UPPSALA
UNIVERSITET

Reinforcement Learning

Lecture 1 - Introduction

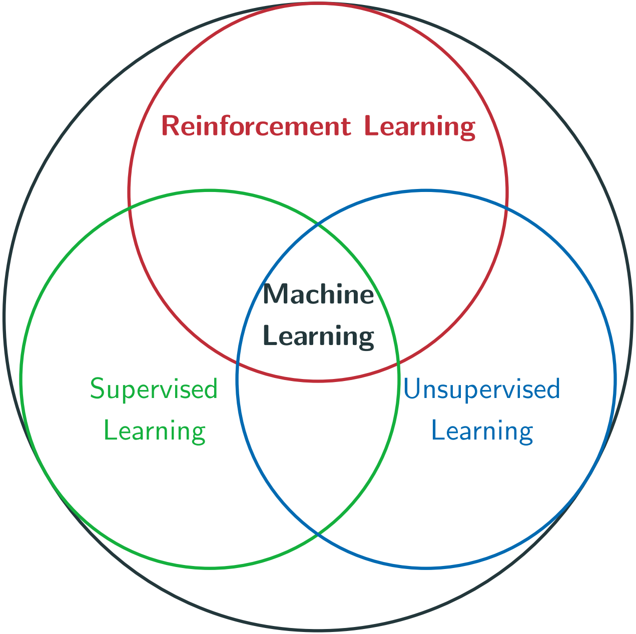
Per Mattsson

2022

Department of Information Technology

Introduction

Branches of Machine Learning



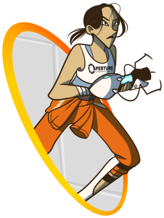
How is RL different?

UPPSALA UNIVERSITET

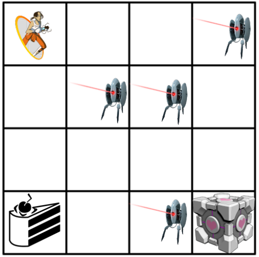
- No supervisor, only a reward signal.
- Feedback is delayed, not instantaneous.
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives.

The agent-environment interface

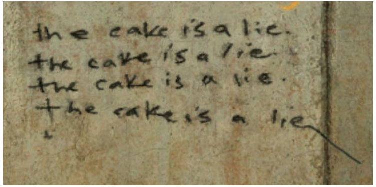
Agent: Decision maker.



Environment: Everything else...



- **Goal:** Maximize the (in some sense) cumulative reward.
- **In RL:** The agent learns how to achieve this from experience.



Examples of Reinforcement Learning

- Playing backgammon on par with top human players (1992).
- Defeating world champions in Go (2016)

(2015)

(2022)

- Navigation of stratospheric balloons (2020).
- Magnetic control of tokamak plasma (2022).

Course information

Teaching

- Lectures.
- Tinkering Notebooks.
- Exercise Sessions. (Zoom + on campus)

All course material can be found on Studium

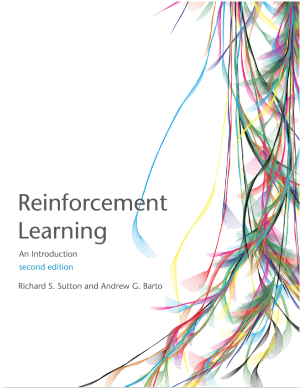
Formalities

- Two courses: 1RT747 (7.5 credits) and 1RT745 (5 credits)
- 1RT747 has an extra lecture and project.
- You have to be registered on the course to get a grade.
- If you quit the course, report to `it-kansli@it.uu.se`.

Reinforcement Learning: An introduction

2nd edition by Sutton and Barto.

Available as a free pdf, see Studium for link.



Examination

UPPSALA UNIVERSITET

The examination consists of two parts (plus a project for 1RT747):

- Assignments.
- Oral exam.

Assignments

UPPSALA UNIVERSITET

Basic Assignments:

- **Mandatory to pass the course!**
- Examined through quizzes on Studium.
- Will be given throughout the course (4 assignments).
- Strongly recommended to finish before soft deadline, must be completed before oral exam.

