



IERG 5350

Reinforcement Learning

Lecture 1: Course Overview

Bolei Zhou

The Chinese University of Hong Kong

Outline

- Course logistics
- RL overview and examples

Course Logistics

- Instructor: Bolei Zhou

- TAs:



Zhenghao Peng



Sun Hao



Xiaohang Zhan

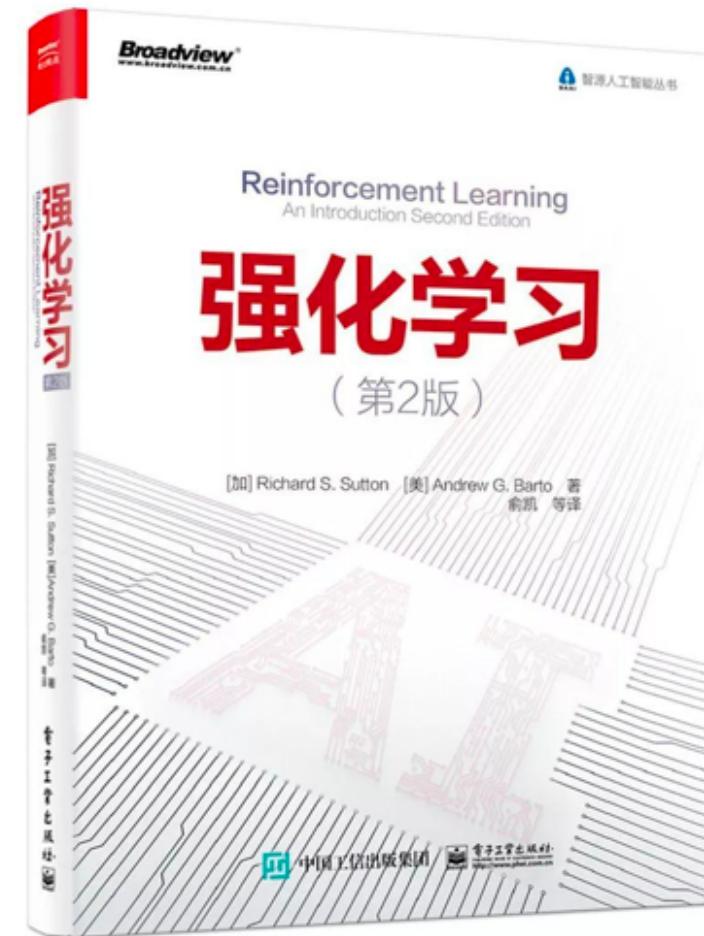
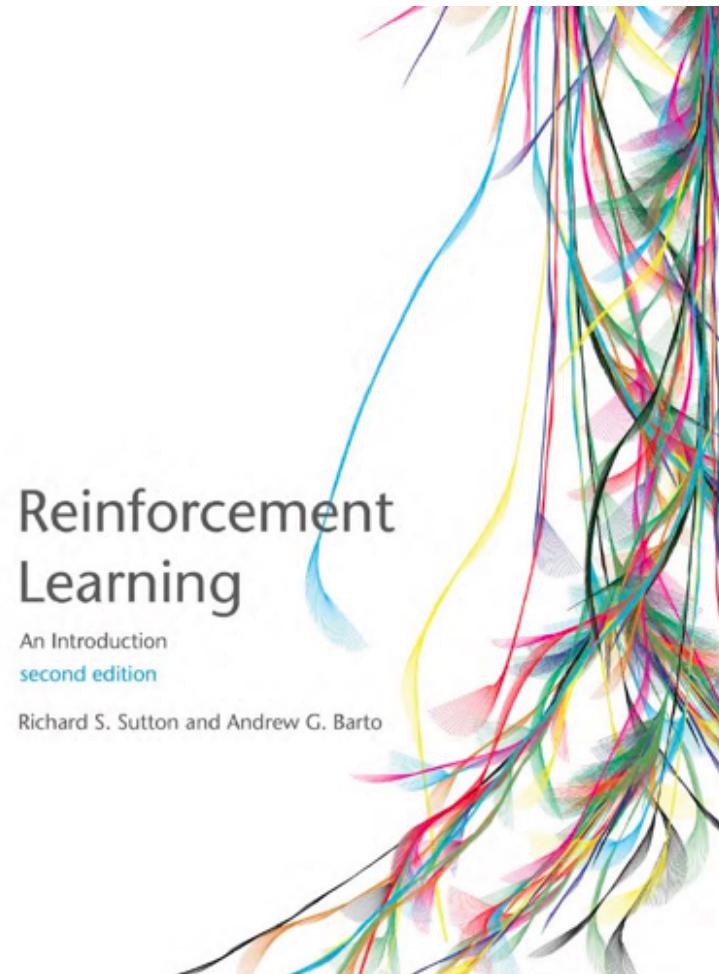
- Time:

- Monday 10:30 am – 11:15 am : one 45-min session
- Tuesday 10:30 am – 12:15 pm: two 45-min sessions
- Course website: <https://cuhkrlcourse.github.io/>
- Piazza: <http://piazza.com/cuhk.edu.hk/fall2020/ierg5350>
- ZOOM virtual: link sent, please keep confidential

