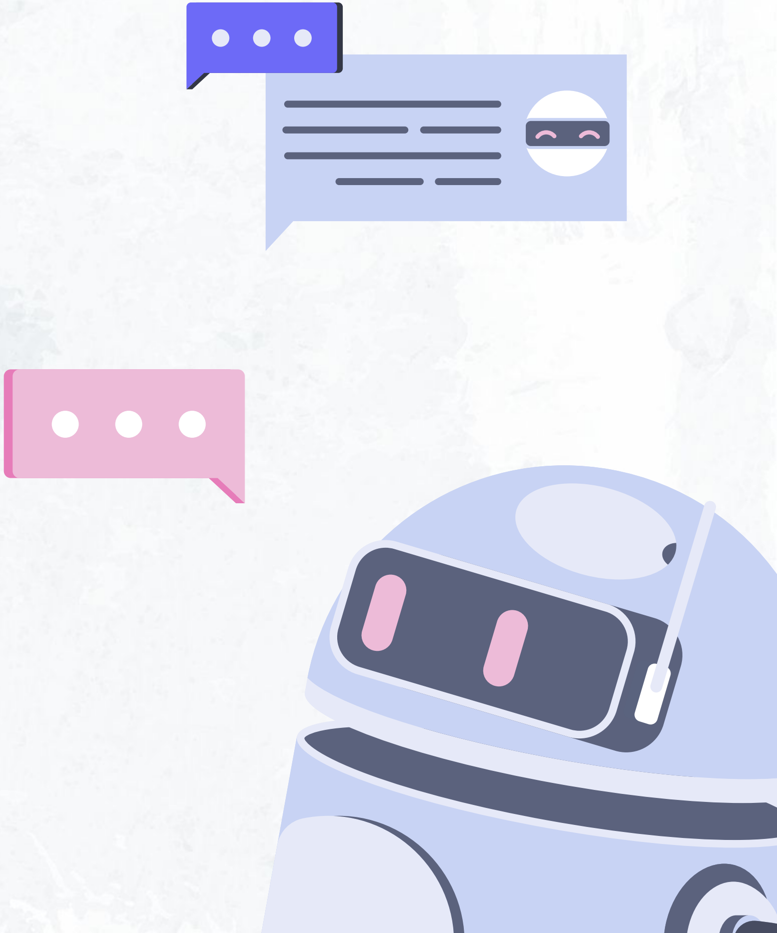
Reinforcement Learning



AI

Table of contents

01 → What is Reinforcement Learning?

02 → Terminologies

03 → Markov Decision Process

04 → Monte Carlo & Q-Learning

01 →

What is Reinforcement Learning?

AI

Reinforcement Learning

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions.
- For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty

02 →

Terminologies

AI

Terminologies

(a) Environment

A Reinforcement Learning Environment is the world context in which the RL agent operates. It's a model or simulation that the agent interacts with.

(b) Agent

An RL Agent is an entity being trained. It is able to perceive and interpret its environment. The agent in RL is the component that makes the decision of what action to take.

(c) Action

An RL agents interacts with the environment using actions.

(d) State

A state in reinforcement learning is a representation of the **current environment** that the agent is in.

