Reinforcement Learning

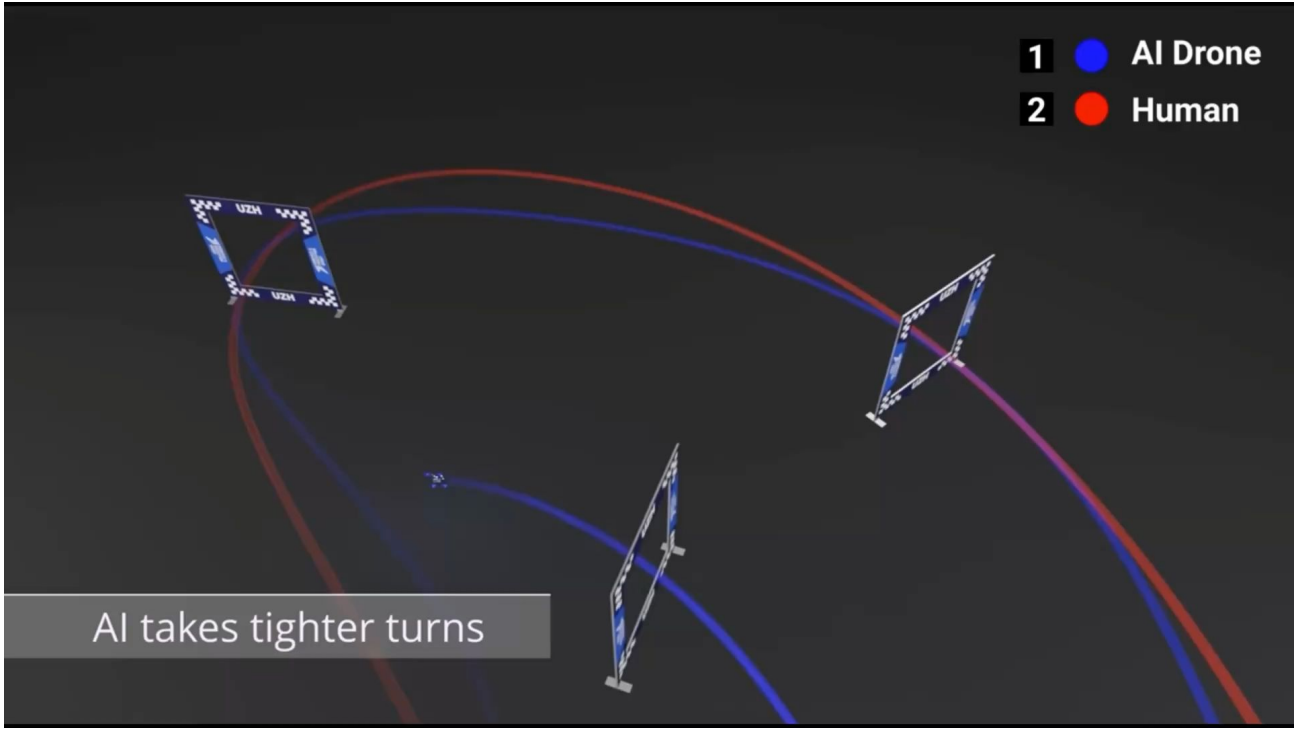
Miłosz Grunwald & Jakub Wilk
Gradient Science Club 2025

Plan for Today

- What is reinforcement learning?
- Policy vs value based methods
- What are environments?
- How to get started with RL?

What is reinforcement learning?

Drone racing

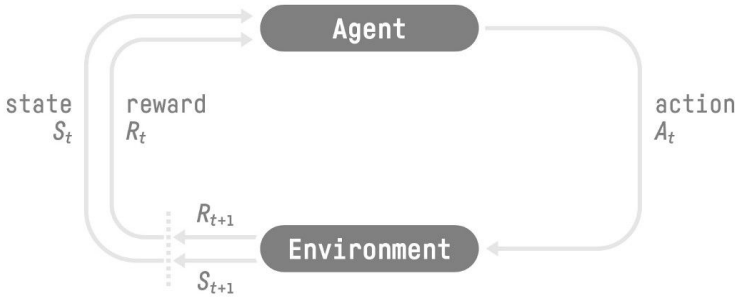


https://www.youtube.com/watch?v=fBiataDpGIo

Reinforcement learning framework

An **agent** interacts with the **environment** through **actions** that he takes based on the **state**, for which he gains **rewards**.

Agent's goal is to maximize the **cumulative** reward.



Supervised vs unsupervised vs reinforcement learning

Unsupervised learning

Learning to capture hidden patterns in data without any labels.

e.g.
Clustering disease symptoms, without knowing its name or characteristics

Supervised learning

Learning what to predict given a specific input, based on a labeled dataset.

e.g.
Learning to classify images as “dog” or “cat” based on a dataset, where for each image we know what the predicted class should be.

