Machine Learning

Università Roma Tre
Dipartimento di Ingegneria
Anno Accademico 2021 - 2022

Reinforcement Learning

Acknowledgements, sources and links

Organization, content and images of the slides are extracted from the following sources:

- Reinforcement Learning: An Introduction. Richard S. Sutton and Andrew G. Barto, second edition, 2018.
- Implementation of Reinforcement Learning algorithms, from Sutton-Barto's book. Denny Britz, GitHub project, 2016.
- Tutorial: Introduction to Reinforcement Learning with Function Approximation. Richard S. Sutton, 2016.
- UCL Course, Reinforcement Learning, videos and slides. David Silver, 2015.
- UCL course, Advanced Deep Learning & Reinforcement Learning, videos and slides. DeepMind, 2018.

1. What is Reinforcement Learning?
2. The RL setup: problem and actors
3. What do we know? State and observability
4. What can we do? Policy and value - and model?
5. The never-ending control loop: prediction = improvement

1. What is Reinforcement Learning?
2. The RL setup: problem and actors
3. What do we know? State and observability
4. What can we do? Policy and value - and model?
5. The never-ending control loop: prediction = improvement

The RL problem

Important points

- Trying to reach a **goal**
- Interactions: active **decision-making agent vs environment**
- **Uncertainty** about the environment
- Effects of actions cannot be fully predicted: adaptation required (**learning**)

The RL reward hypothesis

All goals can be described by the maximization of some expected cumulative **reward**

- Is it true? Interesting analysis at http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html
- Related with the **expected utility hypothesis** from von Neumann-Morgenstern utility theory

The RL problem

RL main task

Decision problem: we would like to choose actions that maximize the **return**, i.e. the total future reward

Sequential decision making

Actions may have long term consequences

Uncertainty

The best we can aim for is maximizing the **value**, i.e. the **expected** total future reward

Exercise

Find an example of a **deterministic** task, that is, a task where your actions gives a fixed outcome (that you may or may not know in advance)

The RL problem

To be **greedy** can be wrong

- A financial investment (may take months to mature)
- Refuelling a helicopter (might prevent a crash in several hours)
- Blocking opponent moves (might help winning chances many moves from now)

Exercise

Discuss the difference between return and value

The RL problem

Examples of reward

- Games: $R_T := -1,\ 0,\ +1$ (win, draw, lose). More generally,
- R_T can be the final score
- Games: $R_T := 0,\ +1$ (win, lose). In this case, the value is the probability of winning. Why?
- Atari games: R_t is the immediate score increment at step t
- Walking robot: $R_t := +1$ for every step he doesn't fall

https://www.youtube.com/watch?v=gn4nRCC9TwQ.

- Financial investment: R_t is the money increment in the last time step in portfolio
- Maze and Gridworld: +100 for reaching the exit, 0 otherwise. Wrong. Why?

The RL problem

Examples of reward

- Games: $R_T := -1,\ 0,\ +1$ (win, draw, lose). More generally,
- R_T can be the final score
- Games: $R_T := 0,\ +1$ (win, lose). In this case, the value is the probability of winning. Why?
- Atari games: R_t is the immediate score increment at step t
- Walking robot: $R_t := +1$ for every step he doesn't fall

https://www.youtube.com/watch?v=gn4nRCC9TwQ.

- Financial investment: R_t is the money increment in the last time step in portfolio
- ~~Maze and Gridworld: +100 for reaching the exit, 0 otherwise. Wrong. Why?~~
- Maze and Gridworld: -1 for every move. Correct. Why?

Examples

Games

TD-Gammon, 1995, ACM Communications.

Atari's family (video).

- 49 out of 57:DQN, 25 Feb 2015, Nature.
- 52 out of 57:R2D2, Sep 2018, ICLR 2019.
- 51 out of 57:MuZero, Nov 2019, arXiv.
- 57 out of 57:Agent57, Mar 2020, arXiv.

AlphaGo's family (video).

