Reinforcement Learning

Introduction

Stefano V. Albrecht, Michael Herrmann

16 January 2024

THE UNIVERSITY of EDINBURGH
informatics

Lecture Outline

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

Course Team

Course organiser:

- Dr. Stefano V. Albrecht
- Dr. Michael Herrmann

TAs:

- Mhairi Dunion
- Trevor McInroe
- Adam Jelley (tutorials)
- Eric Liu (tutorials)

Course Details

Course page:

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

Announcements:

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

Lectures:

- Time: Tuesdays & Fridays, 14.10–15.00
- Place: Appleton Tower Lecture Theatre 1
- Lectures will be recorded (see "Lecture Recordings")

The RL Book

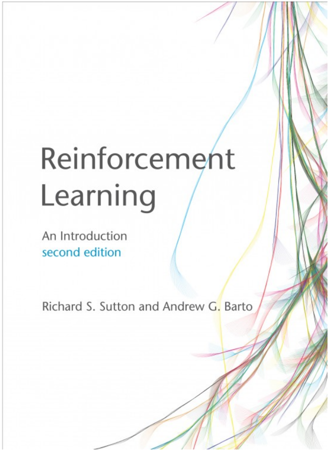
Course book:

Reinforcement Learning: An Introduction (2nd edition)

by Richard Sutton & Andrew Barto

Download free PDF:

`http://incompleteideas.net/book/the-book-2nd.html`



New: The MARL Book

New book published by MIT Press (2024):

Multi-Agent Reinforcement Learning: Foundations and Modern Approaches

Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer

Download free PDF:

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

Multi-Agent Reinforcement Learning: Foundations and Modern Approaches

Stefano V. Albrecht, Filippos Christianos, Lukas Schäfer

To be published by MIT Press
(print version scheduled for fall 2024)

Course Topics

- Multi-armed bandits*
- Markov decision processes*
- Dynamic programming*
- Monte Carlo methods*
- Temporal-difference learning*
- Planning*
- Tutorial lecture: building a RL system
- Value function approximation*
- Policy gradient methods*
- Deep reinforcement learning
- Multi-agent reinforcement learning

**Examined - based on chapter in RL book*

Highly recommended to read chapter/slides before lecture!

Notation

A note on notation:

- RL book uses notation S_t, A_t, R_{t+1} (reward received at $t + 1$), $p(s', r|s, a)$

  We will use this notation for lectures that are based on the RL book

- Other notation also widely used (e.g. in MARL book)
  e.g. s^t, a^t, r^t, $T(s, a, s')$, $R(s, a)$

Tutorials

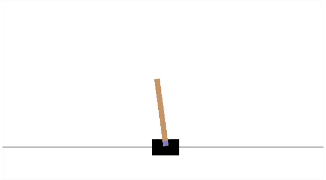
Tutorials:

- Weekly, in weeks 2–10
- Optional attendance – not graded
- 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

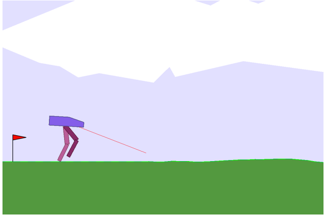
Assignment to tutorial slots is done automatically by ITO

$\Rightarrow$ Contact ITO if you need to change your slot



Coursework — 50% of final grade

- Implement and test RL algorithms in Python
- Out: 13 Feb / Due: 31 March
- Lab sessions in weeks 5–8
- Coursework will be introduced in lecture on 13 Feb



Exam

Exam — 50% of final grade

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

Discussion Forum

We use **Piazza**:

- Forum to post and discuss questions with peers
- Link to Piazza forum on course page
- 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”

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 course page

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

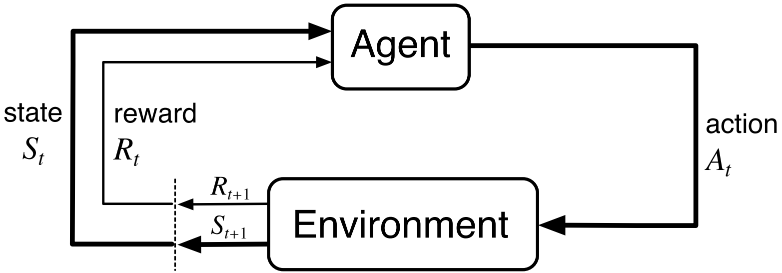
- What is a sequential decision problem?
- What does it mean to “solve” the problem?
- What is learning by interaction?

