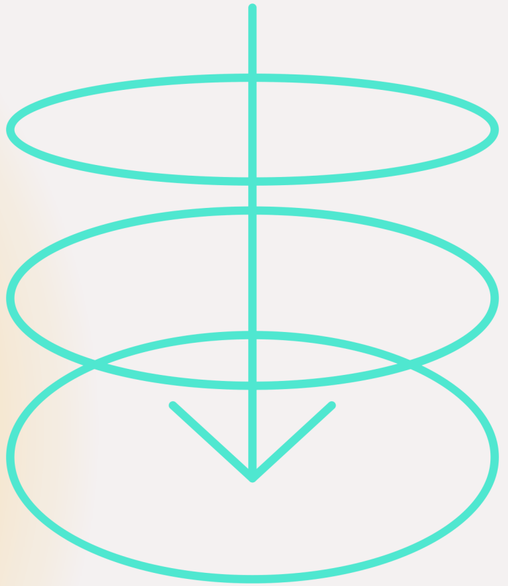
Introduction to Reinforcement Learning

Google Developer Student Clubs



Presented By Sim Sze Yu & Lim Zhe Yu

01 - What is it?
02 - Core Components
03 - How it works?
04 - Terminology

Intro To RL

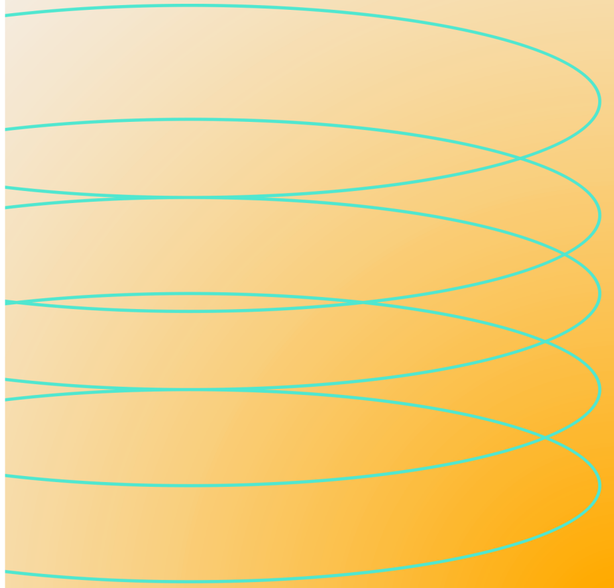
05 - RL Algorithms
06 - Applications
07 - Challenges
08 - Future Trends

What is it?

01 - What is it?

Definition

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent takes actions, and the environment provides feedback in the form of rewards or penalties.



01 - What is it?

Key Characteristics of RL

Interaction

The agent interacts with an environment, performing actions and observing the consequences.

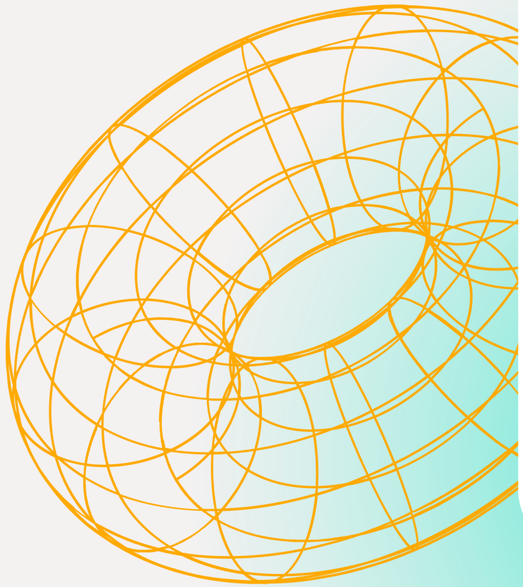
Goal-oriented

The environment provides feedback to the agent in the form of rewards or punishments based on the actions taken.

Feedback

RL is goal-oriented; the agent's objective is to learn a strategy or policy that maximizes the cumulative reward over time.