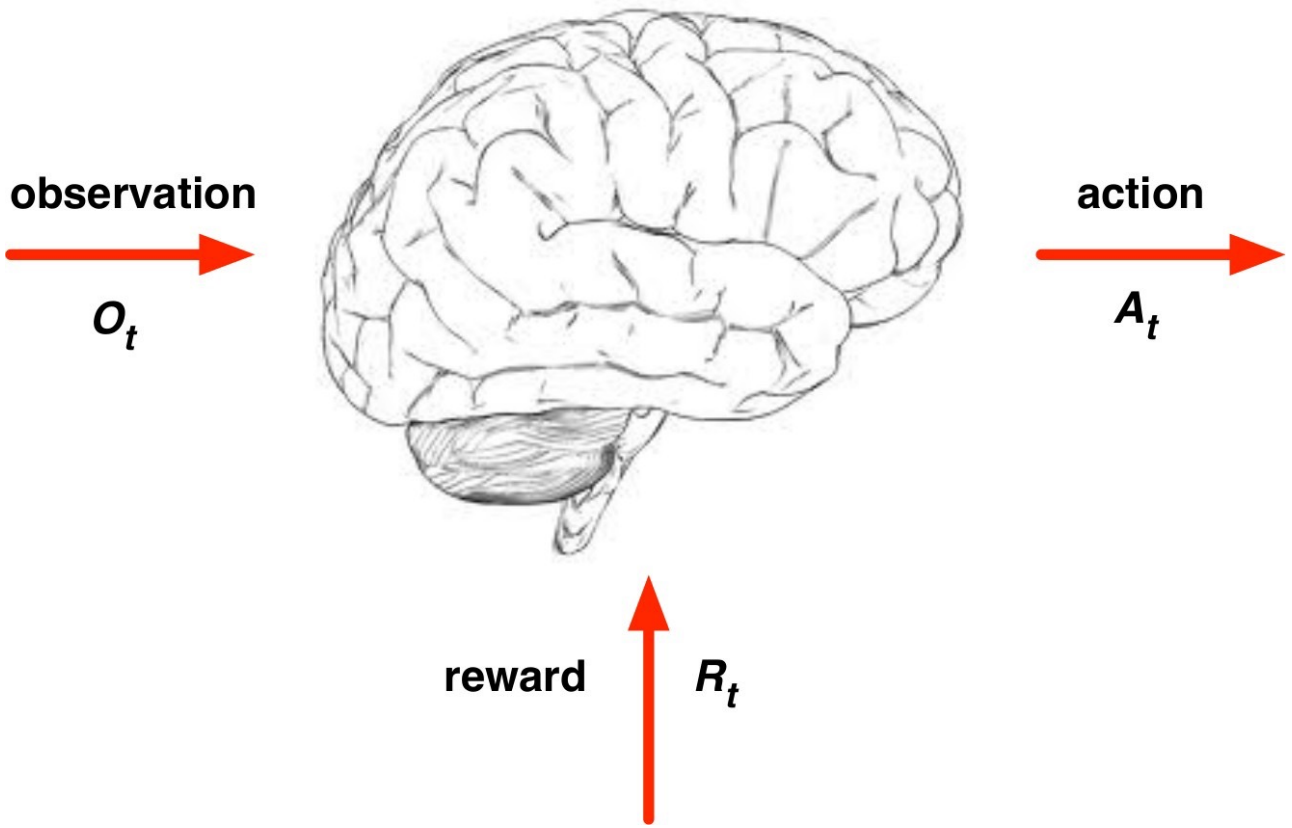
- AlphaGo, 27 Jan 2016, Nature.
- AlphaGo Zero, Oct 2017, Nature.
- AlphaZero, Dec 2018, Science.
- MuZero, Nov 2019, arXiv.

StarCraft II (video).AlphaStar, Nov 2019, Nature.

Protein folding

How a protein's amino acid sequence dictates its three-dimensional structure? AlphaFold: Oct 2019, PROTEINS; Jan 2020, Nature.