- Office hour: Monday 5:00 pm – 6:00 pm, ZOOM virtual

- TA office hour: Tuesday 5:00 pm – 6:00 pm, ZOOM virtual

Textbook: Sutton and Barto:
<http://incompleteideas.net/book/the-book-2nd.html>



Prerequisites & Enrollment

- This is IERG-5XXX course, which means it is a graduate level course
- All enrolled students must have taken the Linear Algebra course and Probability course, and one machine learning relevant course (data mining, pattern recognition, deep learning, etc).
- Coding experience with python

What we will cover

- Key elements of RL: state, reward, action, Markov decision process, exploration and exploitation, etc.
- RL algorithms: Q-learning, policy gradients, actor-critic.
- Others advanced topics: inverse RL, imitation learning, game AI.
- Case studies: Projects in DeepMind, projects at OpenAI, and others.
- Two guest lectures:
 - Researcher from Didi Research on RL in transportation
 - Researcher from Facebook AI Research on RL in game AI

Course Objective

- Know the difference between reinforcement learning, machine learning, and deep learning.
- Knowledge on the foundation and practice of RL
- Given your research problem (e.g. from computer vision, NLP, IoT, etc) decide if it should be formulated as a RL problem, if yes be able to define it formally (in terms of the state space, action space, dynamics and reward model), state what RL algorithm is best for addressing it, and make it work!

Grading

- Attendance: 10%
- 5 assignments: 50%
- Course Project : 40%

Course project

- Relevant to RL or the application of RL
- Suggested projects will be posted in the coming week, but feel free to work on your own project relevant to RL.
- Expected workload of the project: 7 weeks.
- Course projects from last year:
<https://cuhkrlcourse.github.io/2019spring/project.html>
- Deliverables:
 - Proposal due (by the end of Week 3)
 - Mid-term presentation (Week 8): 3-min video recording
 - A github repo containing your work (commits, readme, human-understandable code)
 - Final presentation (Week 14): 5-min video recording and peer-review
 - Course report (in NIPS LaTex Template), due by the end of semester

Course Schedule:

<https://cuhkrlcourse.github.io/schedule.html>

Part 1: Tabular methods,
RL foundation

Part 2: Approximate methods

Part 3: Policy optimization

Part 4: Advanced topics

Week	Time	Topic	Materials
Week 1	Mon	Course overview	
	Tue	Introduction of RL and coding examples	
Week 2	Mon	Markov decision process	
	Tue	Policy iteration and value iteration	HW1 out
Week 3	Mon	Model-free prediction	
	Tue	Model-free control	
Week 4	Mon	On-policy learning and off-policy learning	
	Tue	Connection to optimal control	HW1 due, HW2 out
Week 5	Mon	Value function approximation	
	Tue	Deep Q Learning	
Week 6	Mon	Policy optimization I	
	Tue	Policy optimization II	HW2 due, HW3 out
Week 7	-	Chung Yeung Holiday	
	Tue	Policy optimization III: variants of actor-critic and code	
Week 8	Mon	Policy optimization IV: state of the arts 1	
	Tue	Policy optimization IV: state of the arts 2	HW3 due, HW4 out
Week 9	Mon	Imitation learning	
	Tue	Student project mid-term review	
Week 10	Mon	Guest Lecture by Tony Qin (Didi Research): RL in transportation	
	Tue	Model-based Reinforcement Learning	HW4 due, HW5 out
Week 11	Mon	Exploration and exploitation	
	Tue	Distributed computing and RL system design	
Week 12	Mon	Guest Lecture by Yuandong Tian (Facebook AI Research): Game AI	
	Tue	Inverse RL and Realworld RL	HW5 due
Week 13	Mon	Course Summary	
	Tue	Course Summary	
Week 14	Mon	Student project final presentation	
	Tue	Student project final presentation	

5 Assignments: all programming driven

Week	Time	Topic	Materials
Week 1	Mon	Course overview	
	Tue	Introduction of RL and coding examples	
Week 2	Mon	Markov decision process	
	Tue	Policy iteration and value iteration	HW1 out
Week 3	Mon	Model-free prediction	
	Tue	Model-free control	
Week 4	Mon	On-policy learning and off-policy learning	
	Tue	Connection to optimal control	HW1 due, HW2 out
Week 5	Mon	Value function approximation	
	Tue	Deep Q Learning	
Week 6	Mon	Policy optimization I	
	Tue	Policy optimization II	HW2 due, HW3 out
Week 7	-	Chung Yeung Holiday	
	Tue	Policy optimization III: variants of actor-critic and code	
Week 8	Mon	Policy optimization IV: state of the arts 1	
	Tue	Policy optimization IV: state of the arts 2	HW3 due, HW4 out
Week 9	Mon	Imitation learning	
	Tue	Student project mid-term review	
Week 10	Mon	Guest Lecture by Tony Qin (Didi Research): RL in transportation	
	Tue	Model-based Reinforcement Learning	HW4 due, HW5 out
Week 11	Mon	Exploration and exploitation	
	Tue	Distributed computing and RL system design	
Week 12	Mon	Guest Lecture by Yuandong Tian (Facebook AI Research): Game AI	
	Tue	Inverse RL and Realworld RL	HW5 due
Week 13	Mon	Course Summary	
	Tue	Course Summary	
Week 14	Mon	Student project final presentation	
	Tue	Student project final presentation	

