Reinforcement Learning

Introduction

Michael Herrmann, David Abel
Slides by Stefano V. Albrecht

14 January, 2025

Lecture Outline

- Course details and admin
- What is reinforcement learning?
- Examples

Course Team

Course organiser:

- Dr. Michael Herrmann

Co-lecturer:

- Dr. David Abel

TAs:

- Qiyue Xia
- Adam Jelley (tutorials)
- Eric Liu (tutorials)

… and others

Course Details

Course page:

- `https://opencourse.inf.ed.ac.uk/rl`

Announcements:

- via course page ("Announcements"), via Learn and by email to rl-students@inf.ed.ac.uk

Lectures:

- Time: Tuesdays & Fridays, 14.10–15.00
- Place: 40 George Square, Lecture Theatre B
- Lectures will be recorded (see "Lecture Recordings")

The RL Book

First half of course based on:

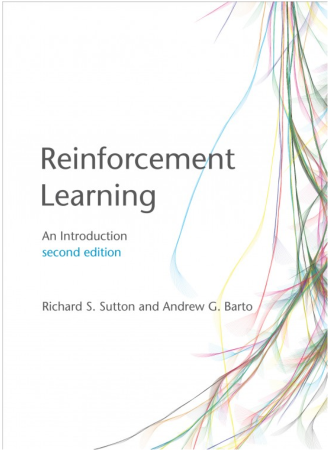
Reinforcement Learning: An Introduction (2nd edition)

Richard Sutton & Andrew Barto

MIT Press (2018)

Download free PDF:

`http://incompleteideas.net/book/the-book-2nd.html` (2022 version)



New: The MARL Book

For the second half of course this book will be useful:

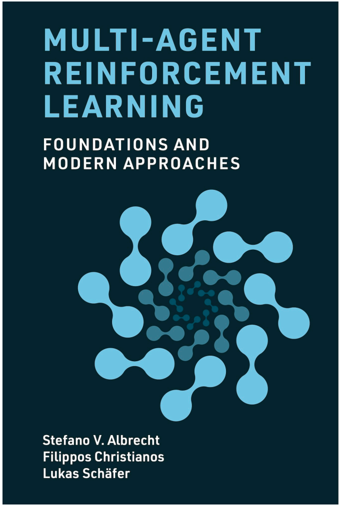
Multi-Agent Reinforcement Learning: Foundations and Modern Approaches

Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer

MIT Press (2024)

Download free PDF:

`https://www.marl-book.com`



Course Topics

- Multi-armed bandits
- Markov decision processes
- Dynamic programming
- Monte Carlo methods
- Temporal-difference learning
- Tutorial lecture: Building a RL system
- Value function approximation
- Policy gradient methods
- Deep reinforcement learning
- Current research in RL

Highly recommended to read the corresponding book chapters

Tutorials

Tutorials:

- Weekly, in weeks 2–10 (mostly on Wednesday)
- Not graded, but attendance will be monitored
- Tutorial sheets released Tuesday noon of previous week (on course page)
- Solutions released in following week

  $\Rightarrow$ See "Tutorials" page on course page for more details

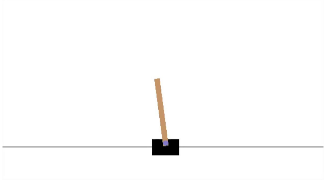
Assignment to tutorial slots is done automatically by ITO

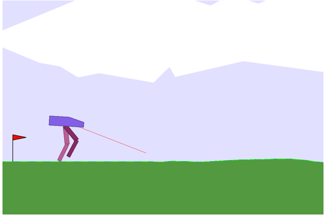
$\Rightarrow$ Use the form at Timetabling (Registry Services), if you need to change your slot

Coursework

Coursework — 50% of final grade

- Implement and test RL algorithms in Python
- Out: 11 Feb / Due: 28 March (noon)
- Lab sessions in weeks 6–9
- Coursework will be introduced in lecture on 11 Feb





Exam

Exam — 50% of final grade

- Testing theoretical and applied knowledge
- Any material covered in *required readings* <u>and</u> *associated lectures* is examinable (excluding exercises and examples in RL book)
- Exams from previous years: `https://exampapers.ed.ac.uk`

Discussion Forum and Office hour

We use **Piazza**:

- Forum to post and discuss questions with peers
- Link to Piazza forum on course page
- Your fellow students as well as TAs and lecturers will answer questions

  $\Rightarrow$ First check whether your question has been answered, then post

  $\Rightarrow$ Use the folders to organise posts (makes it easier for people to find questions)

  $\Rightarrow$ Explain your thinking and where you are “stuck”

Office hour: Wednesdays, 4:10pm - 5pm, IF 1.42 (Michael H)

- Any issues where Piazza or tutorials are not suitable.

