

**DEEP REINFORCEMENT LEARNING**

**COURSE**

**From zero to RLHF**

**ÍNDICE**

[**Heading 1 2**](#_kb47fxl2pcuk)

# Introduction to Deep Reinforcement Learning

Reinforcement learning is a framework for solving control tasks (also called decision problems) by building agents that learn from the environment by interacting with it through trial and error and receiving rewards (positive or negative) as unique feedback**.**

## Why is the goal of the agent to maximize the expected return?

Because RL is based on the reward hypothesis, which is that all goals can be described as the maximization of the expected return (expected cumulative reward).

That’s why in Reinforcement Learning, to have the best behavior, we aim to learn to take actions that maximize the expected cumulative reward

## Markov property

Markov Property implies that our agent needs only the current state to decide what action to take and not the history of all the states and actions they took before.

## Observations/States Space

Observations/States are the information our agent gets from the environment. In the case of a video game, it can be a frame (a screenshot). In the case of the trading agent, it can be the value of a certain stock, etc.

There is a differentiation to make between *observation* and *state*, however:

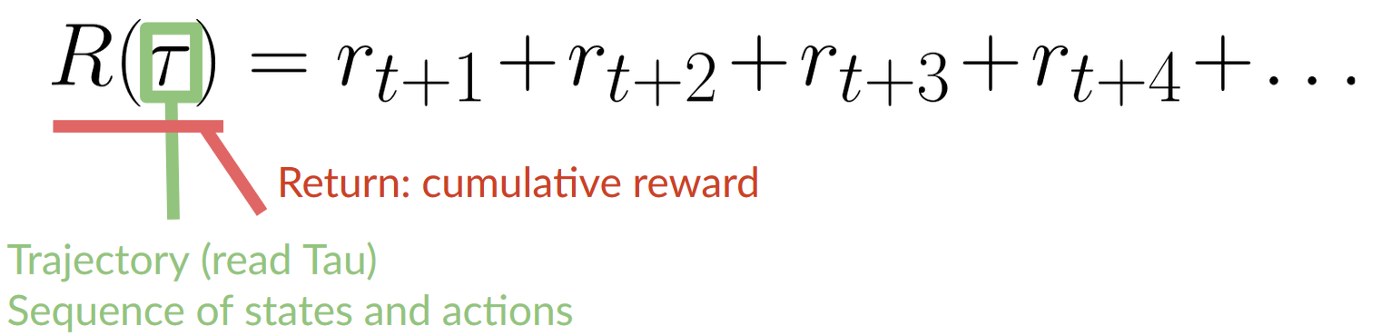
* *State s*: is a complete description of the state of the world (there is no hidden information). In a fully observed environment
* Observation o: is a partial description of the state. In a partially observed environment.