HW1: MDP, Policy and value iterations

HW2: on-policy and off-policy learning in tabular setting

HW3: policy gradient methods

HW4: SOTA methods, PPO, TD3, SAC

HW5: imitation learning/offline RL
open-ended competition

Open-end Competition in Assignment 5



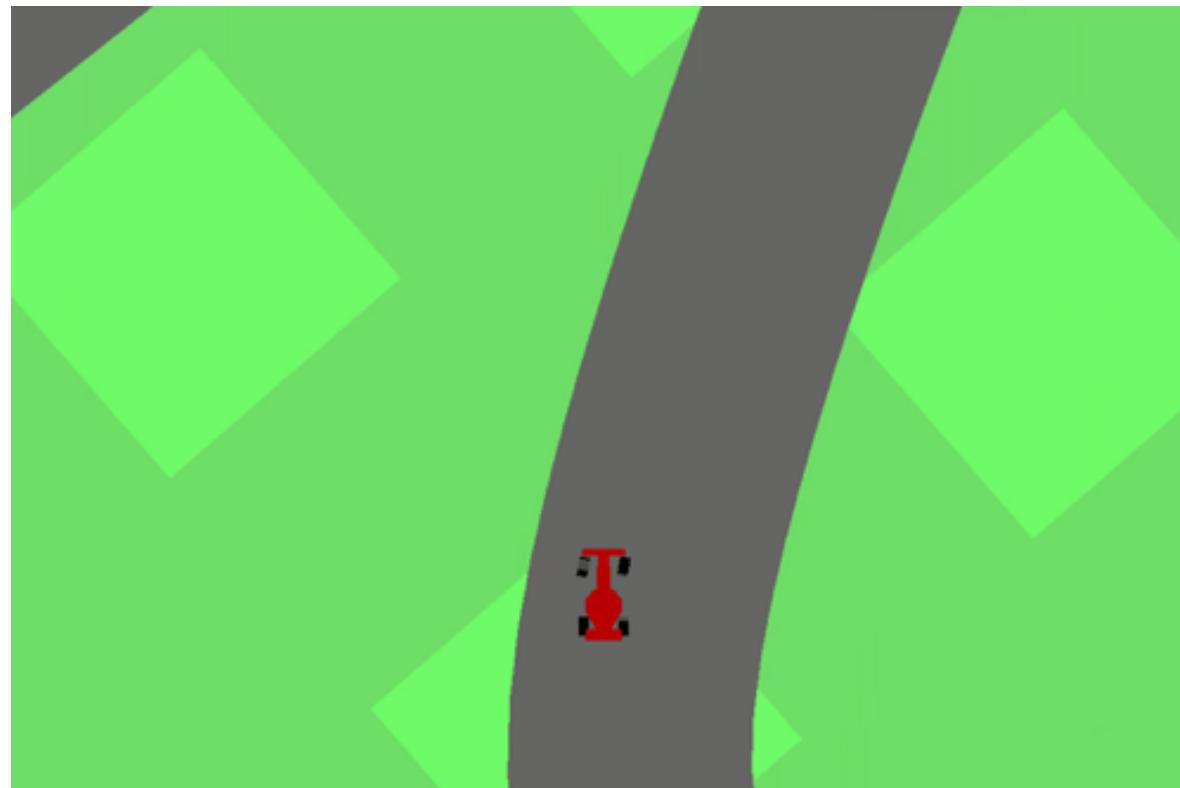
**Competitive Pong
Tournament**
Reinforcement Learning
Course@CUHK

Spring Semester, 2020

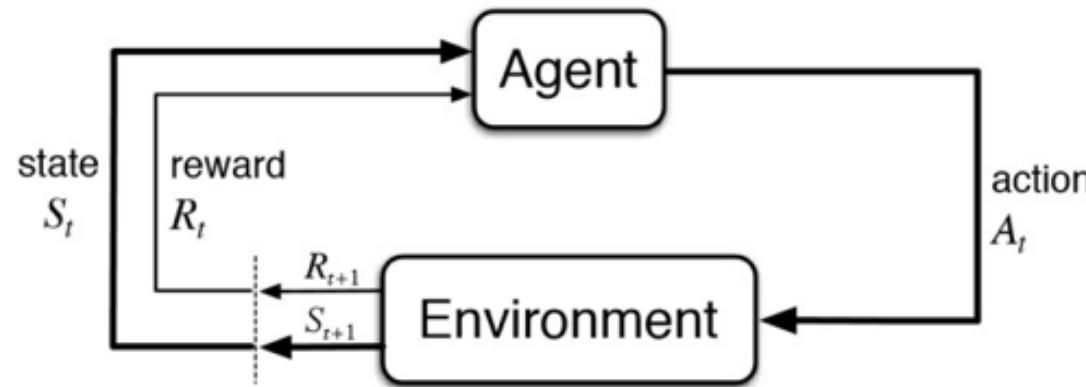
Lecturer: Bolei Zhou
TA: Zhenghao Peng

Open-end Competition in Assignment 5

- Another possible choice: Competitive car-racing (in development)



What is reinforcement learning and why we care



a computational approach to learning whereby **an agent** tries to **maximize** the total amount of **reward** it receives while interacting with a complex and uncertain **environment**.