How RL Differs from Other Machine Learning Paradigms



01 - What is it?

Unlike supervised learning, where the model is trained on labeled examples, and unsupervised learning, where the model identifies patterns without explicit labels, RL learns from interacting with an environment.

RL is more about decision-making and learning optimal strategies over time rather than mapping inputs to outputs as in traditional machine learning.

Core Components

02 - Core Components

Key Concepts

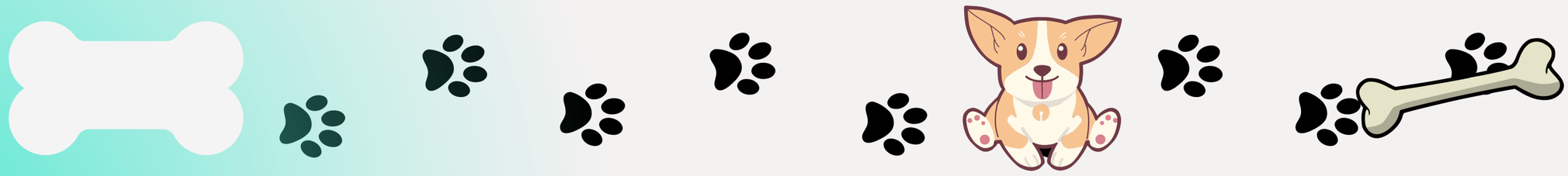
Agent

Environment

Actions

Rewards

State



Workflow

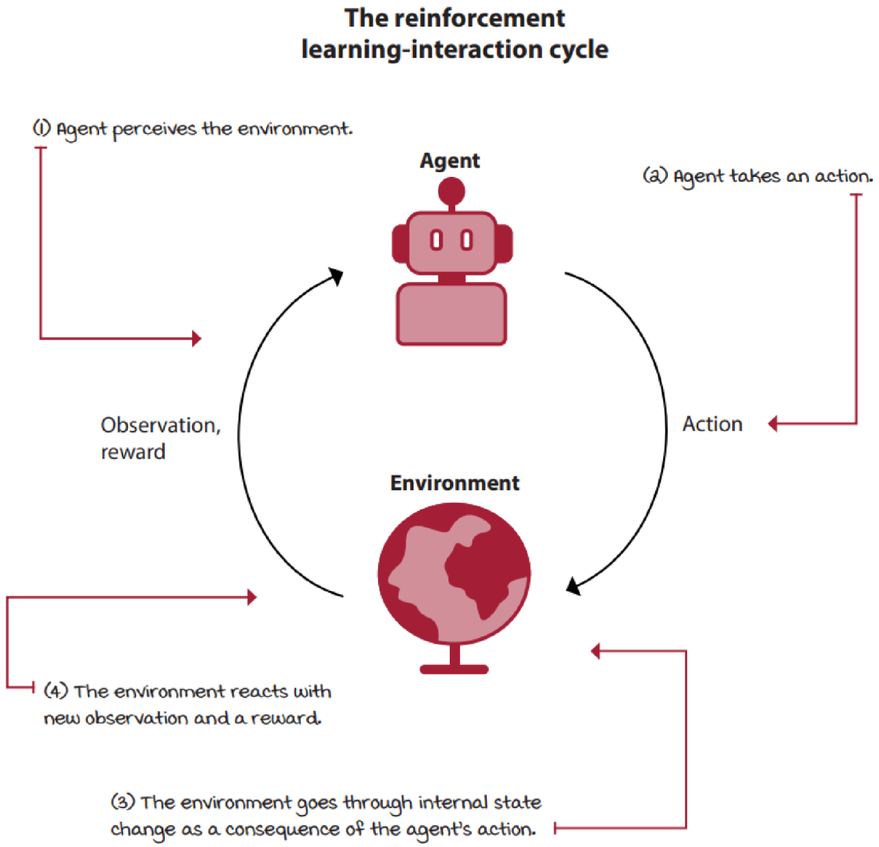
Iteration and Improvement

03 - Workflow

RL is an iterative process where the agent continuously interacts with the environment, refines its understanding, and improves its decision-making strategy.