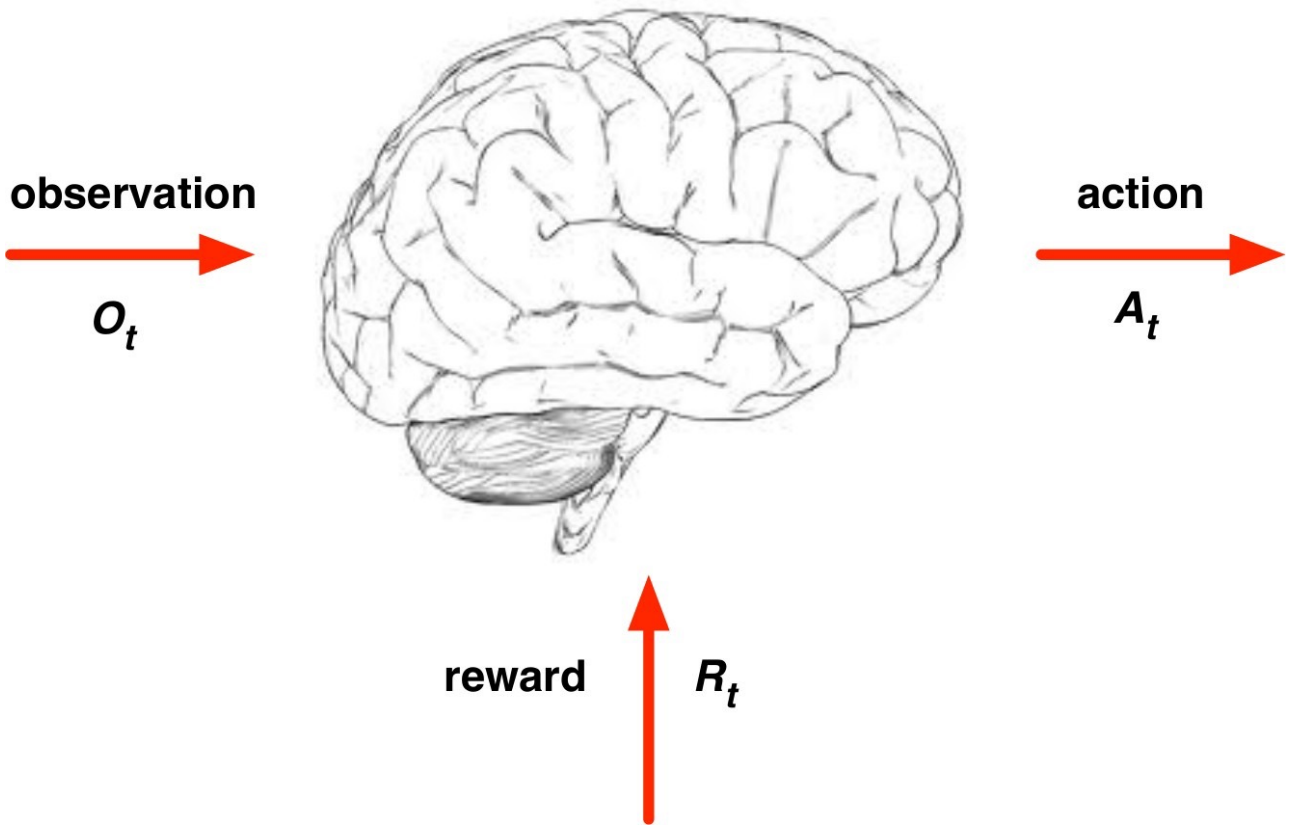
First actor: the **agent**



A never-ending loop

- … we (the agent) receive R_t and observe O_t…
- … we choose the action $A_t \sim \pi(\cdot, f(O_t, R_t, A_{t-1}, O_{t-1}, R_{t-1}, \ldots))$…
- … and because of our action A_t, the environment send us a reward
- R_{t+1} and a new **state**, that we observe as O_{t+1}. ..