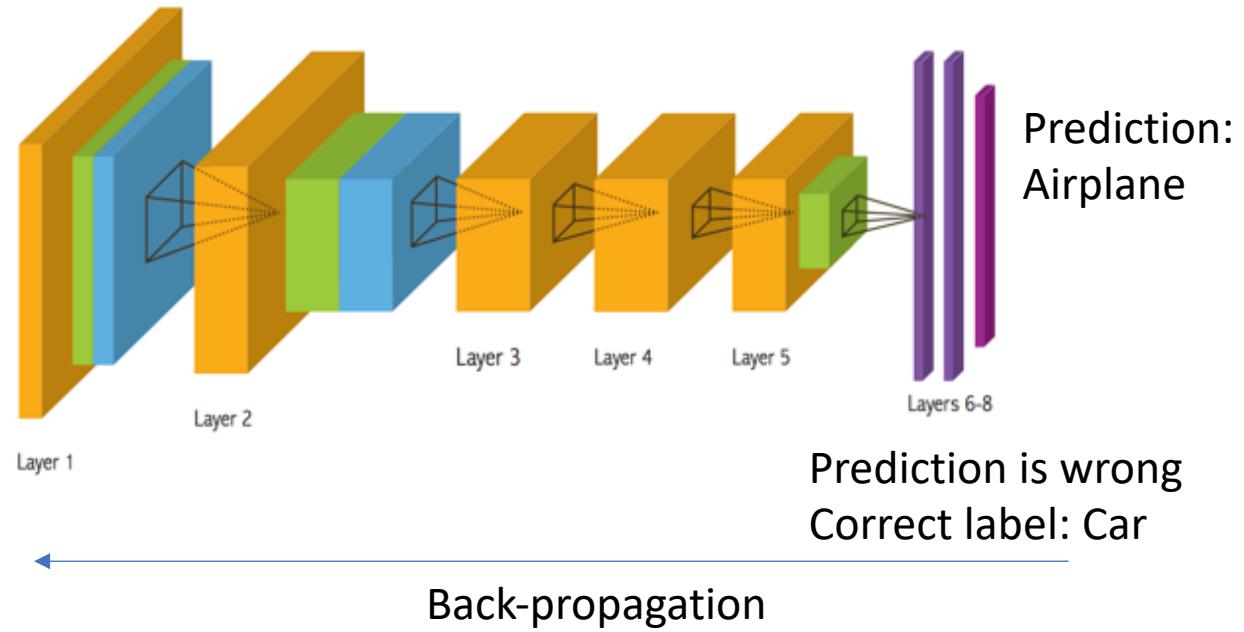
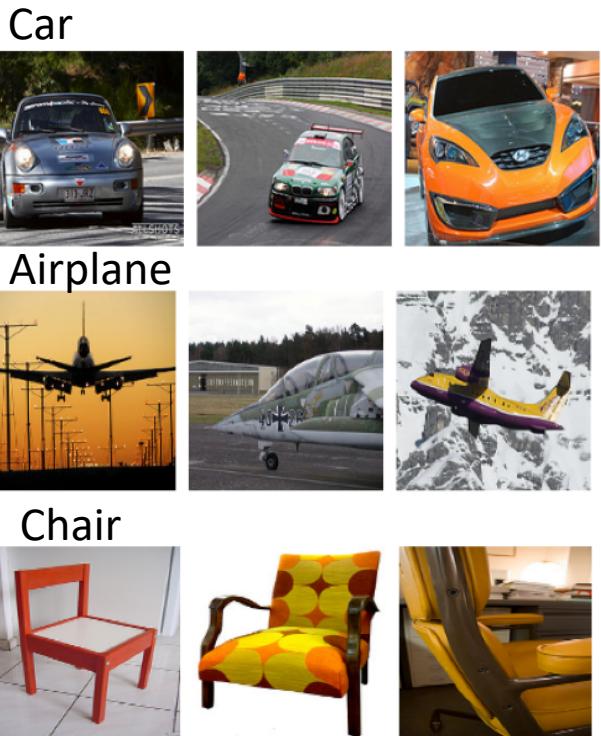
- Sutton and Barto

Supervised Learning

Learn to classify object

- Annotated images, data follows i.i.d distribution.
- Learners are told what the labels are.

Training annotated data



Reinforcement Learning

Learn to play Breakout

- Data are not i.i.d. Instead, a correlated time series data
- No instant feedback or label for correct action