Reinforcement learning

Learning what action to take in a specific state to maximize cumulative reward.

e.g.
Learning how to control a drone to complete the entire track in the shortest time.

Unsupervised learning

Supervised learning

Reinforcement learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Unsupervised learning

Supervised learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Unsupervised learning

Supervised learning

Reinforcement learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Unsupervised learning

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Supervised learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Unsupervised learning

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Supervised learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Dane: 20_000 plansz szachowych z zaznaczonym najlepszym możliwym ruchem Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Unsupervised learning

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Supervised learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Dane: 20_000 plansz szachowych z zaznaczonym najlepszym możliwym ruchem
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Dane: Api/Interface/zestaw funkcji pozwalających na wykonywanie ruchów w szachach
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Unsupervised learning

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Supervised learning

Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Dane: Api/Interface/zestaw funkcji pozwalających na wykonywanie ruchów w szachach
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Dane: 20_000 plansz szachowych z zaznaczonym najlepszym możliwym ruchem
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Unsupervised learning

Dane: Oceny od 1 do 10 o 20 różnych filmach w opini 12 różnych osób (.csv 20 wierszy 12 kolumn)
Zadanie: Sprawdzenie jakie filmy są do siebie podobne.

Supervised learning

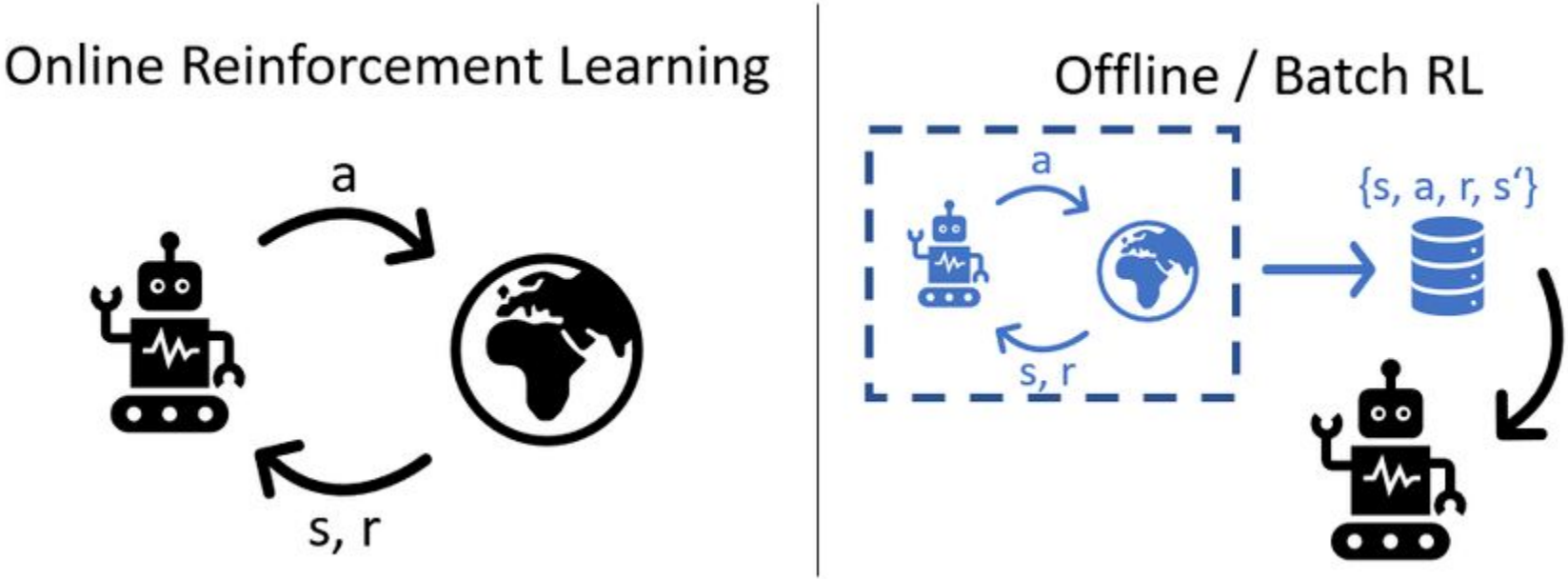
Dane: Długości i szerokości płatków kwiatów razem z id gatunków, bez podanych nazw
Zadanie: Pogrupowanie po gatunku.