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



Example: human infant learning

- Agent: baby
- Environment: physical workspace with coloured rings and stacking pole
- Actions: motor control of arms, legs, ...
- Reward: curiosity, satisfaction upon completion (rings stacked)

Agent does not know what actions to take

$\Rightarrow$ Must *discover*!

Video: ring stacker

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?

Machine Learning

See Tutorial 1

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

- Learning = improving performance with experience (data)

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 $x_1, x_2, x_3, \dots$ (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*)?

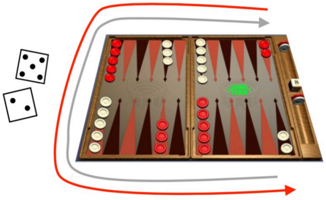
Reinforcement Learning Challenges

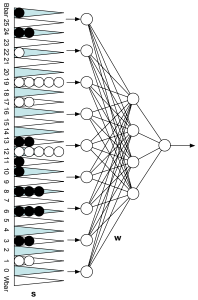
Key challenges in RL

- 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*)?
- 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

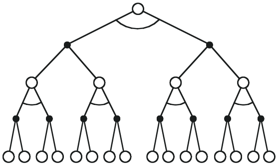
Learning to play Backgammon (Tesauro, 1992-1995)





estimated state value
(≈ prob of winning)

Action selection
by a shallow search



Start with a random Network

Play millions of games against itself

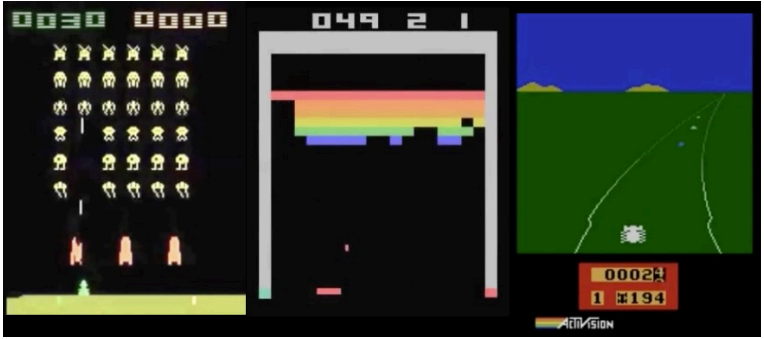
Learn a value function from this simulated experience

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

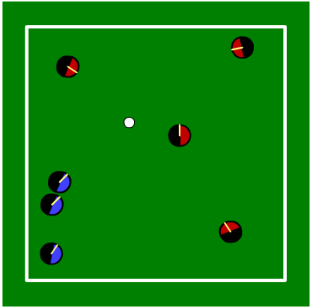
Learning to play Atari games (Mnih et al., 2013, 2015)



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`

Example: Walking

Learning to walk and jump (DeepMind, 2017)

Video: learning to walk

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

Example: Starcraft II

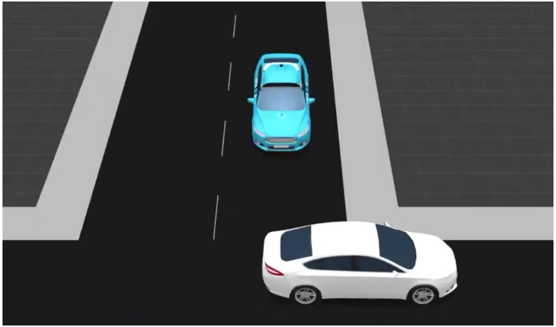
Starcraft Multi-Agent Challenge (Samvelyan et al., 2019)

Video: SMAC

EPyMARL codebase: `https://github.com/uoe-agents/epymarl`

Example: Autonomous Driving

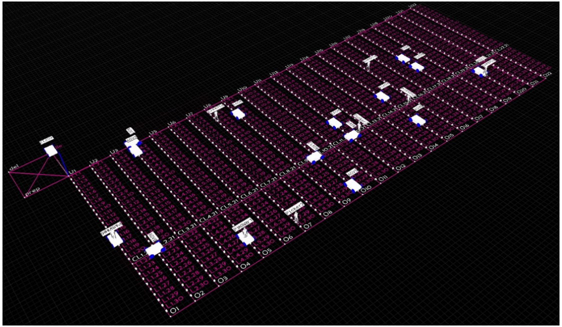
IGP2 autonomous driving system (Five AI, 2021)



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`

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/`