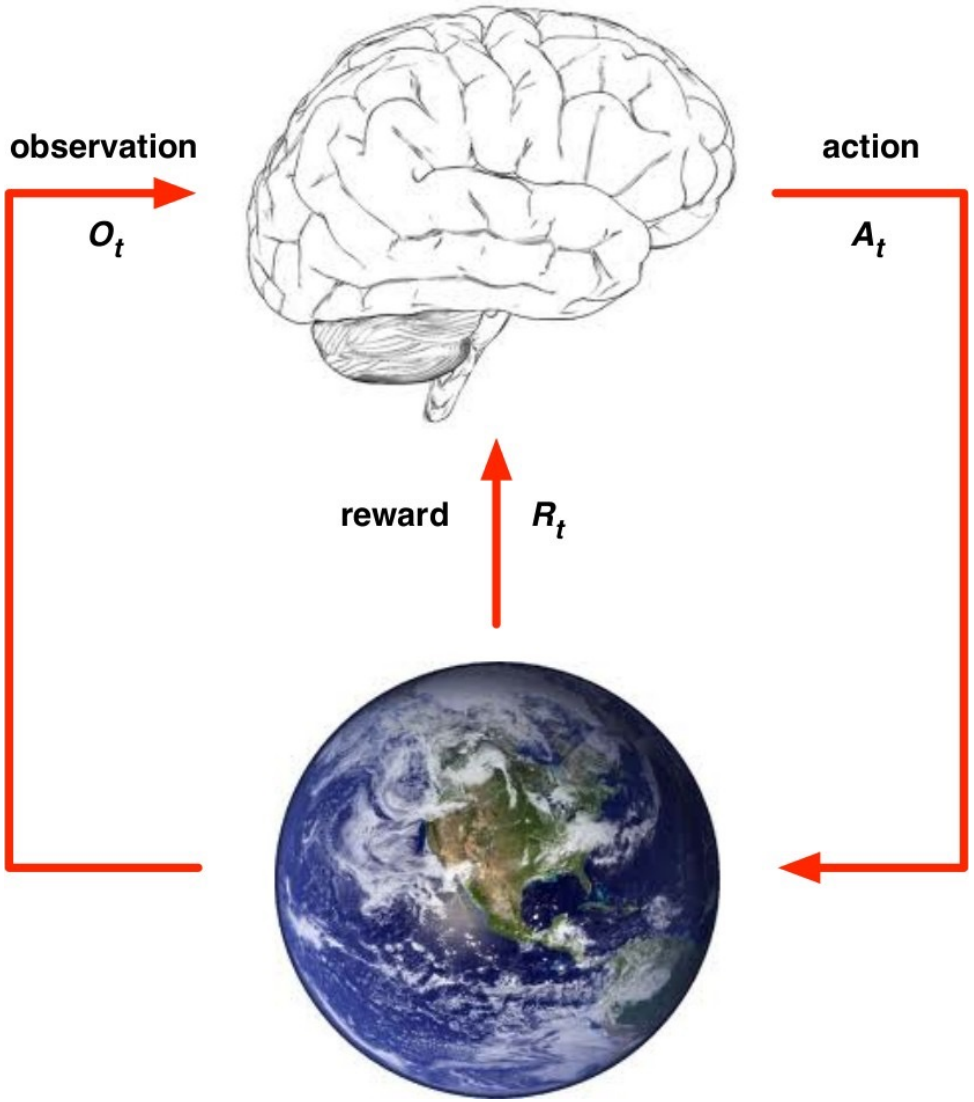
First actor: the **agent**



A never-ending loop

- … we (the agent) receive R_t and observe O_t…
- … we choose the action $A_t \sim \pi(\cdot, f(\textbf{history}))$…
- … and because of our action A_t, the environment send us a reward
- R_{t+1} and a new **state**, that we observe as O_{t+1}. ..

Not alone! Second actor: the **environment**



Agent, step *t*

- Receives observation O_t
- Receives scalar reward R_t
- Computes his own **state** S_t^a
- Executes action A_t.

Environment, step *t*

- Receives action A_t
- Computes his own **state** S_{t+1}^e
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}

1. What is Reinforcement Learning?
2. The RL setup: problem and actors
3. What do we know? State and observability
4. What can we do? Policy and value - andmodel?
5. The never-ending control loop: prediction = improvement

History, agent state, environment state

Notation

- **History**: the sequence of observations, actions, rewards up to time step t:

$$H_t := O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$$

- The agent selects actions, and the environment answers with **observations** and **rewards**
- **State**: the information used (by the agent and the environment) to determine what happens next
- State is naturally a sequence S_t
- Agent state is a function of history: $S_t := f(H_t)$
- **Environment state** S_t^e is different from **agent state** S_t^a

Markov state

Uncertainty

Since we have no control of environment, everything is a **random variable**

Definition

A sequence of states (random variables) is **Markov** if and only if

$$\Pr(S_{t+1}/S_t) = \Pr(S_{t+1}/S_1, \ldots, S_t)$$

- The future is independent of the past given the present:

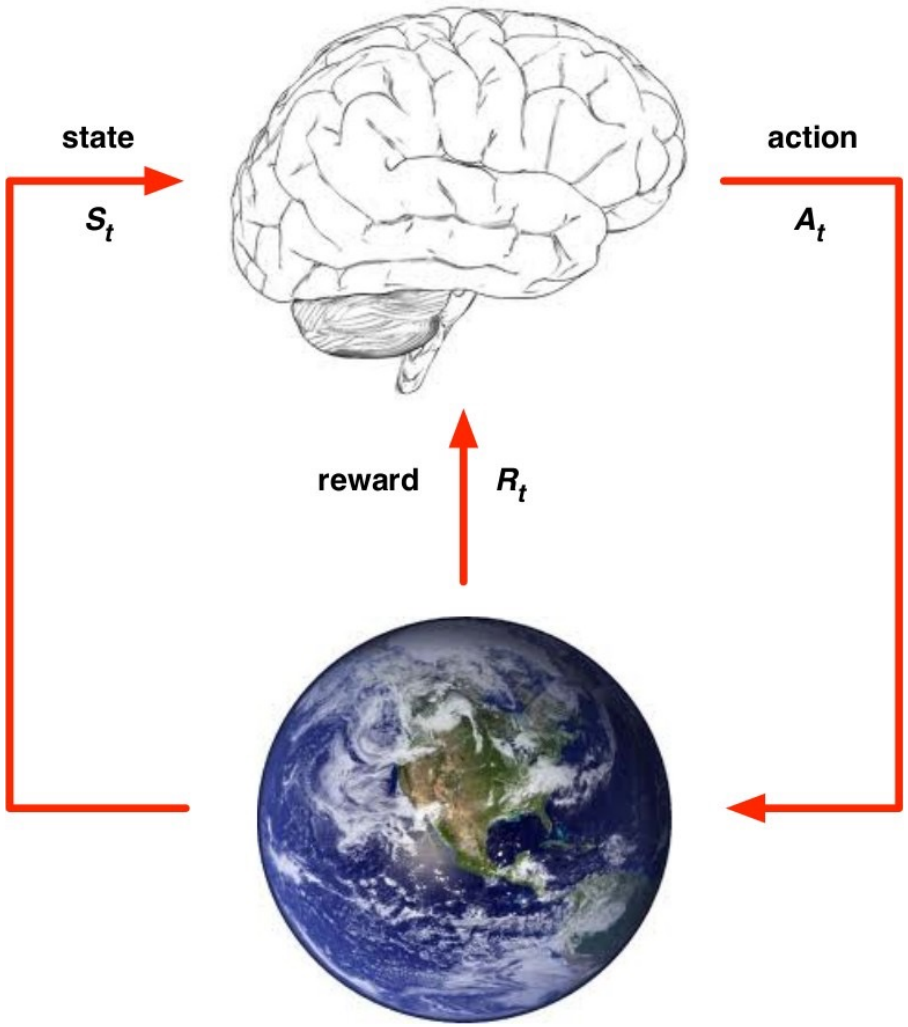
$$S_t \rightarrow H_{t+1:+\infty}$$

- Once the state is known, the history may be thrown away: the state is a sufficient statistic of the future

Exercise

Is the environment state S_t^e Markov? Is the history H_t Markov?

Fully observable environments

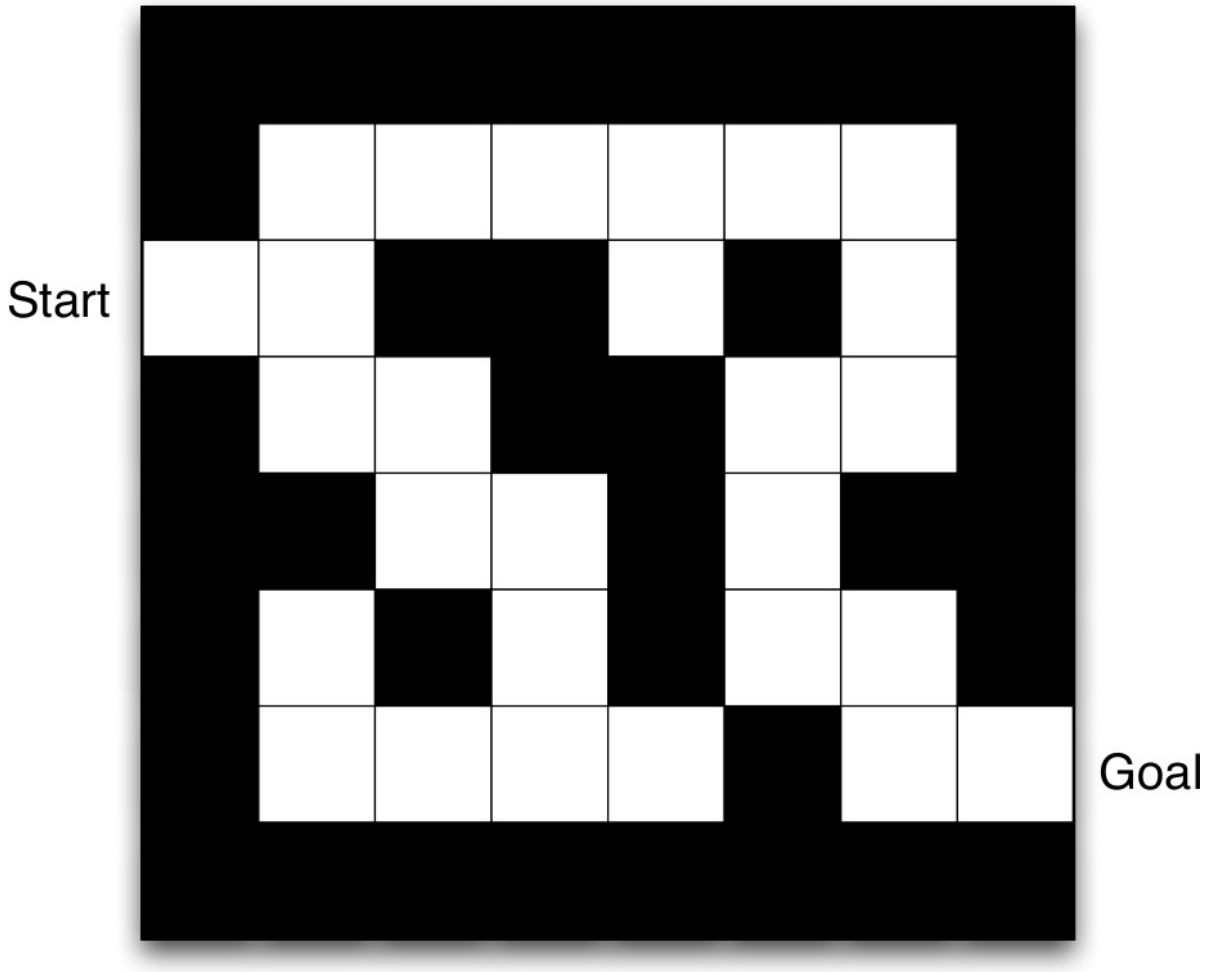


- Agent observes environment state: $O_t = S_t^a = S_t^e$
- Agent state and environment state coincides!