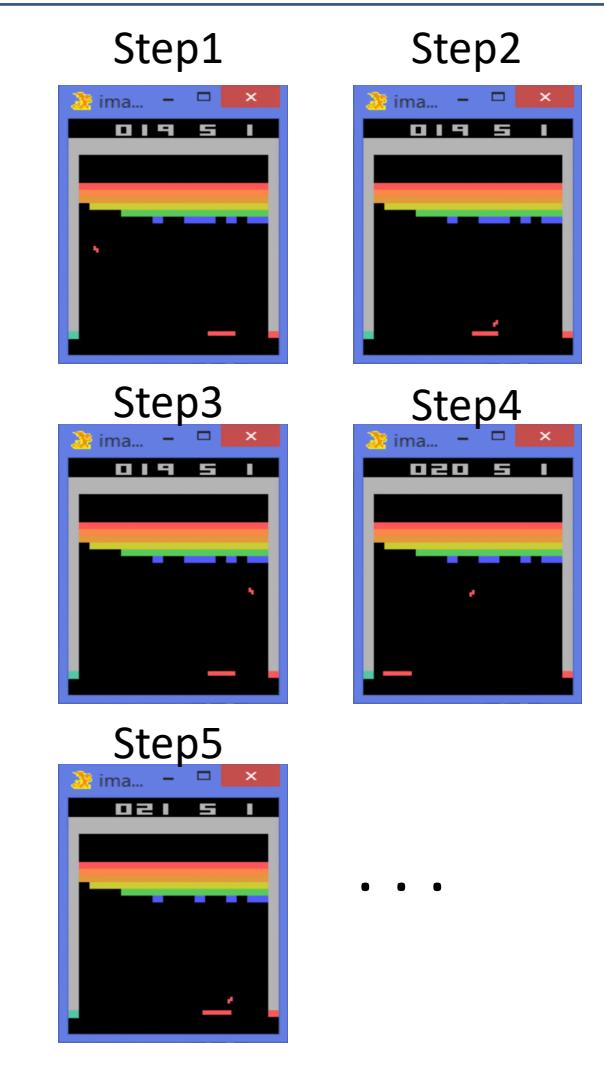
Action: Move LEFT or Right



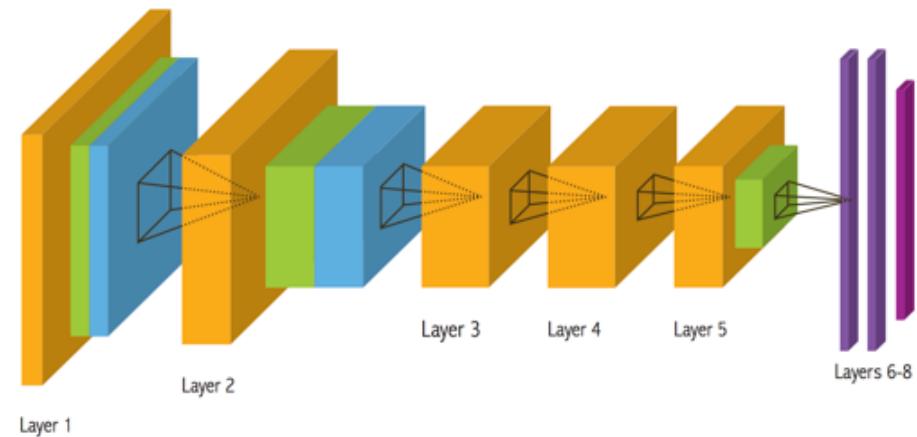
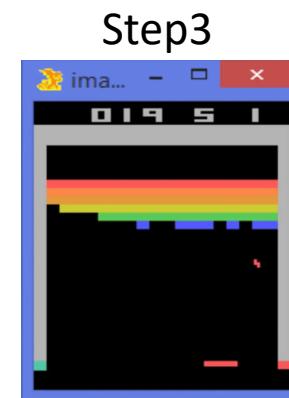
Atari Game: Breakout

Reinforcement Learning

Training data



Human playing?



Correct or Wrong???
Don't know for now, until game is over
Delayed reward

←
Backpropagation?

Difference between Reinforcement Learning and Supervised Learning

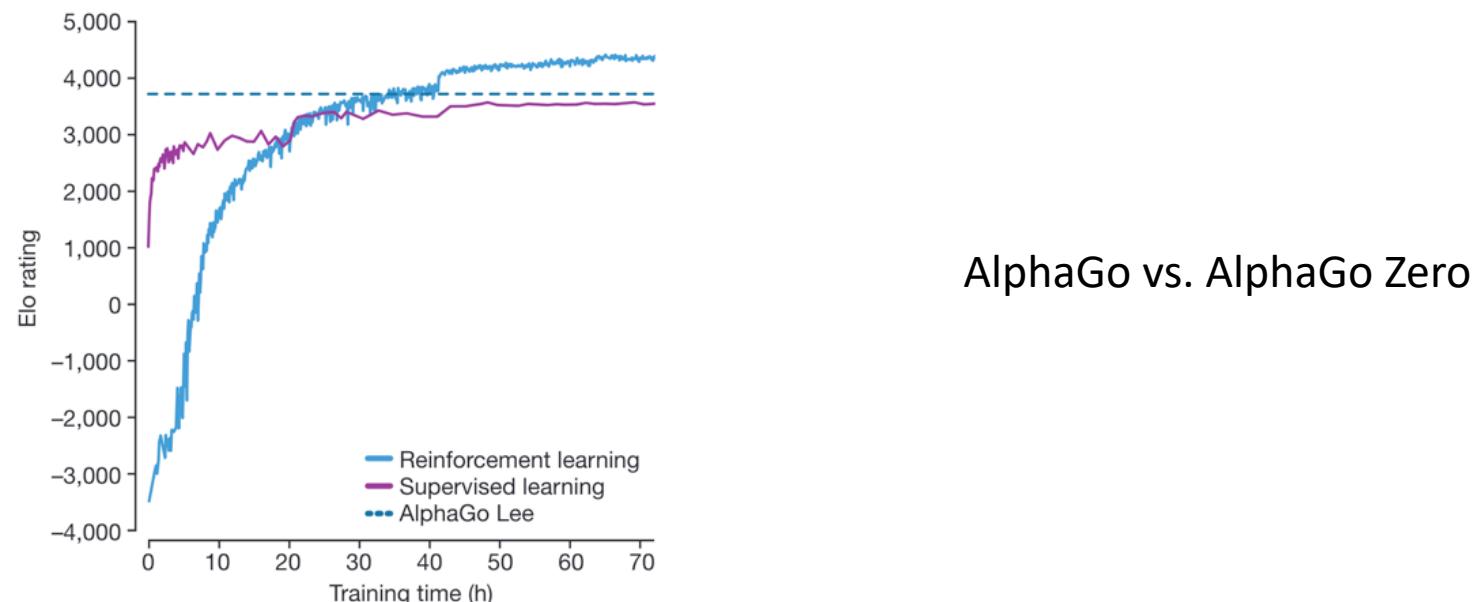
- Sequential data as input (not i.i.d)
- The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them.
- Trial-and-error exploration (balance between exploration and exploitation)
- There is no supervisor, only a reward signal, which is also delayed