03 - Workflow

The RL Loop

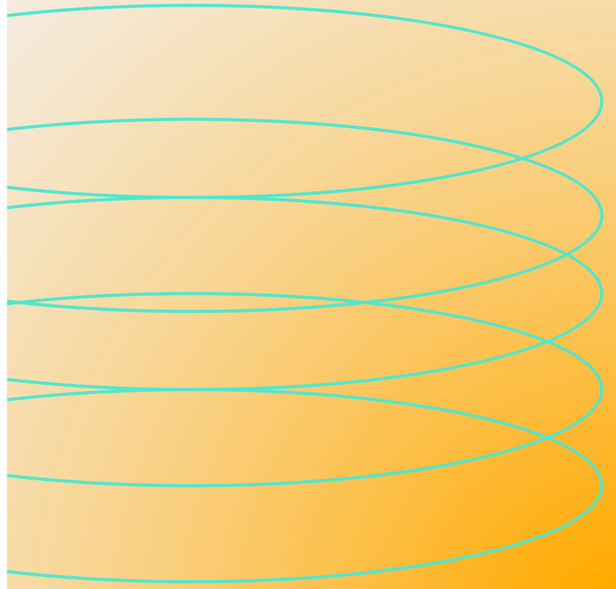


source: Grokking Deep Reinforcement Learning Book

03 - Workflow

Markov Decision Process (MDP) is a mathematical framework used to formalize the RL problem. It defines the interaction between an agent and an environment in terms of states, actions, transition probabilities, and rewards.

Markov Decision Process



Components of an MDP

State Space (S)

The state space is the set of all possible situations or configurations that the environment can be in.

Action Space (A)

The action space is the set of all possible actions that the agent can take.

03 - Workflow

Components of an MDP

Transition Probabilities (P)

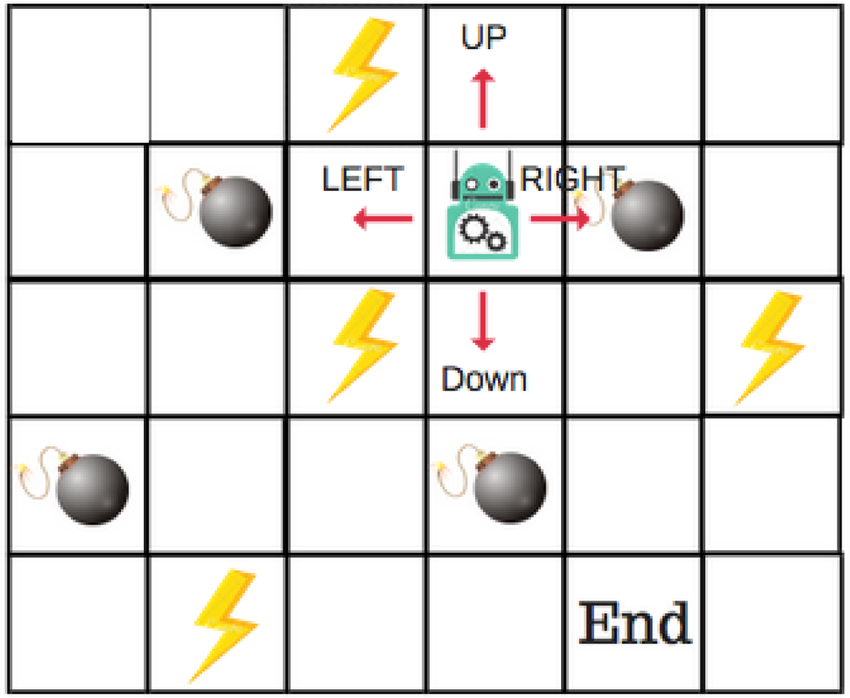
The transition probabilities describe the likelihood of transitioning from one state to another given a particular action. It is represented as P(s' | s,a)

Rewards (R)

Rewards represent the immediate numerical feedback the agent receives after taking a specific action in a particular state. It is denoted as R(s,a,s').