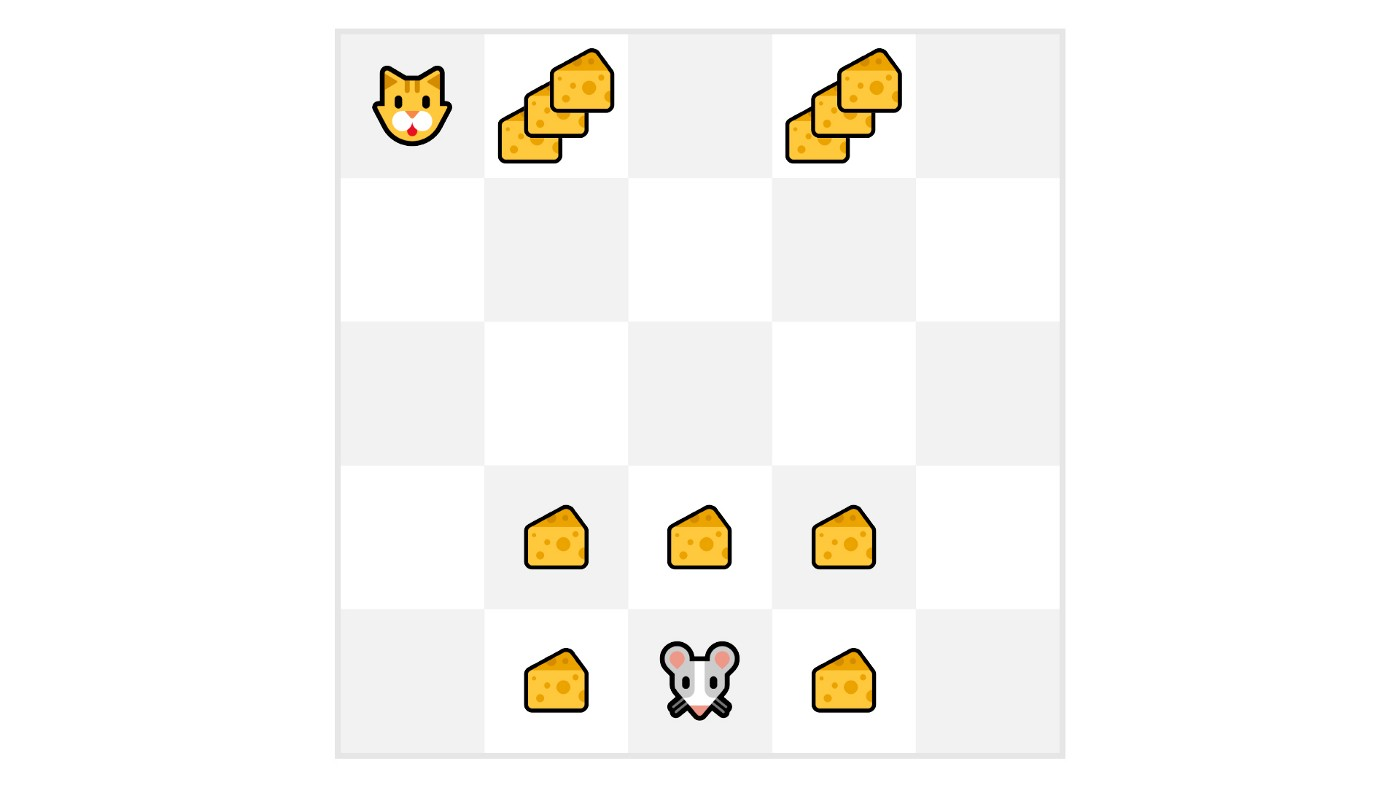
## Action Space

The Action space is a set of all possible actions in an environment.

* Discrete space: number of actions finite. Ej: Game
* Continuous space: number of actions is infinite Ej: Self-driving cars

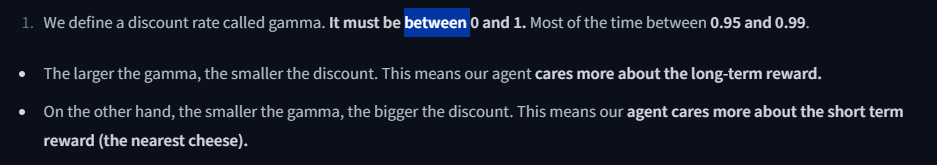
## Rewards and the discounting





As we can see in the diagram, it’s more probable to eat the cheese near us than the cheese close to the cat (the closer we are to the cat, the more dangerous it is).

Consequently, the reward near the cat, even if it is bigger (more cheese), will be more discounted since we’re not really sure we’ll be able to eat it.



## Types of tasks

A task can be episodic or continuous

* Episodic: Starting and ending point (terminal state). Creating an episode. List of states, actions, rewards and new states. Ej: Mario bros game.
* Continuous: No terminal state, learn choosing best action and interact with the environment. Ej: Trading agent.

## Exploration vs Exploitation dilemma

We already know this pretty well. Importance on defining a strong rule to handle this trade-off.

## Learning policy function (**π)**

Policy function is used to estimate which action to take given a current state. Two main approaches:

* **Policy-based method** : Teach agents to learn which action to take.
* **Value-based methods**: Teach agents to learn which state is more valuable, so take action that leads to the state.

## Policy-based methods

Used to learn a policy function directly. With this function we define a mapping from each state to the best corresponding action. Could also define a prob distribution over the actions at a state.

Two types:

* **Deterministic**: Same action for a state.
* **Stochastic**: Prob distribution over actions.

## The “Deep” in Reinforcement Learning

