Reinforcement Learning introduction

What is Reinforcement Learning?

- **Reinforcement learning** is a type of machine learning where an agent learns to make decisions by interacting with an environment.
- Instead of being taught with labeled examples, RL uses a **reward system** that encourages desirable behaviors and discourages undesirable ones.
- The learning process involves moving through a sequence of **states** where the agent must choose **actions** to maximize cumulative reward.

Key Concepts in Reinforcement Learning

- **State (s):**

  Represents the current situation or environment condition the agent is in. For example, the position and speed of an **autonomous helicopter**.

- **Action (a):**

  The decision or move the agent chooses given the current state, such as adjusting helicopter controls.

- **Reward (R):**

  A scalar value received after taking an action in a state that indicates how good or bad the action was. Positive rewards encourage repeating actions, negative rewards discourage them.

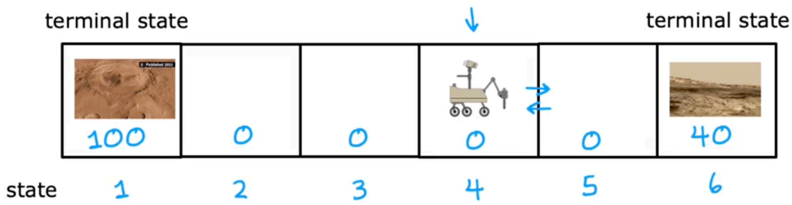
- **Terminal State:**

  A special state where the episode ends, often representing a goal achieved or failure (e.g., helicopter crashes). No further rewards are given after this.

Application:

- Controlling robots
- Factory optimization
- Financial (stock) trading
- Playing games (including video games)

Mars rover example



- The rover can choose to move left or right, impacting its rewards and states.
- Moving left from state 4 leads to state 1, where it receives a reward of 100, while moving right leads to state 6 with a reward of 40.

$$(s, a, R(s), s') = (4, \leftarrow, 0, 3)$$

The Return in reinforcement learning

- The return is defined as the sum of rewards received, adjusted by a discount factor that prioritizes immediate rewards over future ones.
- An analogy is provided comparing the choice between picking up a five-dollar bill immediately or walking to get a ten-dollar bill, illustrating the trade-off between reward value and time.

$$\text{Return} = R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots \text{(until terminal state)} = \sum_{i=0}^{t} \gamma^i R(s_i)$$

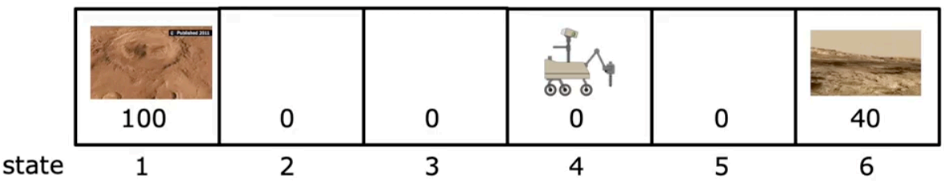
Discount Factor

- The discount factor (γ) is a value less than 1 that reduces the weight of **future rewards** (because some rewards may come later than others), making the algorithm favor **quicker reward**s
- Common values for the discount factor are around 0.9 or 0.99, but for illustrative purposes, a lower value like 0.5 is used to show how it heavily discounts future rewards.

→ A high discount factor means future rewards are valued almost as much as immediate ones.

→ A lower discount factor makes the agent "impatient," valuing immediate rewards much more.

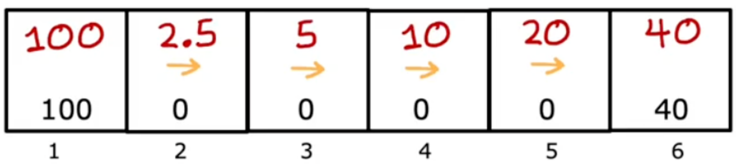
Example:



- With $\gamma = 0.9$: $\longrightarrow \text{Return} = 0 + (0.9)0 + (0.9)^2 0 + (0.9)^3 100 = 0.729 \times 100 = 72.9$
- With $\gamma = 0.9$: $\longrightarrow \text{Return} = 0 + (0.5)0 + (0.5)^2 0 + (0.5)^3 100 = 12.5$

Examples of Returns Based on Actions

- Different starting states yield different returns based on the actions taken. For instance, starting from state 4 and moving left results in a return of 12.5, while moving right yields a return of 10.



- The return varies significantly depending on the chosen actions, demonstrating the importance of strategy in reinforcement learning.

