Reinforcement Learning

Introduction to Reinforcement Learning

Nguyễn Đăng Trị

School of Engineering and Technology
Hue University

Ngày 23 tháng 2 năm 2024

Overview

1. Giới thiệu chung về học phần
2. Nội dung học phần
3. Machine Learning - Reinforcement Learning

Giới thiệu về học phần

- Tên: Học tăng cường - Reinforcement Learning
- Số tín chỉ: 3
- Giáo trình: Richard S. Sutton and Andrew G. Barto (2015). Reinforcement Learning: An Introduction. The MIT Press Cambridge, Massachusetts London, England.
- Giảng viên: Nguyễn Đăng Trị
- Email: tringuyendang@hueuni.edu.vn
- Phone: 0968540108

Điều kiện tiên quyết

- Tiếng Anh
- Python
- Toán
- Tự học và tự nghiên cứu

Phương thức đánh giá kết quả

- Đánh giá quá trình : 10%
- Đánh giá giữa kỳ
  1. Báo cáo/thảo luận: 10%
  2. Thi tự luận/trắc nghiệm: 30%
- Đánh giá cuối kỳ: Thi tự luận/trắc nghiệm: 50%

Cấu trúc học phần

- Chap 1: Giới thiệu về học tăng cường
- Chap 2: Marlove decision processes and planning
- Chap 3: Model-free policy evaluation
- Chap 4: Model-free control
- Chap 5: RL with function approximation and Deep RL
- Chap 6: Policy search
- Chap 7 Exploration and Exploitation
- Advanced topics

Yêu cầu

1. Nghiêm túc thực hiện các bài tập của giảng viên đưa ra
2. Tự học và tự nghiên cứu
3. Trao đổi thẳng thắn và cởi mở
4. Nêu cao tin thần tôn trọng quyền tác giả và tính trung thực

Quick review

Supervised Learning

- **Data**: (x, y) x is data, y is label
- **Goal**: Learn a function to map $x \rightarrow y$
- **Example**: Classification, Regression, object detection, etc.

Unsupervised Learning

- **Data**: x Just data, there is no label
- **Goal**: Learn some underlying hidden structure of the data
- **Example**: Clustering, dimensionality reduction, feature learning, density estimation, etc.

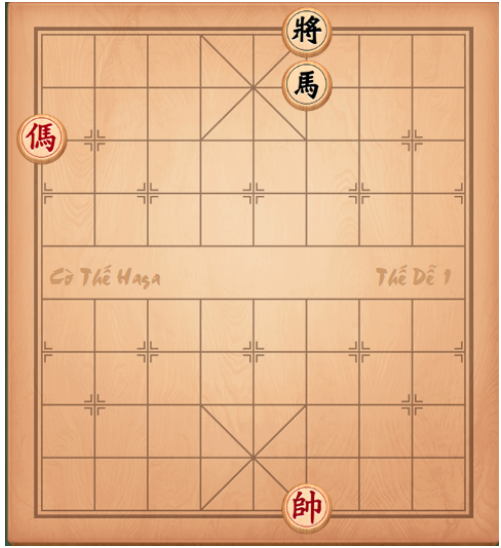
Today's lecture

Reinforcement Learning (RL)

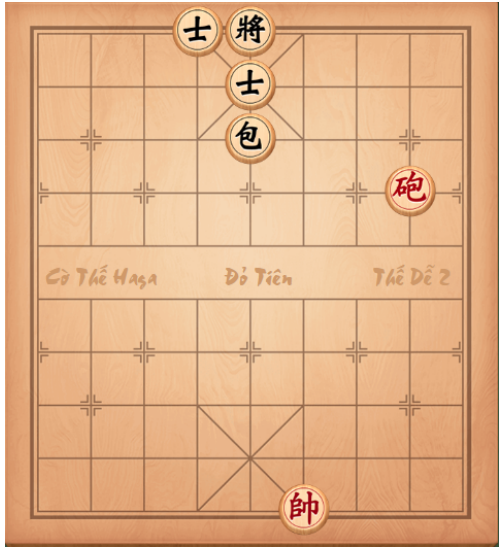
Today's lecture

What is RL?

