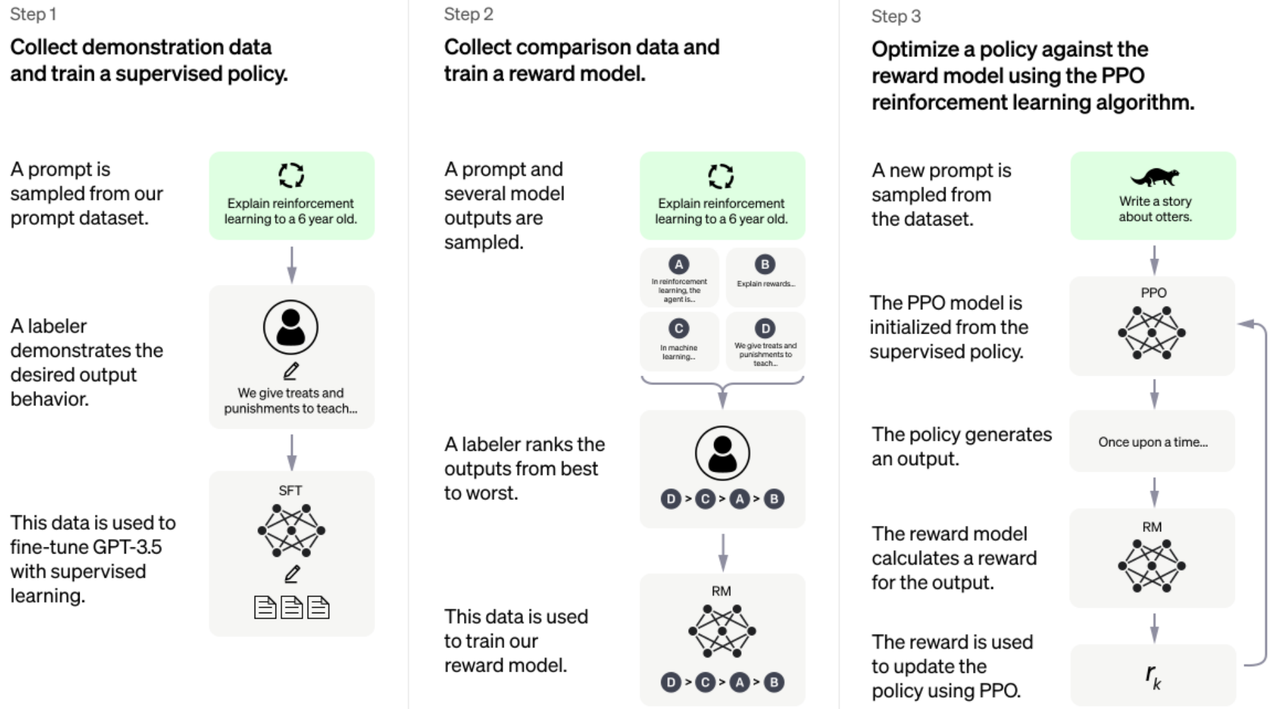
Pour implémenter un modèle de RL en trading, on utilise des réseaux de neurones profonds combinés à des algorithmes comme Deep Q-Networks (DQN) ou Proximal Policy Optimization (PPO).

3. Applications réelles du RL

Le processus d'apprentissage du ChatGPT

ChatGPT utilise l'apprentissage par renforcement pour améliorer ses réponses grâce à une méthode appelée Proximal Policy Optimization (PPO).

D'abord, il est entraîné sur des conversations humaines pour apprendre à répondre correctement. Ensuite, un modèle de récompense est créé en demandant à des humains de classer plusieurs réponses possibles. Enfin, avec l'algorithme PPO (Proximal policy optimization), ChatGPT ajuste ses réponses en fonction des évaluations, en apprenant progressivement à générer des réponses plus pertinentes et naturelles. Ce processus permet d'optimiser la qualité des interactions avec les utilisateurs.



4. Défis et limites du RL

Nécessité d'un grand volume de données

- Un agent RL doit explorer un grand nombre de situations pour apprendre efficacement.
- Cela prend **beaucoup de temps** et **nécessite des ressources puissantes** (GPU/TPU).

Équilibre entre exploration et exploitation

- Un agent doit **tester de nouvelles stratégies (exploration)** tout en **utilisant ce qu'il a appris (exploitation)**.
- Trouver cet équilibre est complexe et peut mener à des stratégies sous-optimales.

Environnements instables ou changeants

- Si l'environnement change trop souvent, l'agent peut **désapprendre** ce qu'il a appris.
- Exemple : un robot RL qui apprend à marcher sur une surface lisse peut échouer sur un terrain accidenté.

Devoir : Implémentation d'un Agent d'Apprentissage par Renforcement

Objectif : Dans cet exercice, vous allez programmer un agent intelligent pour trouver un trésor 🏆 dans une grille tout en évitant les pièges 💀.

Vous devez écrire le code (en python) de l'agent et lui permettre d'apprendre à trouver le chemin optimal jusqu'au trésor.

Règles du Jeu

• L'agent commence en haut à gauche de la grille (case (0,0)).

• Il peut se déplacer : HAUT, BAS, GAUCHE, DROITE.

• Chaque déplacement a un coût de -1 (pour encourager le chemin le plus court).

• S'il atteint un piège (💀), il perd immédiatement (-10 points).

• S'il atteint le trésor (🏆), il gagne la partie (+10 points).

• L'agent doit apprendre par lui-même en jouant plusieurs parties.