A never-ending loop

- ...we (the agent) receive R_t and observe S_t...
- ...and thus we decide to do action $A_t \sim \pi(\cdot, S_t)$...
- ...and environment answers A_t with a new reward, state pair R_{t+1}, S_{t+1}...

Example: the maze

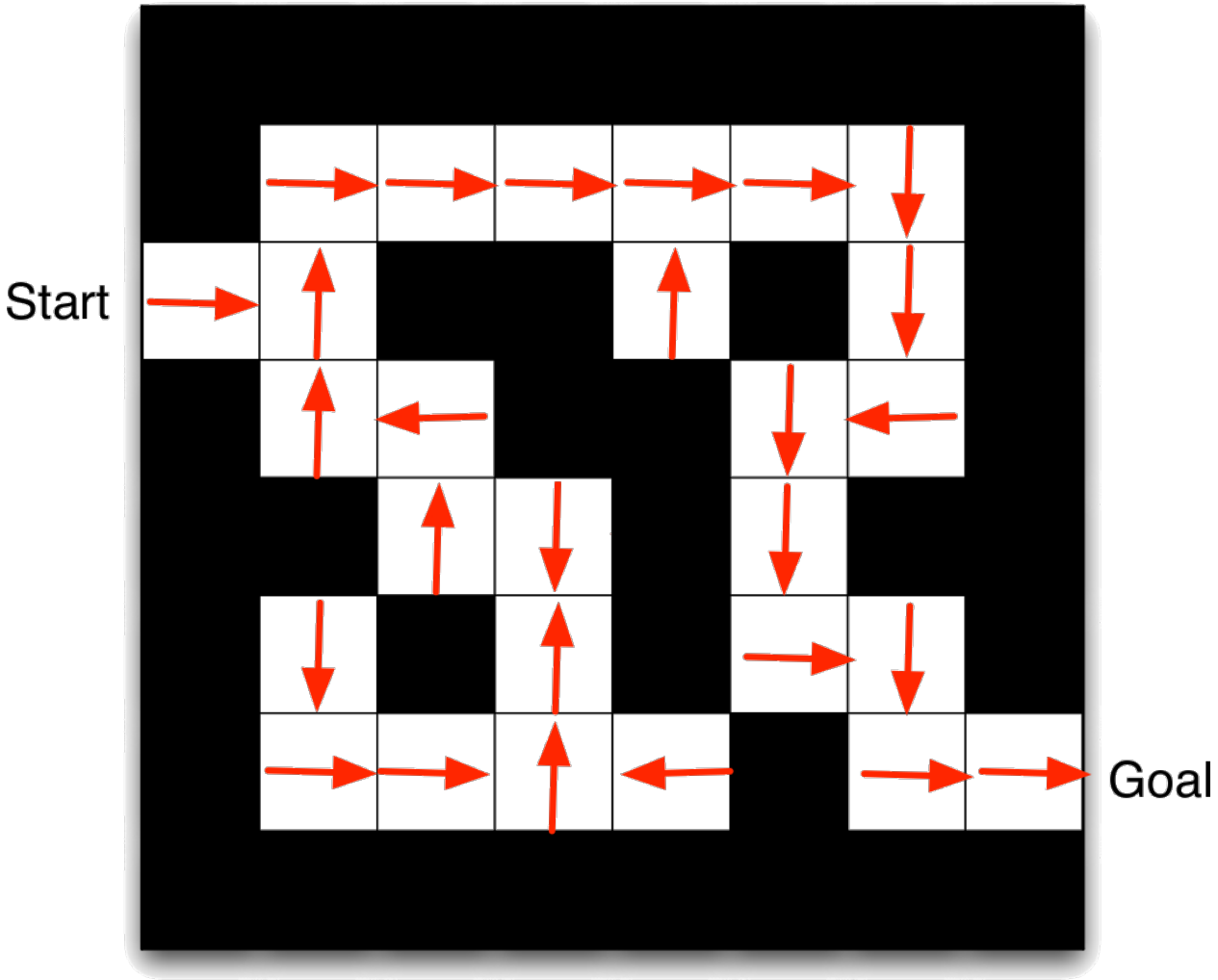


Exercise

Discuss this example in terms of the language you have learned up to now.