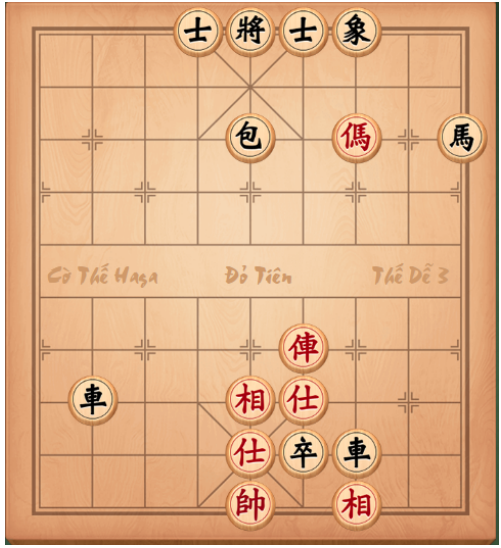
Example



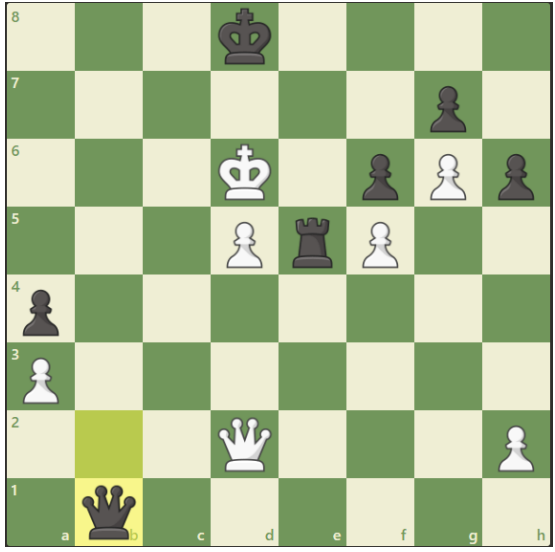
Example



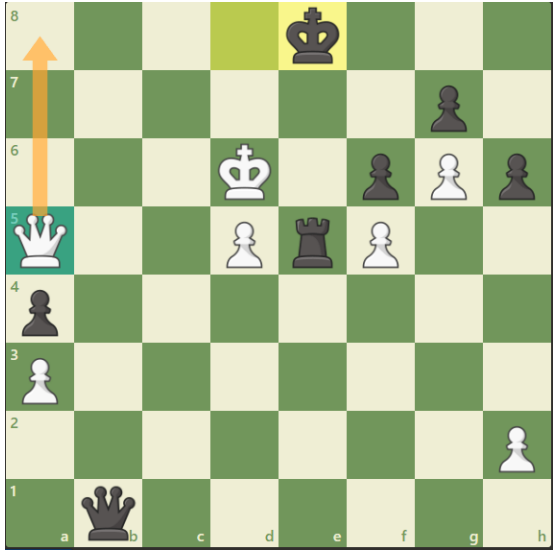
Example



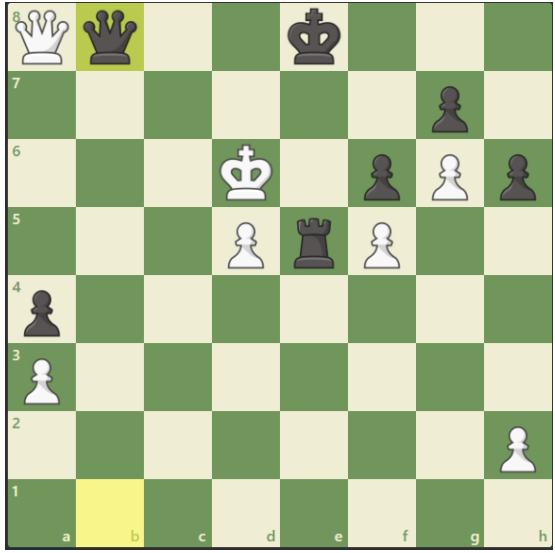
Example



Example

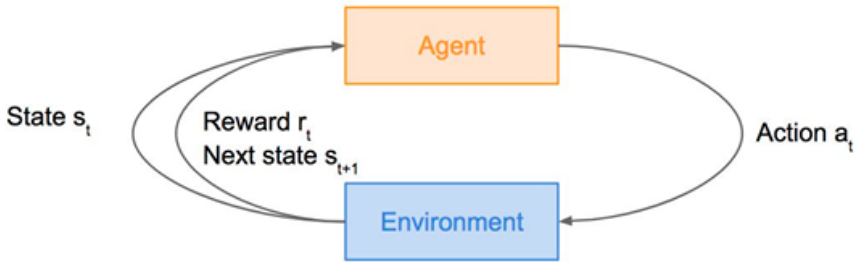


Example



What is RL?

- Problems involving an **agent** interacting with an **environment**, which provides numeric **reward** signals
- Goal: Learn how to make a **policy** in order to make an actions in order that maximizes the **reward**



Một vài thuật ngữ cơ bản trong RL

- Environment (Môi trường) là không gian tương tác
- Agent là chủ thể thực hiện hành động
- Policy là quy luật/chiến thuật
- State mô tả hiện trạng/hình thái/trạng thái của agent/environment
- Reward là phần thưởng tương ứng với các hành động
- Action(s) là những thứ Agent có thể thực hiện

Đặc điểm của RL

Sự khác biệt giữa RL và những loại hình học máy khác là gì?

- There is no supervisor, only a *reward* signal.
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d. data)
- Agent is *active*: its actions affect the environment he lives in.