Advanced Assignment:

- Optional, but...
- ...must pass to get grade 4 or 5 in the course.
- Given towards the end of the course.

Oral exam

Will be given during exam week. We will open up slots that you can book later.

Basic oral exam:

- Must first pass at least all basic assignments.
- If you pass: Grade 3 or Grade 4 (if you also passed advanced assignment)
- A good preparation is to make sure that you understand everything in the assignments.

Advanced oral exam:

- Must have passed both basic and advanced assignments.
- You must be able to show deeper understanding of the course content to pass.
- Can give you grade 5.

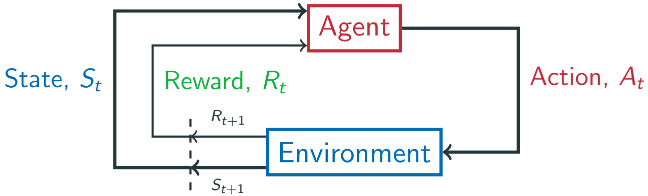
Project (only 1RT747)

UPPSALA UNIVERSITET

- Reading about and implementing deep Q-learning.
- Goal is that an agent should learn to play an Atari game.
- Group work.
- Extra lecture for 1RT747 is a kick-off for the project.

Reinforcement Learning concepts

The agent-environment interface



Agent observes environments state and takes an action $\rightarrow$

Agent receives a reward and environments state changes $\rightarrow$

Agent observes new state and takes an action.

And so on...

In RL, the environment, agent and/or rewards may be stochastic.

Consider two random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are finite sets.

- **Probability:** The probability that we will observe $x \in \mathcal{X}$ is written as

$$\Pr\{X = x\}.$$

- **Conditional probability:** If we have already observed $y \in \mathcal{Y}$, then the probability that we will observe $x \in \mathcal{X}$ is written as

$$\Pr\{X = x | Y = y\}.$$

- **Probability functions:**

$$p(x) = \Pr\{X = x\}, \quad p(x|y) = \Pr\{X = x | Y = y\},$$

UPPSALA UNIVERSITET

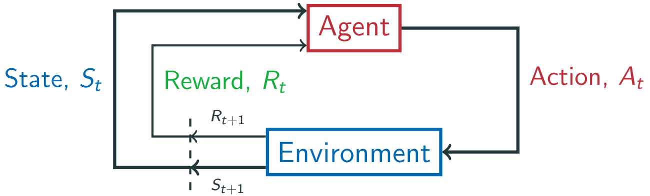
- **Probabilities sum to 1:**

$$\sum_{x\in\mathcal{X}} p(x) = 1, \quad \text{and} \quad \sum_{x\in\mathcal{X}} p(x|y) = 1.$$

- **The expected value:**

$$\mathbb{E}[X] = \sum_{x\in\mathcal{X}} x p(x), \quad \mathbb{E}[X|Y=y] = \sum_{x\in\mathcal{X}} x p(x|y).$$

What is a state?



- State, S_t: A representation of the environment at time t.
- State space, $\mathcal{S}$: The set of all possible states.
- S_t will depend on what happened in the past (before time t).
- S_t contains all information that is relevant to predicting the future at time t.

The Markov Property

$$p(S_{t+1}|S_0, A_0, S_1, A_1 \ldots, S_t, A_t) = p(S_{t+1}|S_t, A_t).$$

Example: The Taxi Environment

- Taxi: 25 different positions.
- Passenger: 5 different positions (including picked up).
- Destinations: 4 possible options.
- In total: $25 \times 5 \times 4 = 500$ possible configurations.
- We can, in some way, enumerate all the 500 possible configurations.
- Then let $\mathcal{S} = \{0, 1, \dots, 499\}$.

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

In this case we have a *finite* state space.

UPPSALA UNIVERSITET

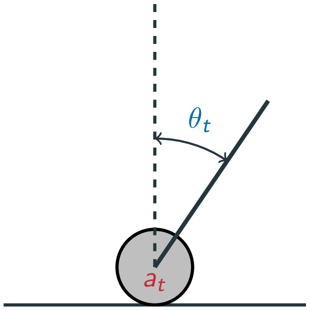
- Balancing a stick. The control signal (action) is the torque at the bottom.
- State?

$$S_t = \theta_t.$$

No! Does not indicate what direction the stick is moving in!