1. What is Reinforcement Learning?
2. The RL setup: problem and actors
3. What do we know? State and observability
4. What can we do? Policy and value - and model?
5. The never-ending control loop: prediction = improvement

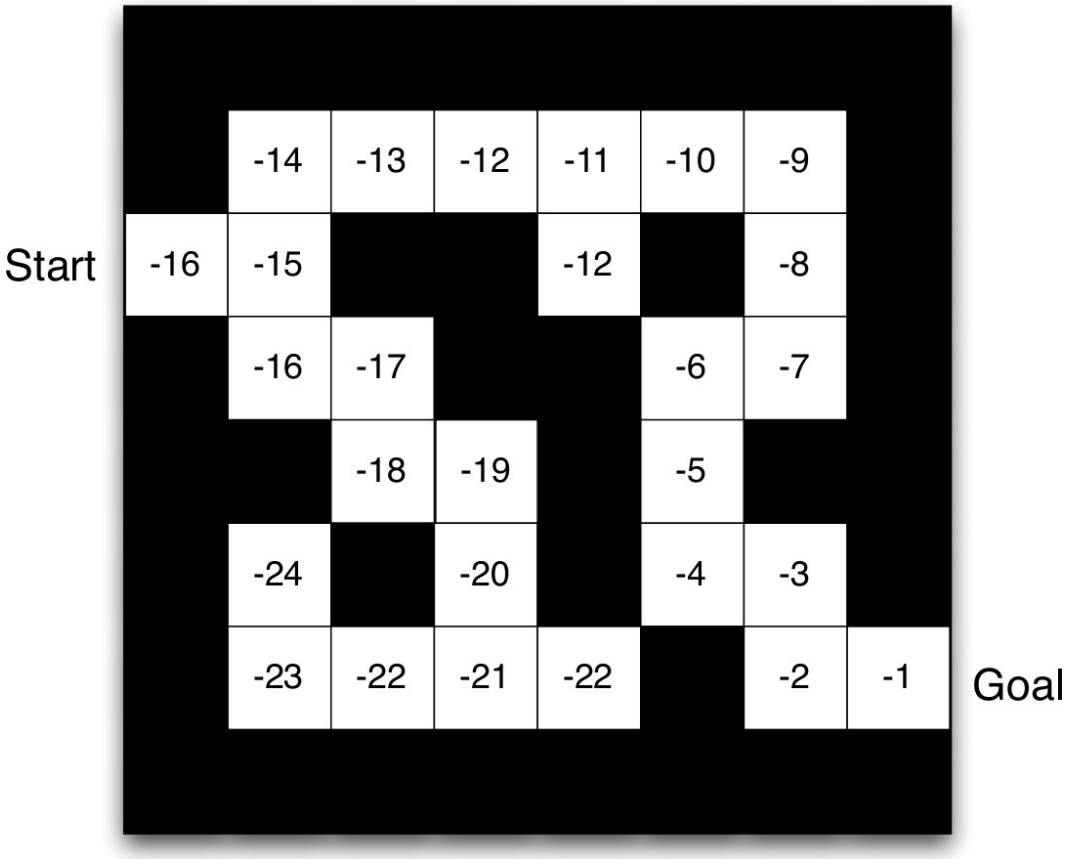
Example: a **policy** for the maze



~~Strategy~~ Policy

Arrows represent the **policy** π : which action to take from every state.

Example: **values** for the policy of the maze



- Let π be the optimal policy
- Value $v_\pi(s)$ for every state s

Exercise

Choose a state s and compute $v_\pi(s)$ by yourself. If s^j denotes the successor state of s, can the value $v_\pi(s^j)$ help with this computation?

1. What is Reinforcement Learning?
2. The RL setup: problem and actors
3. What do we know? State and observability
4. What can we do? Policy and value - and model?
5. The never-ending control loop: prediction = improvement

Prediction, **improvement** and **control**

The **prediction** problem in RL

Forecast the future: can you say from each state how much will be your return? It depends on the policy!

The **improvement** problem in RL

Change the future: can you find a different policy that will give you a better return?

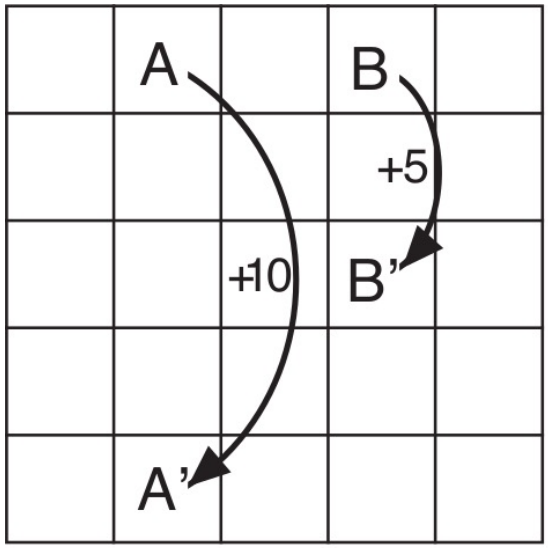
The **control** problem in RL

Change the future: can you find the best policy at all?

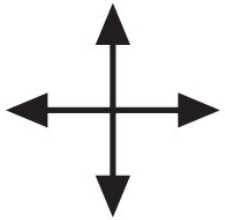
Exercise

State formally the prediction, the improvement and the control problem.