Reinforcement learning

Dane: Api/Interface/zestaw funkcji pozwalających na wykonywanie ruchów w szachach
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Dane: 20_000 plansz szachowych z zaznaczonym najlepszym możliwym ruchem
Zadanie: Identyfikacja najlepszego możliwego ruchu w każdej pozycji

Online/Offline RL



Some key concepts

- **Policy** - function that tells us what to do given the state we are in. It can be as simple as a table („brain" of our agent)
- **Observation** – what our agent can see. It can be as simple as one value or as complex as an image of a game
- **State** - all properties of environment, seen by agent or not
- **Reward** – this is the reward that we get at a specific time step
- **Return** – sum of all rewards in an episode

Reward ≠ Return

Reinforcement learning framework

But how to calculate the **cumulative** reward?

We could just add all the rewards obtained during an episode:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots$$

But... we would rather receive big reward now rather than in let's say 100 steps. That's why we use discounted cumulative reward:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

Where $0 < \gamma < 1$ is a discount factor.

What are environments?

Online/offline

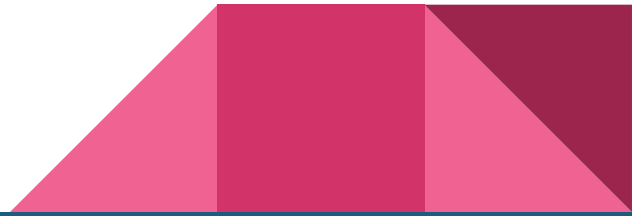
- Online - agent can act in the environment and test out its new strategies
  e.g. RL bot drives it's own car and learns from it

- Offline - a fixed dataset is acquired and given to the agent
  e.g. training RL bot on data from human-driven cars

Discrete/Continuous

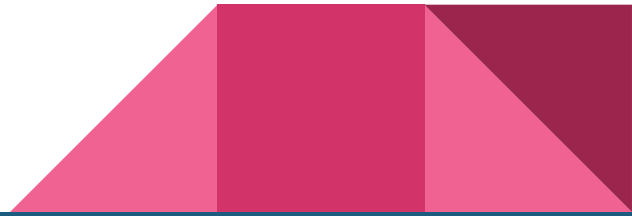
- Discrete - finite set of possible values (or states in general)
  e.g. Tic-tac-toe (kółko i krzyżyk)

- Continuous - infinite set of possible values (or states in general)
  e.g. Pong game agent where either ball or bouncer can be at any point in space



Deterministic/Nondeterministic

- Deterministic - everything happens according to rules, there is no randomness
  e.g. Chess
- Random - not everything can be fully predicted
  e.g. Life
- Stochastic - random but with predefined distribution
  e.g. Ludo (planszówka chińczyk)



Fully/partially observable

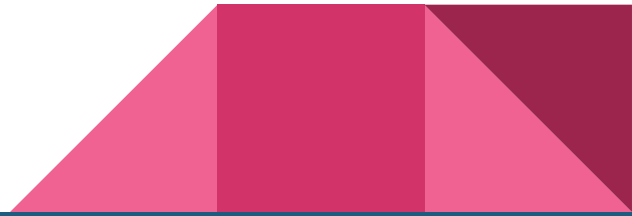
- Fully observable - everything can be seen by agent, observation = state
  e.g.

- Partially observable - not everything can be seen by agent
  e.g. Uno

Single/multi agent

- Single agent - only one agent acts within the environment
  e.g. Traversing a labyrinth

- Multi agent - many agents interact within an environment
  e.g. Poker



Static/Dynamic

- Static - all changes in the environment are made by the agent
  e.g. Solitaire (pasjans)

- Dynamic - world changes without the agent
  e.g. Autonomous car can be approached by a passerby

Policy vs Value based methods

Policy vs value based methods

In reinforcement learning we have two main approaches: policy-based methods and value-based methods.

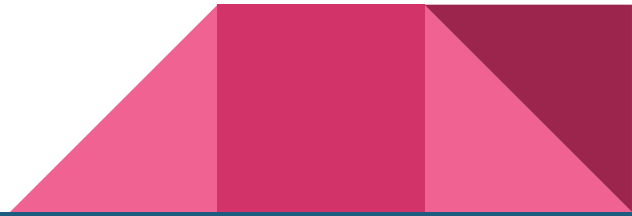
Value-based methods:

- We focus on learning a value function

  (we will explain what a value function is shortly)

Policy-based methods:

- We learn a policy function directly.



Value function

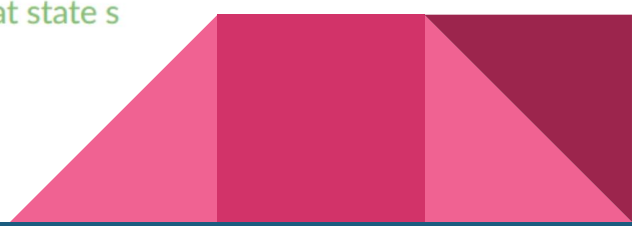
Value function maps a **state** to the expected **value** of being at that state.

But how to define value of being in a state?

By using expected discounted return!