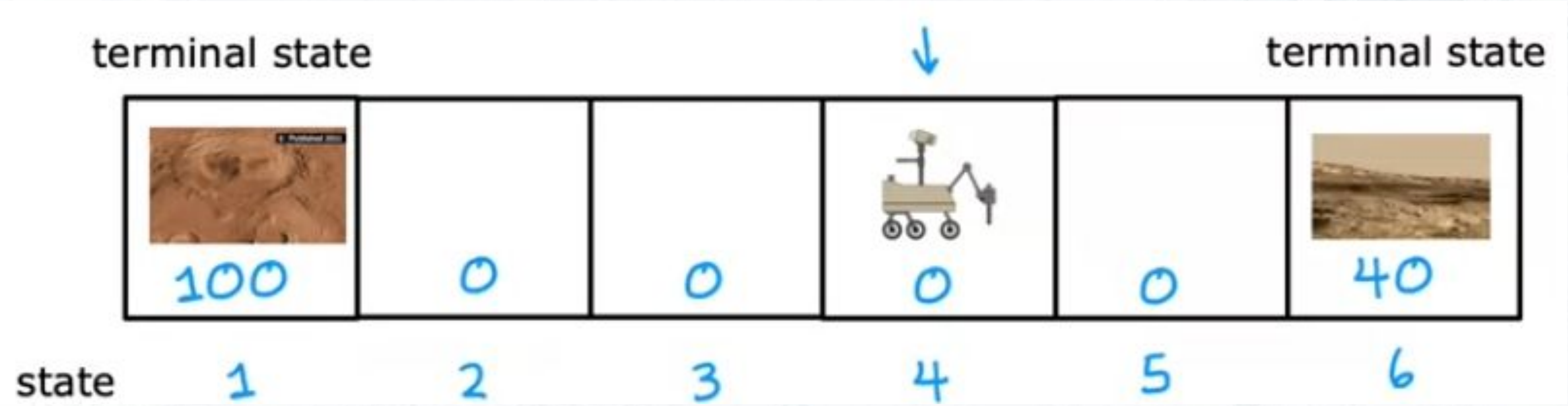
(e) Reward

The Reward measures the immediate effectiveness of taking a particular action. It guides the agent to take correct actions eventually. The objective of the model is to maximize its total reward function(more on this later).

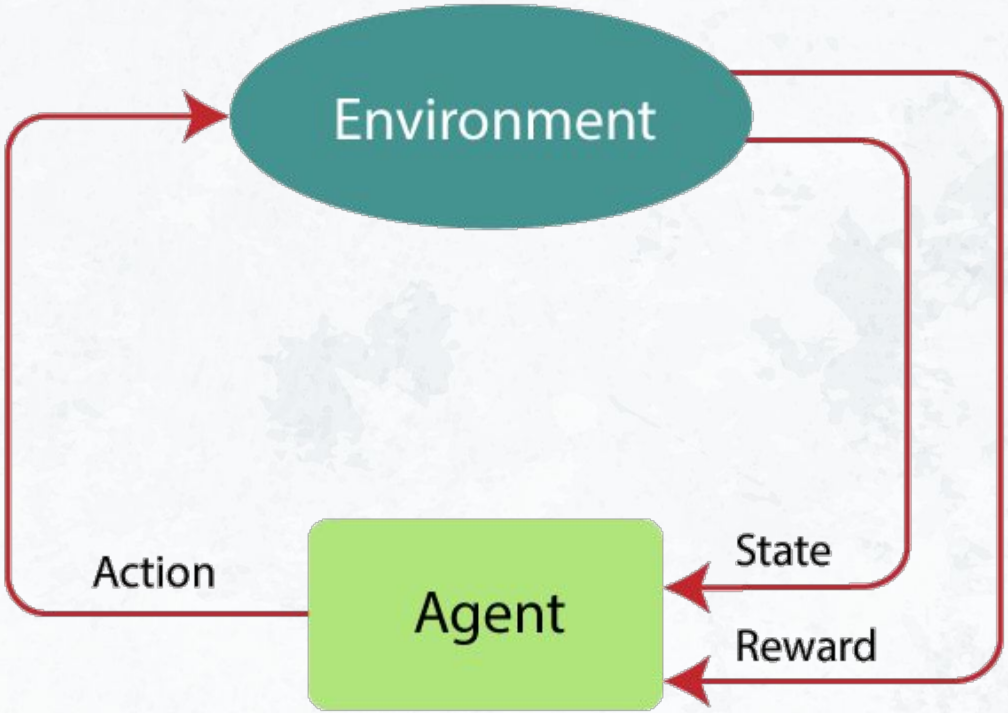
For learning, the model is given a positive reward when it performs correct action. When an incorrect action is taken, a negative reward is given which acts as a punishment.

(f) Discount Factor

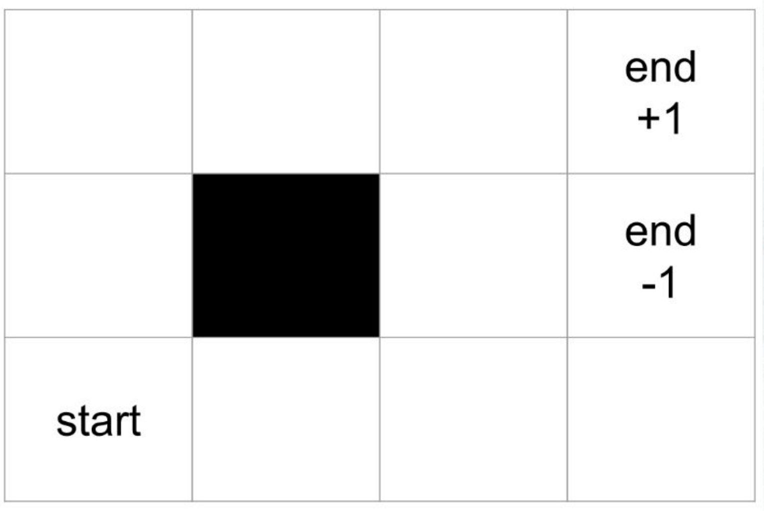
This models the fact that future rewards are worth less than immediate rewards. More often than not, it is just a mathematical constraint on total reward to ensure convergence of algorithms.