03 - Workflow

MDP assumes the Markov property, which states that the future state depends only on the current state and action, not on the sequence of states and actions that preceded it.



source: freecodecamp

Terminology

04 - Terminology

Policy

Return

Q-function

Discount Factor

Exploration - ExploitationTradeoff

- A policy (π) is a strategy or a mapping from states to actions. It defines the agent's way of behaving in an environment.
- Deterministic policy: $\pi: S \rightarrow A$, mapping states directly to actions.
- Stochastic policy: $\pi: S \times A \rightarrow [0,1]$, specifying probabilities for each action in each state.

Policy

Q-function

- The Q-function Q(s,a)) represents the expected cumulative reward of taking action a in state s and following a certain policy thereafter.
- It is fundamental in value-based RL algorithms.

Return

- The return represents the expected cumulative reward an agent can obtain from a given state (or state-action pair) following a particular policy.
- Denoted as V(s) for states and Q(s,a) for state-action pairs.

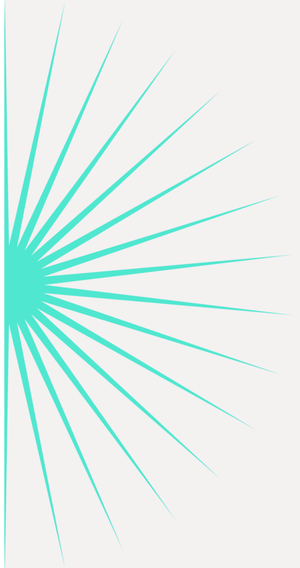
Discount Factor (γ)

- The discount factor (γ) is a value between 0 and 1 that determines the importance of future rewards. It discounts the value of future rewards relative to immediate rewards.
- Mathematically, Q(s,a)=R(s,a)+γ · maxaQ(s',a').
- The lower the discount factor, the more impatient the model gets

return | $R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$

Exploration vs. Exploitation

- Exploration: Trying new actions to discover their effects on the environment.
- Exploitation: Choosing actions that are known to yield high rewards based on past experience.



Exploration-Exploitation Tradeoff

- The exploration-exploitation tradeoff is the dilemma an agent faces in choosing between known high-reward actions (exploitation) and exploring new actions to discover potentially higher rewards (exploration).
- Balancing these aspects is crucial for effective learning.

Overview

	Mars rover 	Helicopter 	Chess 
states	6 states	position of helicopter	pieces on board
actions	← →	how to move control stick	possible move
rewards	100, 0, 40	+1, −1000	+1, 0, −1
discount factor γ	0.5	0.99	0.995
return	$R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$	$R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$	$R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$
policy π	100 ← ← ← → 40	Find $\pi(s) = a$	Find $\pi(s) = a$

source: coursera

RL Algorithms

05 - RL Algorithms

Q-Learning

Q-Learning is a simpler algorithm suitable for problems with discrete state and action spaces.

Deep Q-Learning (DQN)

DQN extends Q-Learning by incorporating deep neural networks, making it suitable for problems with high-dimensional state spaces