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right]$$

Value function — Expected discounted return — Starting at state s



Expected discounted return

Expected discounted return means the sum of discounted rewards that we get if we start at state s and then act according to our policy.

- If our environment and agent are **deterministic,** we could just run a whole episode once, collect the rewards, discount them and the expected discounted return would be equal to this value
- If our environment or agent are **stochastic,** it is as if we ran infinite number of episodes, collected rewards, discounted them and calculated the mean discounted return that we got in one episode

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s\right]$$

Value function

Expected discounted return

Starting at state s

Value function

What the value function could look like?

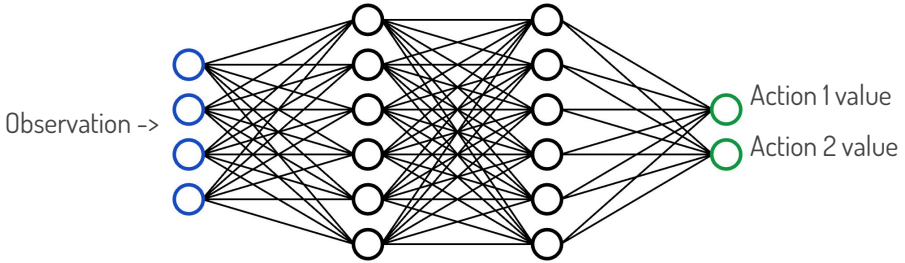
Table:

	Action 1	Action 2
State 1	0.3	0.2
State 2	10.2	2.3
State 3	1.1	9.8

For example we can use argmax as our policy to select our action based on the table.

So if we are in **State 2** we would select **Action 1** as it has a value of 10.2 and Action 2 has only 2.3

Neural network:

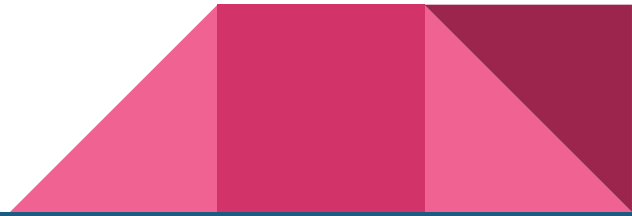


For more complex environments, for example where our observation is an image (it would be hard to store every possible image in a table)

Updating the Q-Table

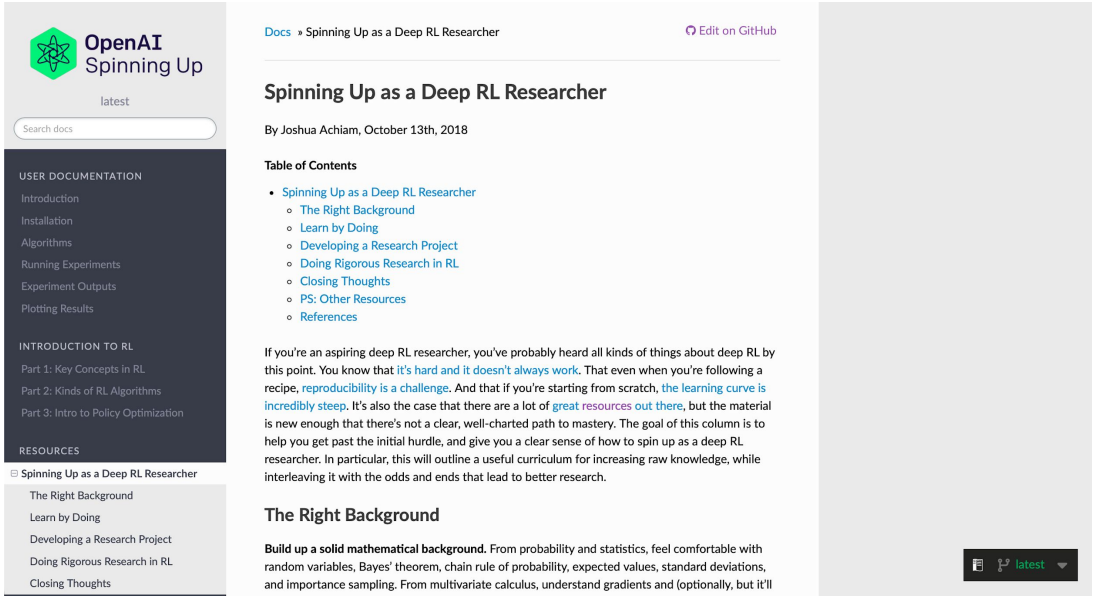
Table:

	Action 1	Action 2
State 1	0.3	0.2
State 2	10.2	2.3
State 3	1.1	9.8



How to get started with RL?

How to get started with RL?



https://spinningup.openai.com/en/latest/

Projects, projects, projects

Questions
&
Discussion

Thank you!
See you next week on Clustering & Deployment.