	terminal state			↓		terminal state
	100	0	0	0	0	40
state	1	2	3	4	5	6


Environment
Action
Agent
State
Reward

Understanding the terminologies better



			end +1
	■		end -1
start			

The Grid acts as an environment where we let the computer to play to learn. So, the computer acts as an agent.

Understanding the terminologies better

- All the points in the grid represent states. Those states where game ends are called Terminal States.
- **Actions**: Right, Left, Down, Up
- **Rewards**: +1 on of the one terminal states, -1 on other. 0 on the rest.

Can You guess the effects of having this rewards system?

How would agent behave if the reward model is changed to -1 everywhere except terminal states?

03 →

Markov Decision Process

MDP

Markov Decision Process (MDP) is a mathematical framework to describe an environment in reinforcement learning.

Markov Property: Future is Independent of the past given the present.

Mathematical speaking, the best action in a particular state, only depends on that state and none of the previous states.

Reinforcement Learning Environments are designed in such a way that the Markov Property is satisfied, if at any point some extra information about some previous state is required, we must redesign the state itself to hold any such information.

Some more Definitions

(a) Q- function: $q_\pi(s,a)$

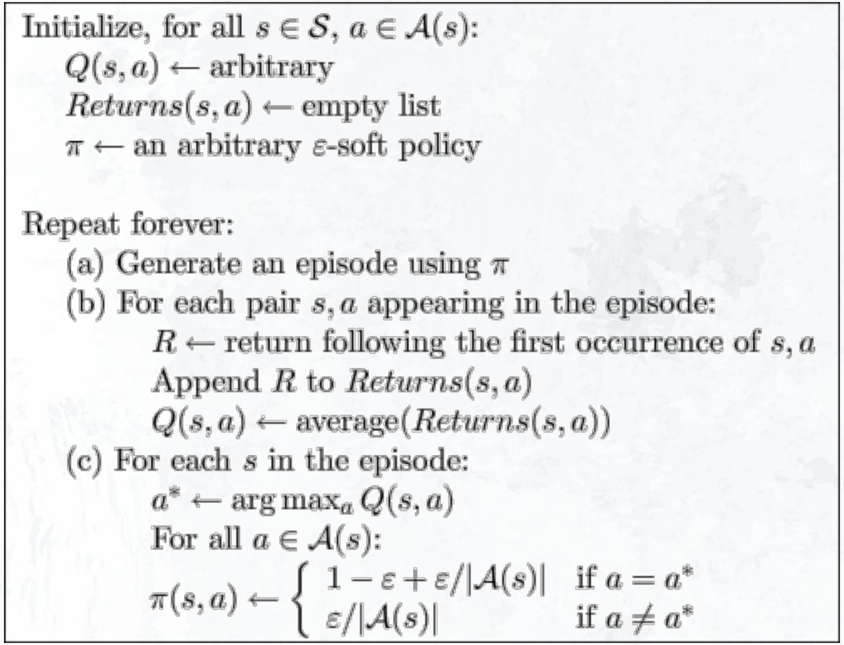
q(s,a) is the expected cumulative reward gained when agent takes action a, starting from state s.

(b) Value function: $v_*(s)$

v(s) is the expected cumulative reward gained when agenet starts from state s.