Applications

06 - Applications

Game playing

Robotics

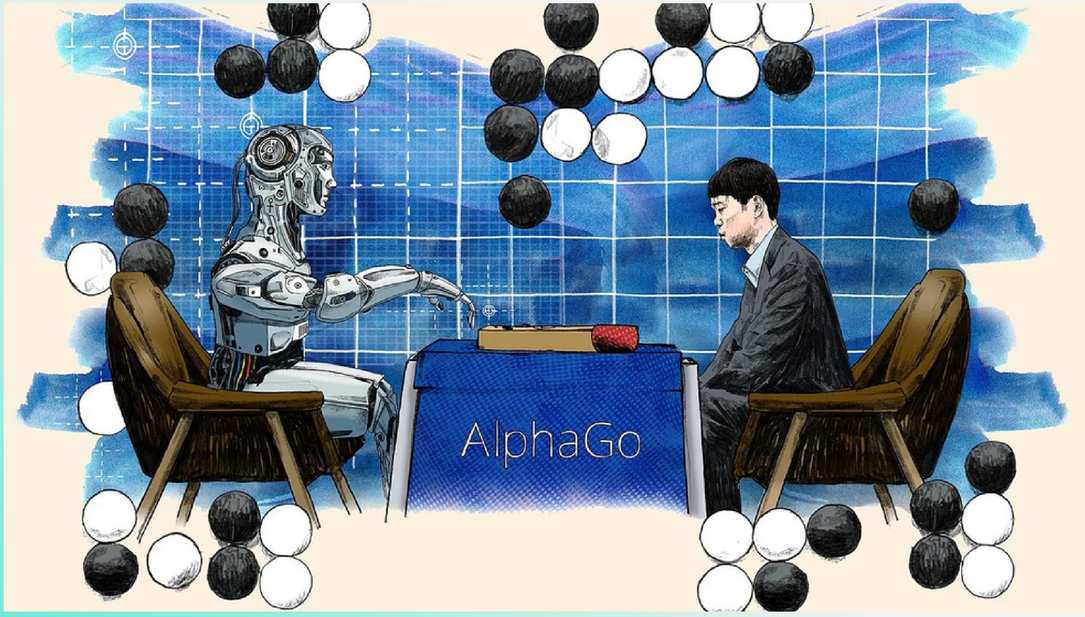
Finance

Healthcare

Autonomous Vehicles

06 - Applications

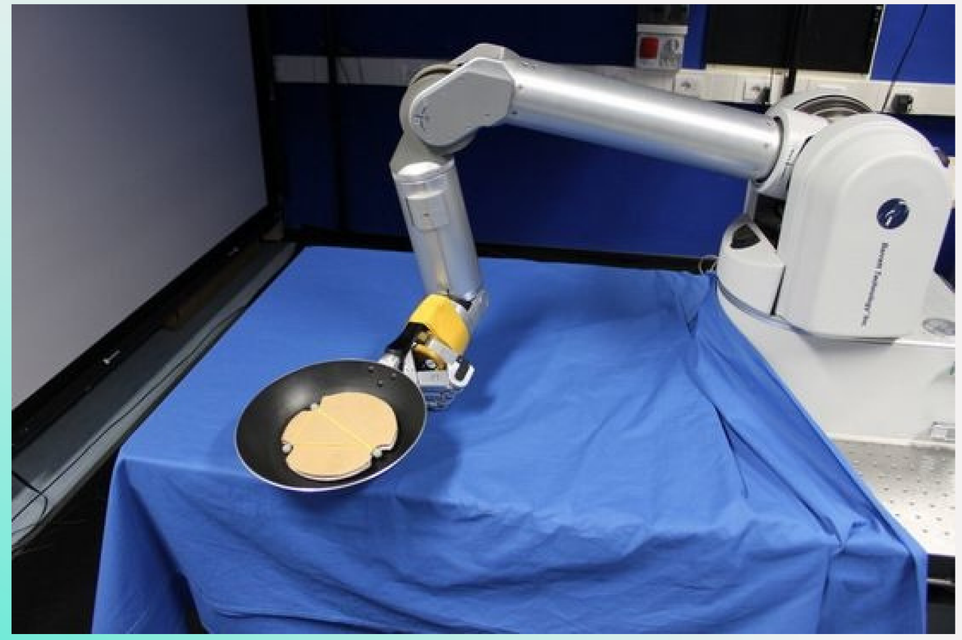
Game playing



- **Defeated world champion Go player - Lee Sedol**
- **Supervised and Reinforcement learning**
- **Neural Network Training**
- **Fine-tuning through Self-Play**
- **Monte Carlo Tree Search (MCTS)**

06 - Applications

Robotics



- **Trial and Error Learning**
- **Continuous Action Space**
- **Deep Deterministic Policy Gradients (DDPG)**
- **Adaptation through Rewards and Penalties**
- **Efficient Policy Optimization**