- State: For example

$$S_t = \begin{bmatrix} \theta_t \\ \frac{d\theta_t}{dt} \end{bmatrix} \in \mathcal{S} \subset \mathbb{R}^2.$$



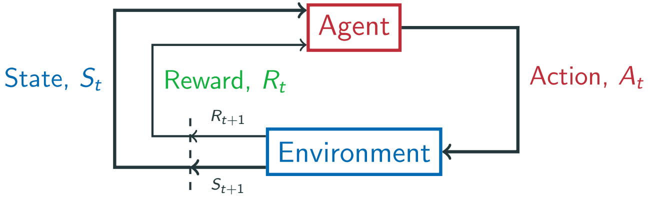
Continuous state space. (Infinitely many possible states)

- Many real-world systems can be approximated quite well as a linear system:

$$S_{t+1} = FS_t + GA_t + W_t$$

  where W_t is some stochastic white noise (W_t does not depend on the past).
- **Note:** S_{t+1} is not determined exactly by S_t (due to the white noise W_t), but S_t is a state since we cannot improve our predictions of the future even if we also know $S_{t-1}, S_{t-2}, \ldots$.

UPPSALA UNIVERSITET



- Reward, R_t: A scalar signal that tells it how it is doing at step t.
- **Goal:** Maximize the long-time cumulative reward.

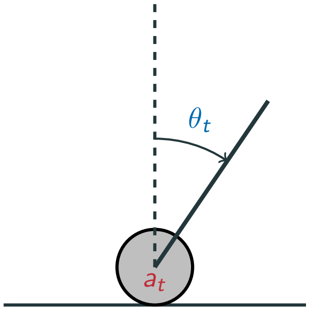
- Illegal 'pick-up' or 'drop-off' gives -10 reward.
- Successful 'drop-off' gives $+20$ in reward.
- All other actions gives -1 in reward.