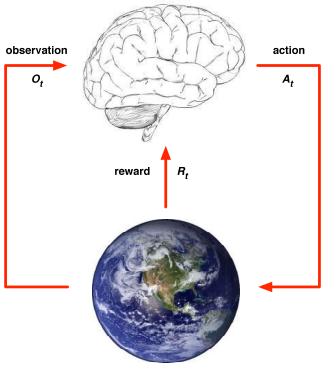
1

[1]Source: Davide Abati, University of Modena and Reggio Emilia

Sequential Decision Making

- Goal: *select actions to maximise total 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.
  - A financial investment may take months to mature
  - Refuelling a helicopter now might prevent a crash in several hours
  - Blocking opponent moves might help winning chances many moves from now

Agent and environment



- At each step t the agent:
  - Receives observation O_t
  - Receives scalar reward R_t
  - Executes action A_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

State

- The **history** is the sequence of observations, actions, rewards

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

- The **state** is the information used to determine what happens next.
  - It is a function of the history:

$$S_t = f(H_t)$$

Agent and environment states

Agent state S_t^a	Environment state S_t^e
whatever information the agent uses to pick the next action	whatever data the environment uses to pick the next observation/reward
it is the information used by RL algorithms	usually not visible by the agent

- **Full observability**: agent directly observes environment state
- **Partial observability**: agent indirectly observes environment state

Inside a reinforcement learning agent

- An agent may include one or more of these components:
  - Policy: agent's behaviour function
  - Value function: how good is each state and/or action
  - Model: representation of the environment

Policy

- A **policy** is the agent's behaviour
- It is a map from state to action
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = s|S_t = s]$

Return

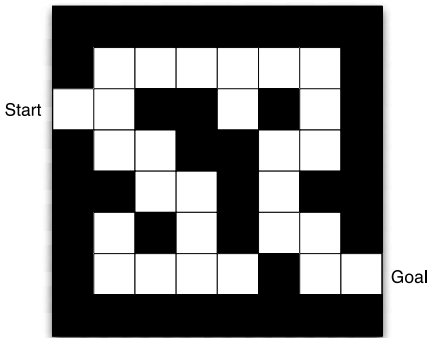
Definition

The return G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

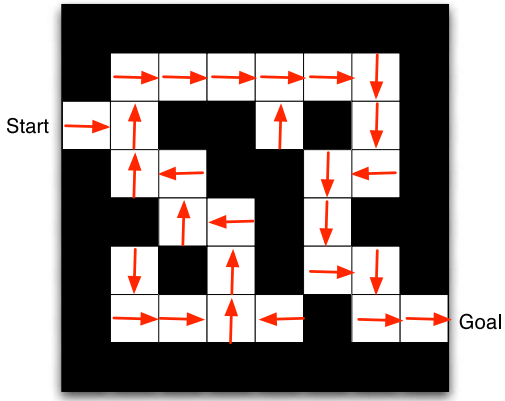
- The discount $\gamma \in [0, 1]$ is the present value of future rewards
- The value of receiving reward R after $k + 1$ time-steps is $\gamma^k R$.
- This values immediate reward above delayed reward.
- γ close to 0 leads to *myopic* evaluation
- γ close to 1 leads to *far-sighted* evaluation

Learn through example: Maze



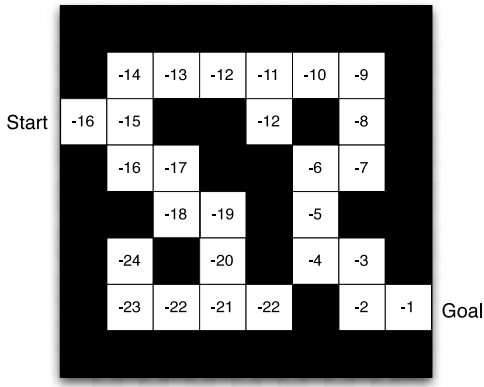
- Rewards: -1 per time-step
- Actions: N, S, W, E
- States: Agent's location

Maze example: policy



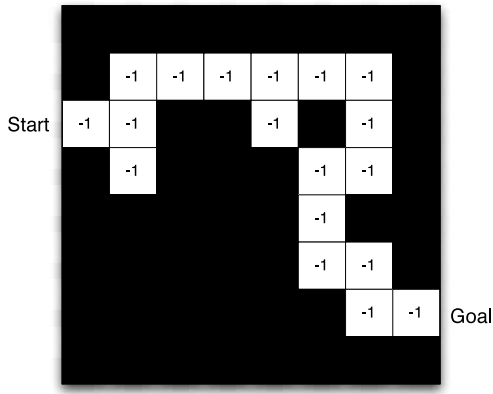
- Arrows represent policy $\pi(s)$ for each state s

Maze example: value function



- Numbers represent policy $v_\pi(s)$ for each state s

Maze example: model



- Grid layout represent transition model $\mathcal{P}^a_{ss'}$
- Numbers represent immediate reward R^a_s from each state s (same for all a)

Q&A

The End