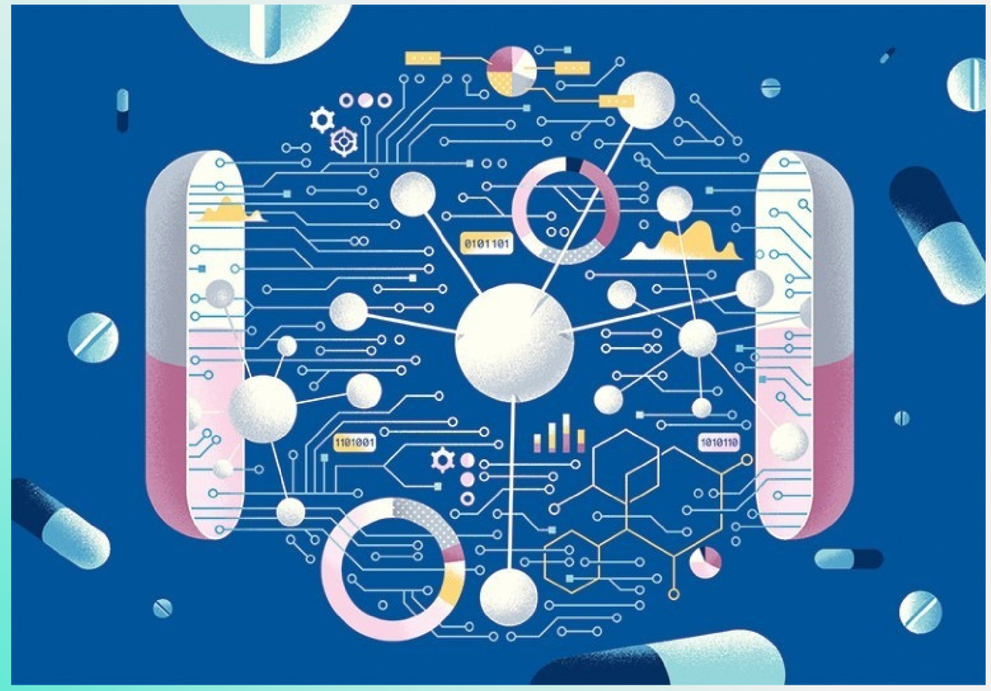
06 - Applications

Finance

- **Market Decision Engine**
- **Rewards and Penalties System**
- **Q-learning and DQN Techniques**
- **High-Frequency Trading (HFT)**
- **Market Liquidity and Efficiency**

06 - Applications

Healthcare



- **Reinforcement Learning for Exploration**
- **Rewards for Desired Properties**
- **Iterative Proposal and Evaluation**
- **Algorithmic Acceleration**
- **Penalties for Undesirable Properties**

06 - Applications

Autonomous Vehicles

- **Learning from Simulation and Real-World Data**
- **Rewards for Safe and Efficient Driving**
- **Policy Optimization with Proximal Policy Optimization (PPO) and DDPG**
- **Learning Complex Decision-Making Parameters (acceleration, braking, and lane-changing policies)**
- **Simulation Testing for Safety Validation**
- **Integration of Sensor Fusion**

Challenges

07 - Challenges

Efficiency

- RL requires significant computational resources.
- Learning efficiency can be hindered by a need for extensive data.

Reward Shaping

- Striking the right exploration-exploitation balance is challenging
- Limited positive feedback complicates effective exploration

Ehtical Considerations

- Crafting precise reward functions is a design challenge
- Issues arise when rewards are delayed or tied to a sequence of actions

Exploration

- RL models may perpetuate biases in training data
- Model opacity poses challenges in ensuring transparent decisions
- Concerns emerge when RL agents produce unintended and potentially harmful outcomes

Future Trend

08 - Future Trends

- Advancements in Deep Reinforcement Learning (DRL)
- Transfer Learning and Generalization
- Efficient Algorithms for Real-Time Applications

Thanks