Course Pre-requisites

Maths:

- Basic statistics and probability theory
- Linear algebra and calculus (vectors, derivatives, limit analysis)
- See also last year's exam for maths requirements

Programming:

- Advanced programming for coursework (we use Python)

  $\Rightarrow$ **Course is <u>not</u> an introduction to programming!**
- Use our Coding Proficiency Self-Check PDF on OpenCourse page

Feedback from the 2023/24 season

- “The lecture content was great.”
- “Lecture was quite difficult to keep up with and felt rushed.”
- “Slow down the lectures slightly to allow more time for questions.”
- “I found the drawings in the tutorials (frog, monkey etc) confusingly weird and unclear (maybe its an artist view).”
- “In the labs, probably a 10 min introduction on what we’re going to learn today.”
- Coursework-related feedback will be discussed with CW release

More feedback from last year: Advice for this year's students

- "Read RL book to be in the loop with the lectures."
- "Go to tutorial. I found lecture extremely hard to keep up with, and the tutorial is the only way I managed to hold on."
- "Must attend tutorials."
- "You must know Python Fundamentals and have a decent mathematics background."
- "Also, be sure to practice pytorch before starting the coursework, as a lot of the CW relies on it"
- "Start coursework as soon as it is released."
- "For the coursework: Try to work on it weekly and visit each drop-in lab to ask questions."

Reading Group

Reading group meetings to discuss recent research papers

- Open to all students, but basic RL knowledge assumed
- Read paper before meeting to participate actively in discussion
- Sign up here:

`https://agents.inf.ed.ac.uk/reading-group/`

Questions about the course?

What is Reinforcement Learning?

Reinforcement learning (RL):

Learning to solve sequential decision problems via **repeated interaction with environment**

What is Reinforcement Learning?

Reinforcement learning (RL):

Learning to solve sequential decision problems via repeated interaction with environment

- What is a sequential decision problem?

What is Reinforcement Learning?

Reinforcement learning (RL):

Learning to solve sequential decision problems via repeated interaction with environment

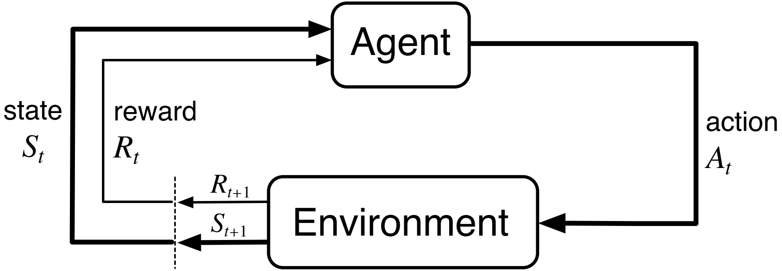
- What is a sequential decision problem?
- What does it mean to “solve” the problem?

What is Reinforcement Learning?

Reinforcement learning (RL):

Learning to solve sequential decision problems via **repeated interaction with environment**

- What is a sequential decision problem?
- What does it mean to “solve” the problem?
- What is learning by interaction?

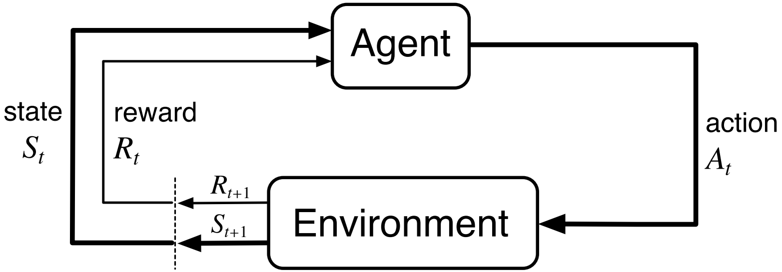

Agent
state
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment

What is Reinforcement Learning?

Agent takes actions in environment

- Take action, observe new state and reward from environment
- Goal is to maximise total rewards received

  $\Rightarrow$ Learning: find best actions by *trying* them



What is Reinforcement Learning?

Example: Chickens finding food

Video: chicken

Example: Chickens finding food

- Agent: chicken

Video: chicken

Example: Chickens finding food

- Agent: chicken
- Environment: flat table with coloured circles

Video: chicken

Example: Chickens finding food

- Agent: chicken
- Environment: flat table with coloured circles
- Actions: motor control of legs, beak.