Properties of reinforcement learning

- Trial-and-error exploration
- Delayed reward
- Time matters (sequential data, non i.i.d data)
- Agent's actions affect the subsequent data it receives
(agent's action changes the environment)

Why we care about RL

- Learn to control something via trial-and-error in model-free setting
- May achieve super-human performance
 - Upper bound for supervised learning is human-performance.
 - Upper bound for reinforcement learning?



Examples of reinforcement learning

- A chess player makes a move: the choice is informed both by planning-anticipating possible replies and counterreplies.
- A gazelle calf struggles to stand, 30 min later it is able to run 36 kilometers per hour.
- Portfolio management.
- Playing Atari game



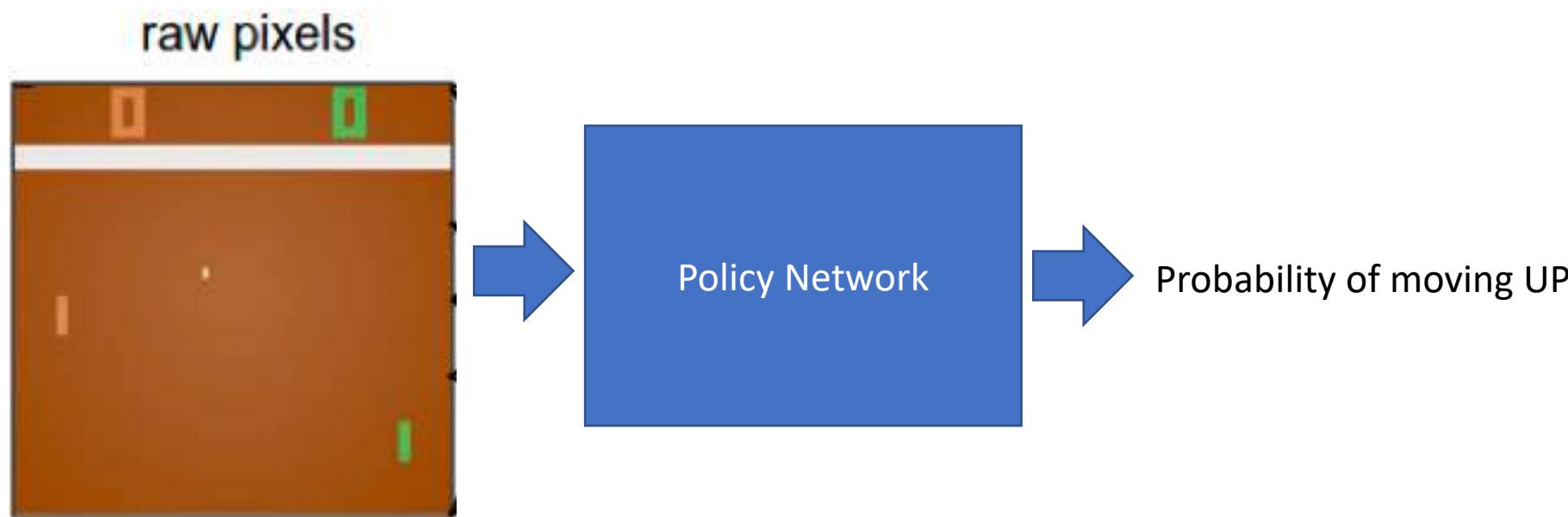
RL example: Pong

Action: move UP or DOWN



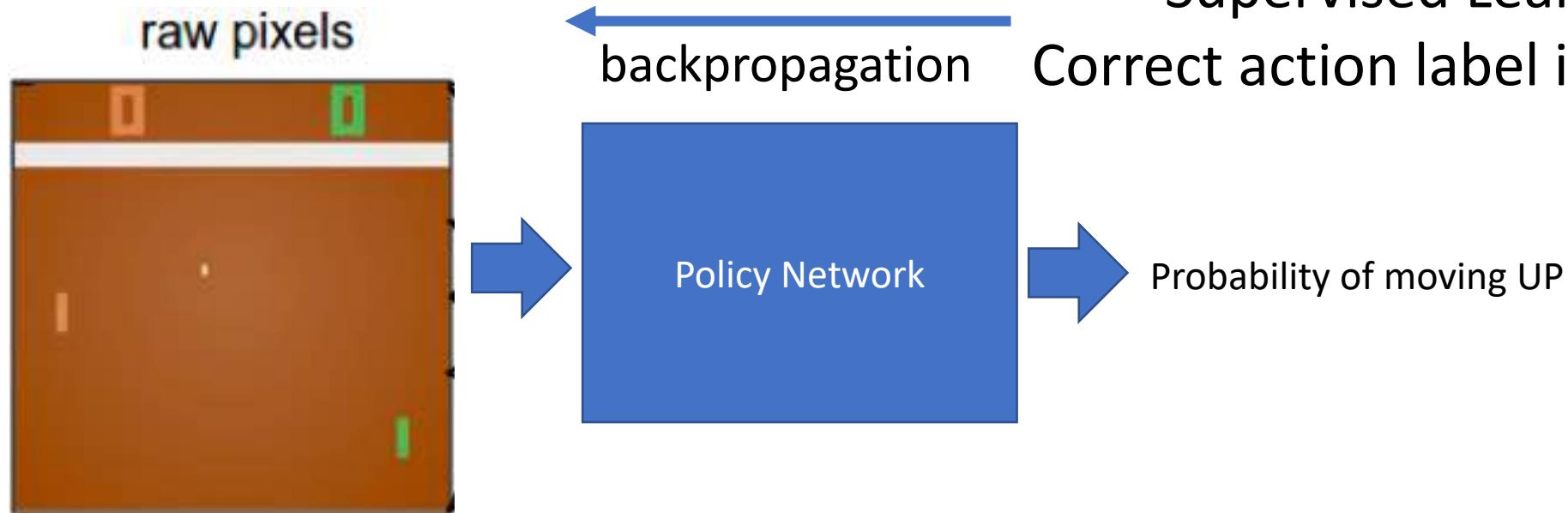
RL example: Pong

Action: move UP or DOWN



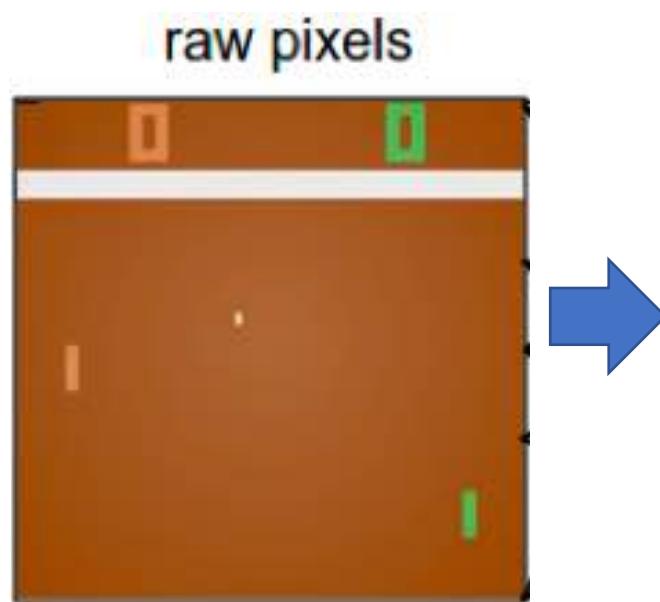
RL example: Pong

- Action: move UP or DOWN

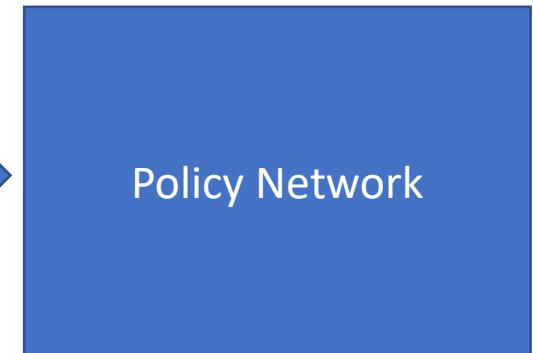


RL example: Pong

- Action: move UP or DOWN



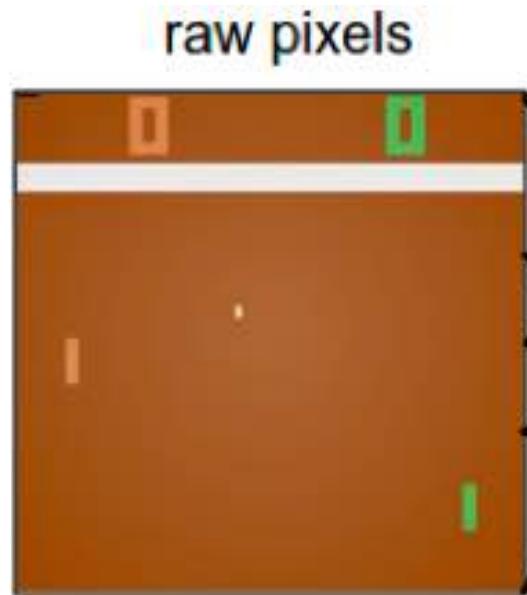
Reinforcement Learning:
Sample actions (rollout), until game is over,
Then penalize each action