Making decisions: Policies in reinforcement learning

- A policy, denoted as π, is a function that maps any given state s to an action a that the algorithm should take.
- Different strategies can be employed to choose actions, such as opting for the nearest reward or the largest reward.

$$\underset{state}{s} \xrightarrow[\pi]{policy} \underset{action}{a} \quad : \quad \pi(s) = a$$

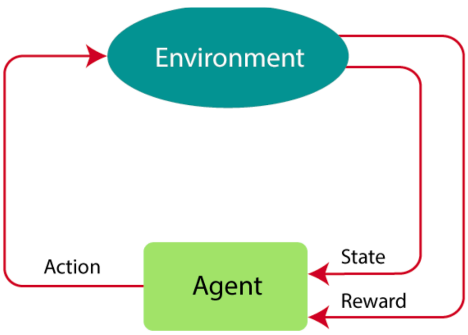
The primary objective is to find a policy that **maximizes** the return by determining the **best action** to take in each state. The term "**policy**" is standard in reinforcement learning, although some may argue that "**controller**" could be a more **intuitive** term.

Review of key concepts

	Mars rover	Helicopter	Chess
→ states	6 states	position of helicopter	pieces on board
→ actions	$\longleftarrow \quad \longrightarrow$	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	$[100] \leftarrow [] \leftarrow [] \leftarrow [40]$	Find $\pi(s) = a$	Find $\pi(s) = a$

Markov Decision Process (MDP)

- Reinforcement learning problems can be modeled as ***Markov Decision Processes***.
- The **Markov property** means the **future state depends only on the current state** and action, not on **past states**.
- This simplifies decision-making as the past history is not explicitly needed.



Why Use Reinforcement Learning?

- Situations like controlling an autonomous helicopter are extremely difficult to solve with supervised learning because the correct action for a given state is ambiguous and hard to label.
- RL focuses on learning what to do through trial, error, and feedback (rewards).
- Analogous to training a dog: rewarding good behavior and discouraging bad behavior without explicitly telling the dog what to do.

You are using reinforcement learning to control a four legged robot. The position of the robot would be its ____.

- return
- reward
- action
- state 👍👍

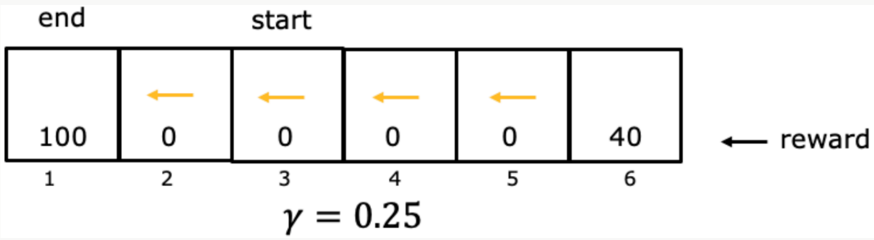
You are controlling a Mars rover. You will be very very happy if it gets to state 1 (significant scientific discovery), slightly happy if it gets to state 2 (small scientific discovery), and unhappy if it gets to state 3 (rover is permanently damaged). To reflect this, choose a reward function so that:

- $R(1) > R(2) > R(3)$, where $R(1)$ and $R(2)$ are positive and $R(3)$ is negative. 👍👍
- $R(1) < R(2) < R(3)$, where $R(1)$ and $R(2)$ are negative and $R(3)$ is positive.
- $R(1) > R(2) > R(3)$, where $R(1)$, $R(2)$ and $R(3)$ are positive.
- $R(1) > R(2) > R(3)$, where $R(1)$, $R(2)$ and $R(3)$ are negative.

You are using reinforcement learning to fly a helicopter. Using a discount factor of 0.75, your helicopter starts in some state and receives rewards -100 on the first step, -100 on the second step, and 1000 on the third and final step (where it has reached a terminal state). What is the return?

- $-100 - 0.25 * 100 + 0.25^2 * 1000$
- $-0.25 * 100 - 0.25^2 * 100 + 0.25^3 * 1000$
- $-100 - 0.75 * 100 + 0.75^2 * 1000$ 👍👍
- $-0.75 * 100 - 0.75^2 * 100 + 0.75^3 * 1000$

Given the rewards and actions below, compute the return from state 3 with a discount factor of $\gamma = 0.25$



- 0.39
- 25
- 6.25 👍👍
- 0

Explain: If starting from state 3, the rewards are in states 3, 2, and 1. The return is $0 + (0.25) \times 0 + (0.25)^2 \times 100 = 6.25$