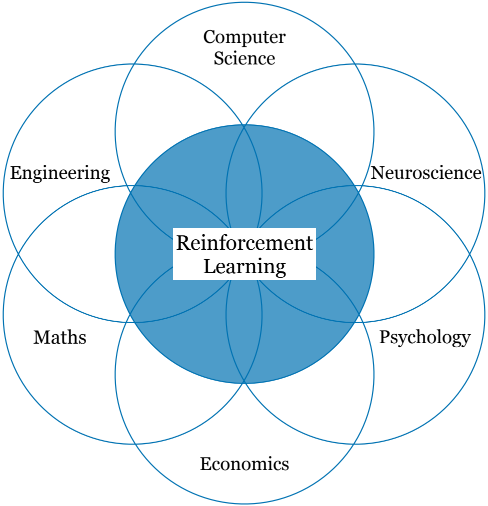
Video: chicken

Example: Chickens finding food

- Agent: chicken
- Environment: flat table with coloured circles
- Actions: motor control of legs, beak.
- Reward: curiosity, food.

Video: chicken

The Many Disciplines of RL



Thanks to Dave Silver for the inspiration for the diagram

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Examples:

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Examples:

- Reach target state s^*: reward is 1 if $S_t = s^*$, else 0 (or -1? what's the difference?)

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Examples:

- Reach target state s^*: reward is 1 if $S_t = s^*$, else 0 (or -1? what's the difference?)
- Win Chess game: reward is $+1$ if won, -1 if lost, 0 otherwise

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Examples:

- Reach target state s^*: reward is 1 if $S_t = s^*$, else 0 (or -1? what's the difference?)
- Win Chess game: reward is $+1$ if won, -1 if lost, 0 otherwise
- Manage investment portfolio: reward?

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Examples:

- Reach target state $s*$: reward is 1 if $S_t = s*$, else 0 (or -1? what's the difference?)
- Win Chess game: reward is $+1$ if won, -1 if lost, 0 otherwise
- Manage investment portfolio: reward?
- Make humanoid robot walk: reward?

Reward Hypothesis

Reward hypothesis:

All goals can be described by the maximisation of the expected value of cumulative scalar rewards.

Discussion Question (2 minutes): Is this true? False? What are some other goals or tasks that can be described through maximization of scalar values?

See Tutorial 1

RL is a type of **machine learning** (ML):

- Learning = improving performance with experience (data)

Machine Learning

See Tutorial 1

RL is a type of **machine learning** (ML):

- Learning = improving performance with experience (data)

Supervised learning:

- Discover unknown function $f(x) = y$ given examples $(x, y = f(x))$

  $\Rightarrow$ RL not supervised: correct actions are not provided

See Tutorial 1

RL is a type of **machine learning** (ML):

- Learning = improving performance with experience (data)

Supervised learning:

- Discover unknown function $f(x) = y$ given examples $(x, y = f(x))$

  $\Rightarrow$ RL not supervised: correct actions are not provided

Unsupervised learning:

- Discover hidden structure in data $x_1, x_2, x_3, \ldots$ (no y given)

  $\Rightarrow$ RL not unsupervised: reward signal informs correct action

See Tutorial 1

RL is a type of **machine learning** (ML):

- Learning = improving performance with experience (data)

Supervised learning:

- Discover unknown function $f(x) = y$ given examples $(x, y = f(x))$

  $\Rightarrow$ RL not supervised: correct actions are not provided

Unsupervised learning:

- Discover hidden structure in data $x_1, x_2, x_3, \ldots$ (no y given)

  $\Rightarrow$ RL not unsupervised: reward signal informs correct action

Reinforcement learning is third category of ML: learning to act to maximise rewards

Reinforcement Learning Challenges

Key challenges in RL

- Unknown environment:
  How do actions affect environment state and rewards?

Reinforcement Learning Challenges

Key challenges in RL

- Unknown environment:
  How do actions affect environment state and rewards?

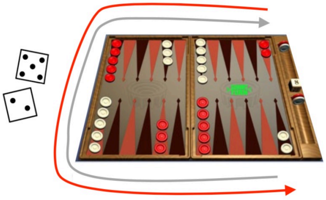
- Exloration-exploitation dilemma:
  When to try new actions (*explore*)?
  When to stick with what we think is best (*exploit*)?

