# Reinforcement Learning

## Introduction

Stefano Albrecht, Pavlos Andreadis

14 January 2020

# Lecture Outline

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

# Course Details

**Lecturers:**

- Dr. Stefano Albrecht, `s.albrecht@ed.ac.uk`
- Dr. Pavlos Andreadis, `pavlos.andreadis@ed.ac.uk`

**TAs:**

- Arrasy Rahman, `arrasy.rahman@ed.ac.uk`
- Filippos Christianos, `f.christianos@ed.ac.uk`
- Lukas Schäfer, `l.schaefer@ed.ac.uk`

**Lectures:**

- Tuesdays & Fridays, 14.10–15.00, Lecture Theatre 5, Appleton Tower

# Course Details

- **Course page:** `http://learn.ed.ac.uk` → search “Reinforcement Learning”
- **Announcements:** via mailing list `rl-students@inf.ed.ac.uk`
  ⇒ Check spam filter
- **Piazza forum:** online forum to post and discuss questions (peer-support)
  ⇒ Link to forum is on course page
- **Lecture video recording:** available on “Media Hopper Replay” (on course page)

# The RL Book

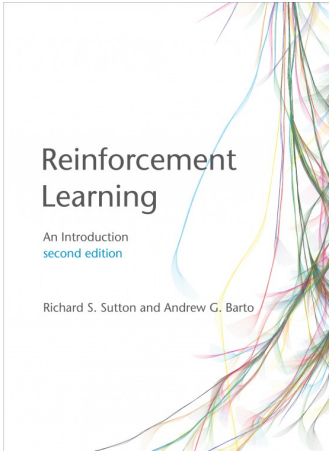
Course book:

**Reinforcement Learning: An Introduction (2nd edition)**

by Richard Sutton & Andrew Barto

Download free PDF:

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



# Course Content

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

**Examined - based on chapter in RL book*

Highly recommended to read chapter/slides before lecture!

# Notation

A note on notation:

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

  We will stick to this notation for lectures based on the book

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

# Tutorials

**Tutorials:**

- Bi-weekly, in weeks 2, 4, 6, 8, 10
- Optional attendance – not graded
- Tutorial sheets released Tuesday noon of previous week (on course page)
- Solutions released in following week

Assignment to tutorial slots is done automatically by ITO

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

# Assessment

**Coursework:**

- Implement and test various RL algorithms in Python
- 30% of final grade
- Out: mid-Feb / Due: end of March
- Coursework introduction in extra lecture

**Exam:**

- Testing theoretical and applied knowledge
- 70% of final grade
- Any material covered in *required readings* and *associated lectures* is examinable

# Course Prerequisites

**Prerequisites:**

- Basic statistics and probability theory
- Linear algebra and calculus (vectors, derivatives, limit analysis)
- Programming skills for coursework (in Python)

  $\Rightarrow$ Course is not an introduction to programming!

See also last year's exam for maths requirements (course page)

# Reading Group

**Reading group** meetings to discuss recent research papers

- Open to all students, but basic RL knowledge assumed
- Expectation is that people will read paper before meeting
- To get updates about future meetings:

  e-mail *"subscribe"* to `rl-reading-group-request@inf.ed.ac.uk`

Questions about the course?

# What is Reinforcement Learning?

**Reinforcement learning (RL):**

Learning to solve sequential decision problems via repeated interaction with environment (trial and error)

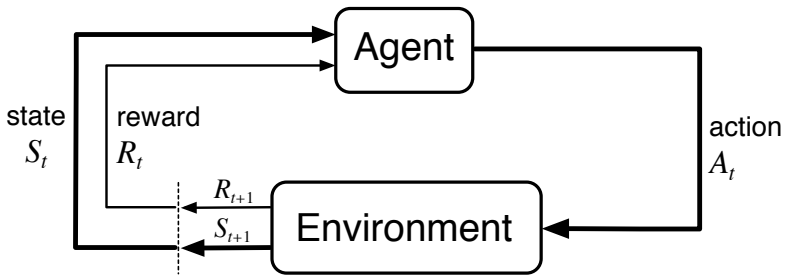
- What is a sequential decision problem?
- What does it mean to “solve” the problem?
- What is learning by interaction?

# 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



# What is Reinforcement Learning?

**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)

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:

- 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

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

- Learning = improving performance with experience (data)

# Machine Learning

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

- Learning = improving performance with experience (data)

**Supervised learning:**

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

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

# Machine Learning

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

- Learning = improving performance with experience (data)

**Supervised learning:**

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

  $\Rightarrow$ 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

# Machine Learning

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

- Learning = improving performance with experience (data)

**Supervised learning:**

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

  $\Rightarrow$ 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*)?

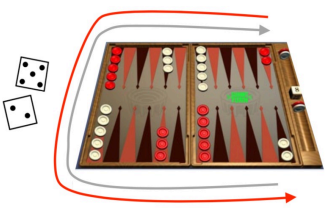
# Reinforcement Learning Challenges

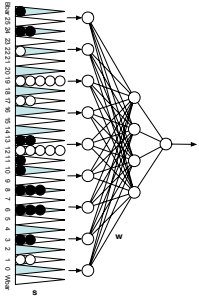
## Key challenges in RL

- 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*)

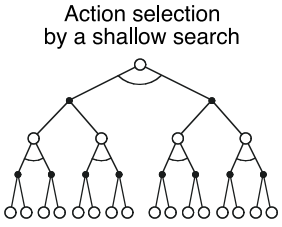
# Examples

## Learning to play Backgammon (Tesauro, 1992-1995)





estimated state value
(≈ prob of winning)




Start with a random Network

Play millions of games against itself

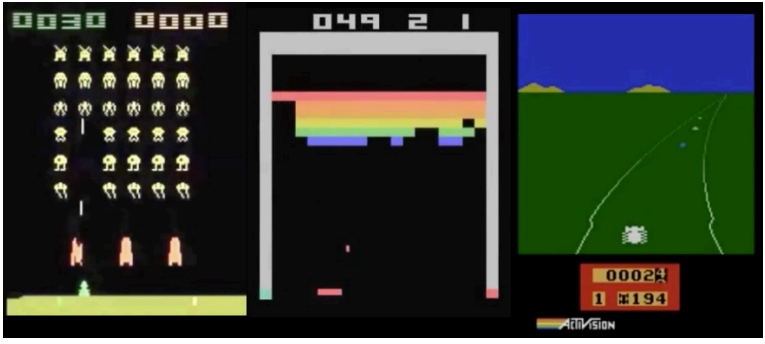
Learn a value function from this simulated experience

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

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

# Examples

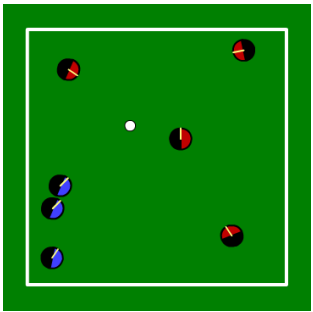
Learning to play Atari games (Mnih, 2013, 2015)




**Video:** DQN in Atari games

# Examples

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



**Video:** 4v3 keepaway soccer

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

# Examples

Learning to walk and jump (DeepMind, 2017)


**Video:** learning to walk

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

# Reading

Required:

- RL book, Chapter 1 (1.1–1.6)

Optional:

- *Artificial Intelligence: A Modern Approach*
  by Stuart Russell and Peter Norvig
  Search on Google...
- *The Quest for Artificial Intelligence*
  by Nils J. Nilson
  Free download: `https://ai.stanford.edu/~nilsson/QAI/qai.pdf`