Gridworld example: **prediction**



(a)



Actions

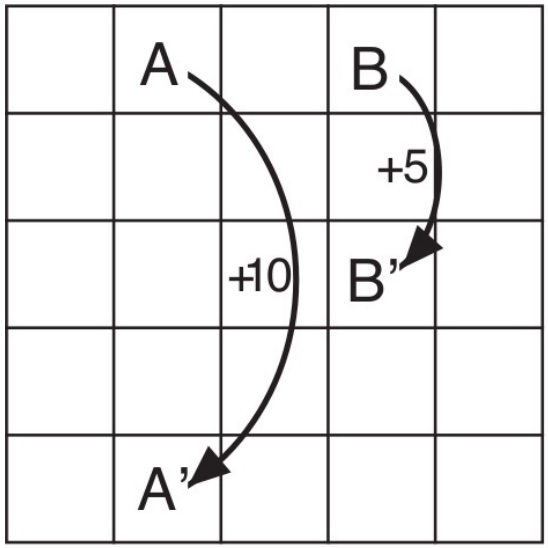
3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

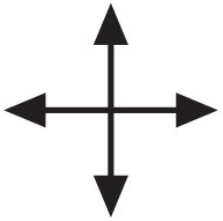
Exercise

Compute the value function for the uniform random policy.

Gridworld example: **improvement**



(a)



Actions

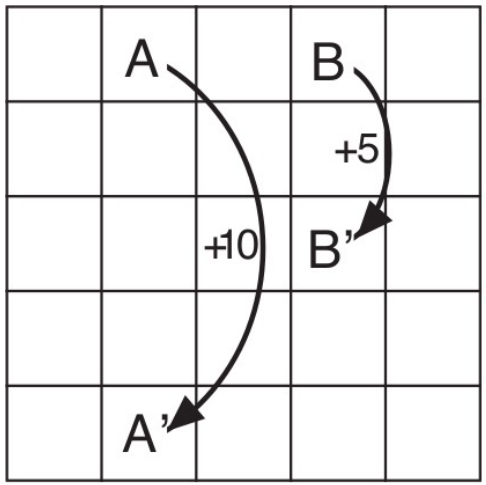
3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

Exercise

Find an improvement of the uniform policy.

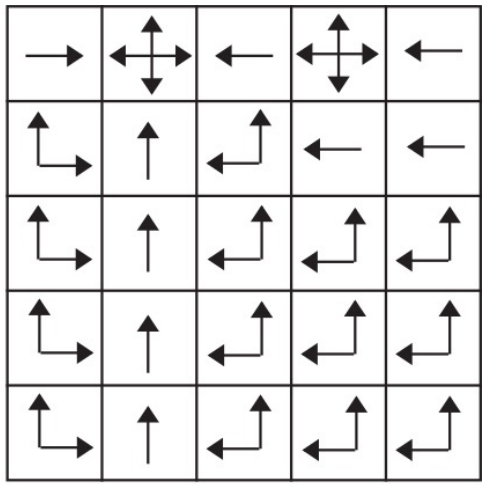
Gridworld example: optimal **control**



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

b) v_*



c) π_*

Exercise

Compute the **optimal value** function over all possible policies.

Given the optimal value v_* as above, find the optimal policy. Is the optimal policy unique?

Wrapping up

Learning goals

- Understand the RL problem, and how RL differs from supervised learning
- Understand reward, return and how they are used to make decisions
- Understand actions, states and rewards in term of agent/environment interactions
- Understand the optimal control problem

Wrapping up

What we (hopefully) have learnt

- Reinforcement Learning (RL) is concerned with goal-directed learning and decision-making. In RL an agent learns from experiences it gains by interacting with the environment. In supervised learning we cannot affect the environment
- In RL rewards are often delayed in time and the agent tries to maximize the cumulative sum of rewards, called *return*. Return is a long-term goal. For example, one may need to make seemingly suboptimal moves to reach a winning position in a game
- An agent interacts with the environment via actions. The environment answers with states and rewards ... and so on in a loop, that can finish after a certain number of steps or go on forever
- Optimal control can be achieved by a prediction-improvement loop

Exercises

- Can every decision task be represented as an optimization problem with respect to a suitable reward?
- Is the reward an intrinsic datum of the decision task?
- Make an example where the greedy policy is optimal.
- Make an example of a decision task with non-Markov environment state.
- Make an example of a decision task with non-Markov agent state.

Exercises

The generic definition of policy is a time-dependant stochastic function of the history: $\pi_t(A_t|H_t) = \Pr(A_t|H_t)$. Give a definition of the policy in the following cases, and find a corresponding task.

- Fully observable environment, stochastic and non-stationary policy.
- Partially observable environment, stochastic and stationary policy.
- Fully observable environment, deterministic and stationary policy.