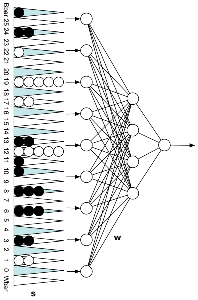
Key challenges in RL

- Unknown environment:
  How do actions affect environment state and rewards?
- Exloration-exploitation dilemma:
  When to try new actions (*explore*)?
  When to stick with what we think is best (*exploit*)?
- Delayed rewards:
  Actions may have long-term consequences and affect future rewards
  When we get reward, which prior actions led to it? (*credit assignment*)

Example: Backgammon

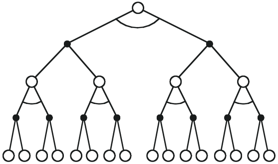
Learning to play Backgammon (Tesauro, 1992-1995)





estimated state value
(≈ prob of winning)

Action selection
by a shallow search



Start with a random Network

Play millions of games against itself

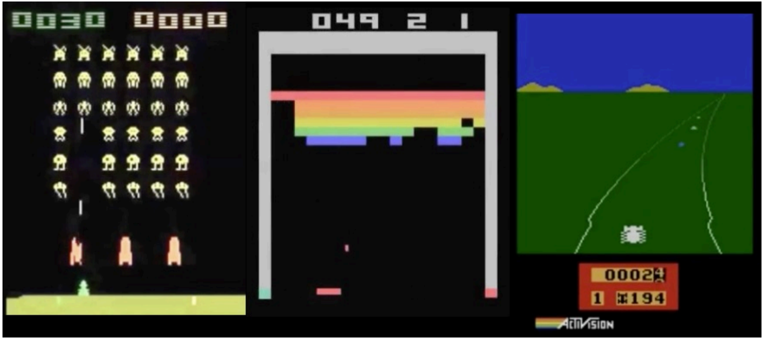
Learn a value function from this simulated experience

Six weeks later it's the best player of backgammon in the world

Originally used expert handcrafted features, later repeated with raw board positions

Example: Atari

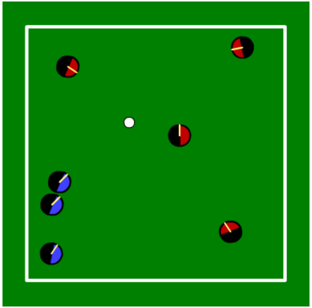
Learning to play Atari games (Mnih et al., 2013, 2015)



Video: DQN in Atari games

Example: Soccer

Learning to keep the ball in team (Stone et al., 2005)



Video: keepaway soccer

Source: `http://www.cs.utexas.edu/~AustinVilla/sim/keepaway`

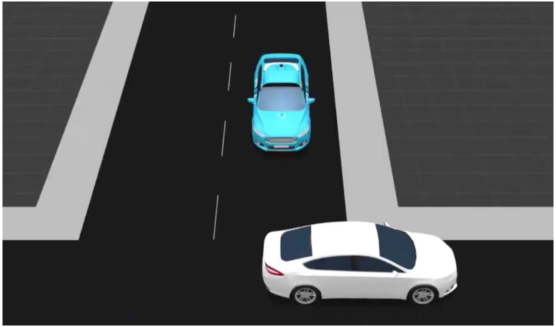
Learning to walk and jump (DeepMind, 2017)

Video: learning to walk

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

Example: Autonomous Driving

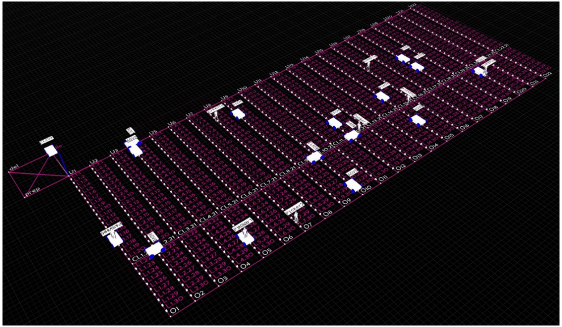
IGP2 autonomous driving system (Five AI, 2021)



Video: IGP2 autonomous driving system
Source: `https://www.five.ai/igp2`

Example: Warehouse Control

Mobile robots and humans managing a warehouse (Dematic/KION, 2022)



Video: Multi-robot warehouse

Source: `https://sites.google.com/view/scalablemarlwarehouse`

Reinforcement Learning: The Big Picture

Learning how to act

Reading

Required:

- RL book, Chapter 1 (1.1–1.4)

Optional (for keen students):

- Silver et al.: “Reward is enough”. Artificial Intelligence (2021) `https://doi.org/10.1016/j.artint.2021.103535`
- List of survey papers for RL: `https://agents.inf.ed.ac.uk/blog/reinforcement-learning-surveys/`
- Past MSc dissertations in RL: `https://agents.inf.ed.ac.uk/blog/master-dissertations/`