$$V^*(s) = \max_a Q^*(s,a)$$

Monte Carlo Methods

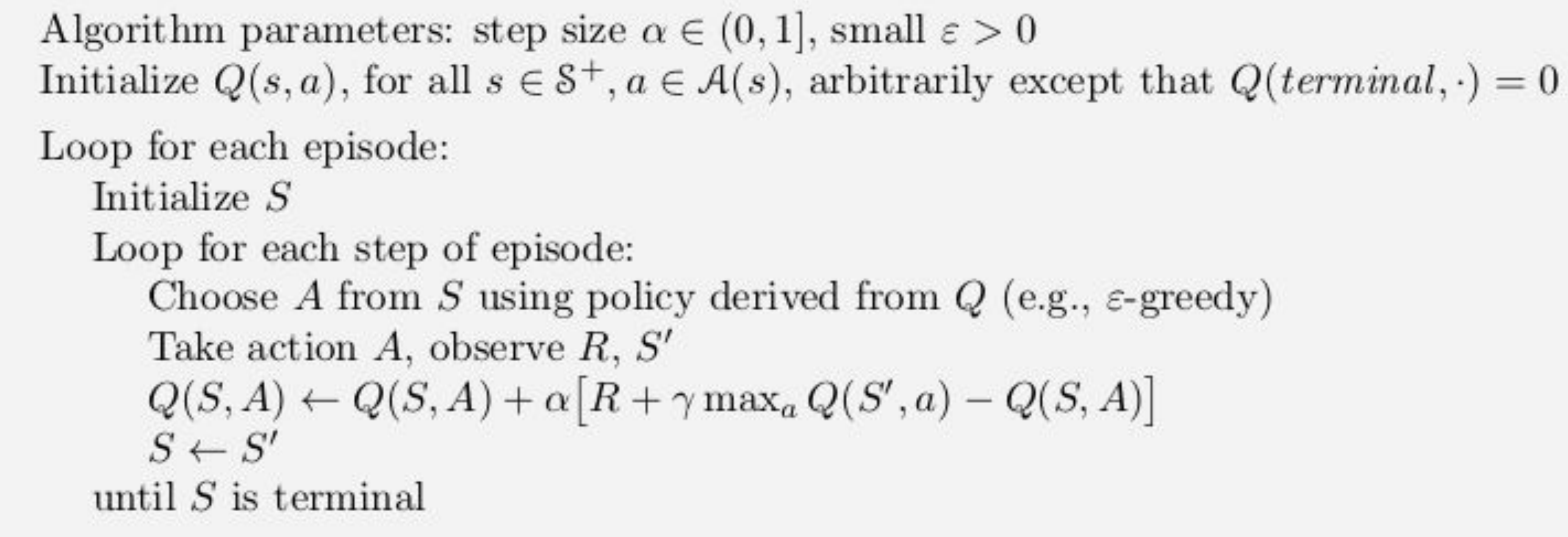


```
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
    $Q(s,a) \leftarrow$ arbitrary
    $Returns(s,a) \leftarrow$ empty list
    $\pi \leftarrow$ an arbitrary $\varepsilon$-soft policy

Repeat forever:
    (a) Generate an episode using $\pi$
    (b) For each pair $s, a$ appearing in the episode:
            $R \leftarrow$ return following the first occurrence of $s, a$
            Append $R$ to $Returns(s,a)$
            $Q(s,a) \leftarrow$ average($Returns(s,a)$)
    (c) For each $s$ in the episode:
            $a^* \leftarrow \arg\max_a Q(s,a)$
            For all $a \in \mathcal{A}(s)$:
            $\pi(s,a) \leftarrow \begin{cases} 1-\varepsilon+\varepsilon/|\mathcal{A}(s)| & \text{if } a = a^* \\ \varepsilon/|\mathcal{A}(s)| & \text{if } a \neq a^* \end{cases}$
```

We will move to the code for better understanding

Q-Learning



```
Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize $S$
    Loop for each step of episode:
        Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        Take action $A$, observe $R$, $S'$
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until $S$ is terminal
```

We will move to the code for better understanding

References

- https://colab.research.google.com/drive/1nTVxfYtHTDwkWWHZTpK84mak3b9US5rd
- https://colab.research.google.com/drive/1l9GQWxMRE9_JyEXxXIKPMZSG75F28l_
- https://towardsdatascience.com/learning-to-win-blackjack-with-monte-carlo-methods-61c90a52d53e?gi=1a93e7648106
- https://www.analyticsvidhya.com/blog/2021/07/a-guide-to-monte-carlo-simulation/
- https://towardsdatascience.com/reinforcement-learning-explained-visually-part-4-q-learning-step-by-step-b65efb731d3e
- https://towardsdatascience.com/q-learning-for-beginners-2837b777741

Resources

- Reinforcement Learning: An Introduction by Andrew Barto and Richard S. Sutton
- https://gymnasium.farama.org/
- https://www.youtube.com/playlist?list=PLZbbT5o_s2xoWNVdDudn51XM8lOuZ_Njv
- https://youtu.be/JgvyzIkgxF0?si=nE3wsNqy1lHPtlHB
- https://youtu.be/Mut_u40Sqz4?si=54rqVmwmA70NmLIH

Follow Us