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
```

Hence, to maximize the total reward we should deliver the passenger in as few steps as possible.

Example: Inverted pendulum

- We want to keep the angle close to zero, so maybe
$$R_{t+1} = -\theta_t^2.$$
- If we also want to use a low torque, maybe we can try
$$R_{t+1} = -c_1\theta_t^2 - c_2 a_t^2.$$



With $\theta_t = 0$ and $a_t = 0$ we thus get the maximum reward $R_{t+1} = 0$.

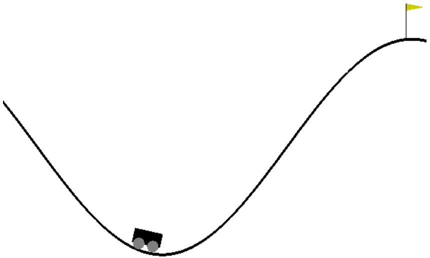
Sequential Decision Making

- **Goal:** Select actions to maximize *future* reward.
- Actions may have long term consequences.
- Reward may be delayed.
- It may be better to sacrifice immediate reward to gain more long-term reward.
- Examples:
  - A financial investment (may take months to mature)
  - Refuelling a helicopter (might prevent a crash several hours later)
  - etc

Designing reward functions

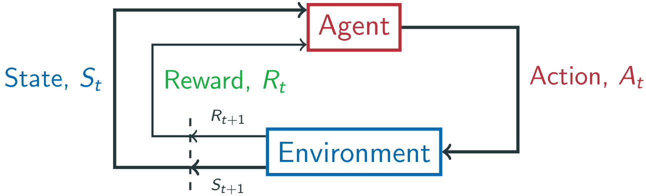
- In this course the reward function will often be given.
- However, when RL is used in practice, the design of the rewards is very important:
- It will determine *what* the agent tries to achieve,
  - The agent may find unintended ways to increase the reward.
- It will influence *how* the agent learns.

Example: MountainCar



- **Goal:** Reach the flag in as few steps as possible.
- **Reward:** -1 for each step until you reach flag.
- **Sparse reward:** All actions look equally bad until you reach the flag the first time.
- Maybe possible to use a more informative reward function? But then we must make sure that the agent still optimize the correct thing...

What is an action?



- Action, A_t: The agent can affect the state by actions.
- Action space, $\mathcal{A}$: The set of all possible actions.

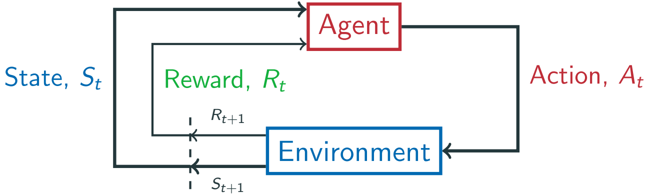
Examples:

- Taxi environment: Go north, south, west, east, pickup or drop-off. ($|\mathcal{A}| = 6$).
- Inverted pendulum: The action is the torque, and thus $\mathcal{A}$ is a continuous set of actions.

The power of feedback

- **Deterministic:** Given S_0 we can pre-compute the optimal actions $A_0, A_1, \ldots$.
- **Stochastic:** The states that the actions will result in is random.
- **Feedback:** Decide on action A_t after we have observed S_t.
- Hence, instead of trying to find good actions, we will try to find a good policy.

What is a policy?



- Policy: Formally a distribution over actions given states

$$\pi(a|s) = \Pr\{A_t = a | S_t = s\}.$$

- A policy defines the agents behavior in different states.
- When the policy is deterministic, we sometimes write

$$a = \pi(s)$$

$$S_{t+1} = FS_t + GA_t + W_t.$$

Assume that we want to maximize

$$\mathbb{E}\left[\sum_{t=0}^{\infty}(-c_1 S_t^\top S_t - c_2 A_t^\top A_t)\right]$$

Optimal policy: When we have observed s_t, choose the action

$$a_t = \pi(s_t) = -Ls_t$$

for some fixed L (that can be found if F and G are known).

Terminology: Reinforcement Learning vs Optimal Control

Reinforcement Learning	Optimal Control
Environment	System / Plant
State, S_t	State, $x(t)$
Action, A_t	Input, $u(t)$
Reward, R_t	Cost, $c(t)$
Policy	Controller
Maximize reward	Minimize cost
Learn policy from experience	Use model to find controller (e.g LQG or MPC)

Problems in RL

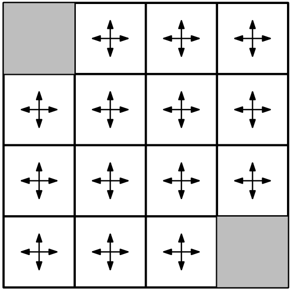
"Evaluate the future given a policy"

Example: GridWorld.

Reward: -1 for each action until you reach a gray square.

Policy: Uniform random (choose all equations with same probability)

Policy:



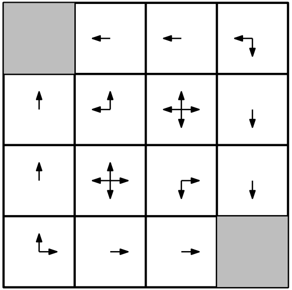
Expected total reward:

	-14	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	

UPPSALA UNIVERSITET

"Optimize the future (find the best policy)"

Optimal policy:



Expected total reward:

	-1	-2	-3
-1	-2	-3	-2
-2	-3	-2	-1
-3	-2	-1	

Exploration vs Exploitation

- The agent should learn a good policy without losing to much reward.
- **Exploration:** Learn more about the environment.
- **Exploitation:** Use the information you have to maximize reward.
- We usually have to both explore and exploit.

Example: Go out for dinner

- Exploitation: Buy my (current) favourite meal at my favourite restaurant.
- Exploration: Try a new restaurant/meal.

Model-based vs model-free

Model-based RL:

- Learn a model from experience.
- Use model to find good policy and/or predictions.

Model-free RL:

- Learn policy and/or predictions directly, without first learning a model.

In this course the main focus will be on model-free RL, but we will also look at some model-based methods.

Summary

- Course information.
- Some fundamental RL concepts (state, reward, action etc).

Whats next?

- Start working on Tinkering Notebook 1. If you have questions, join the exercise session.
- Next lecture: Markov Decision Processes. (Chapter 3)