Probability of moving UP

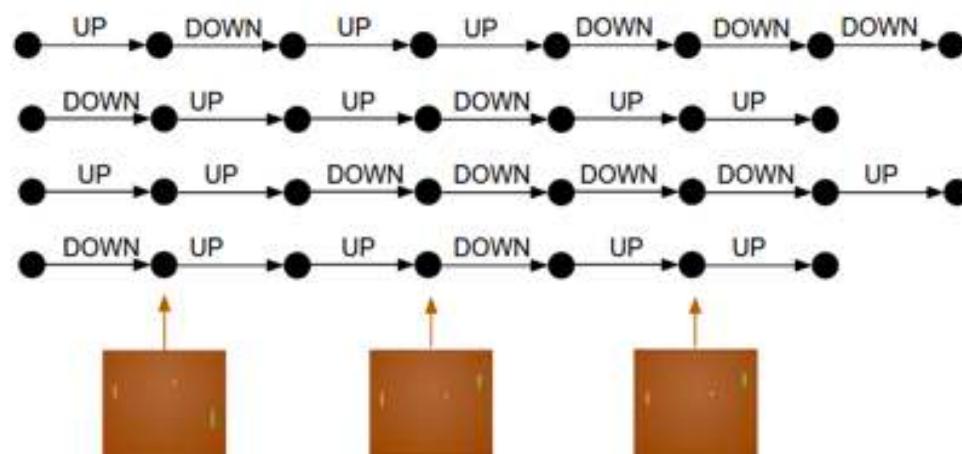
RL example: Pong

- Action: move UP or DOWN



Reinforcement Learning:
Sample actions (rollout), until game is over,
Then penalize each action

Possible rollout sequence:



Eventual Reward:

WIN

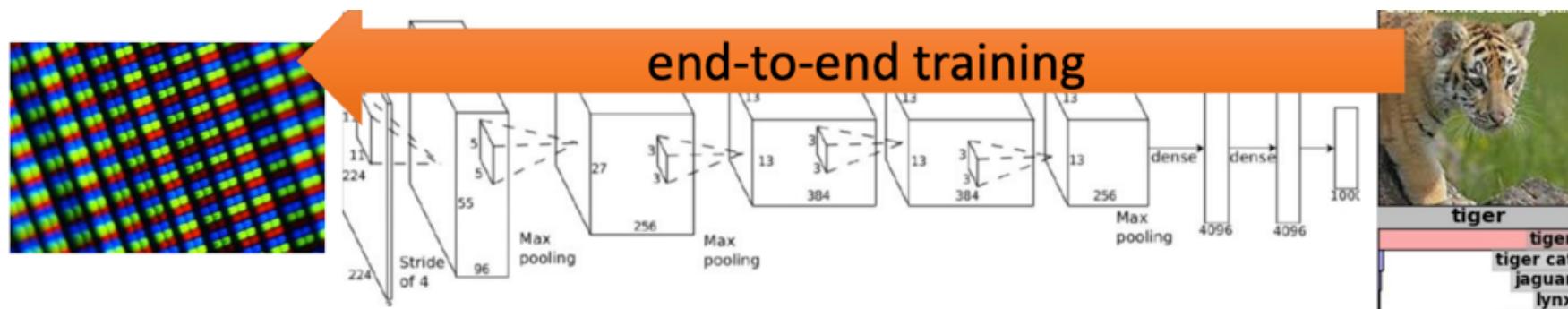
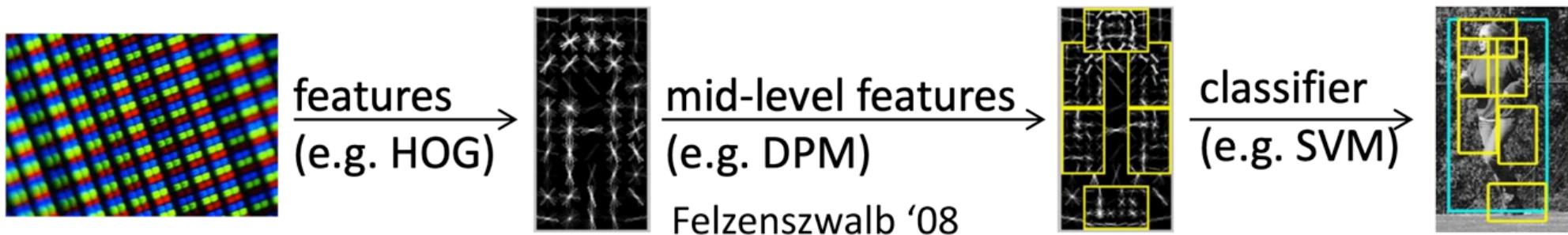
LOSE

LOSE

WIN

Why deep reinforcement learning?

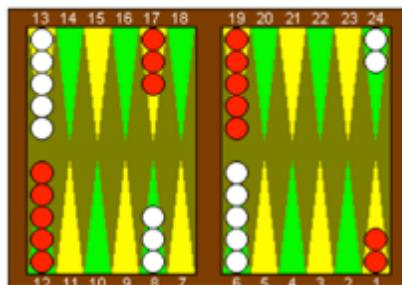
- Analogy to traditional CV and deep CV



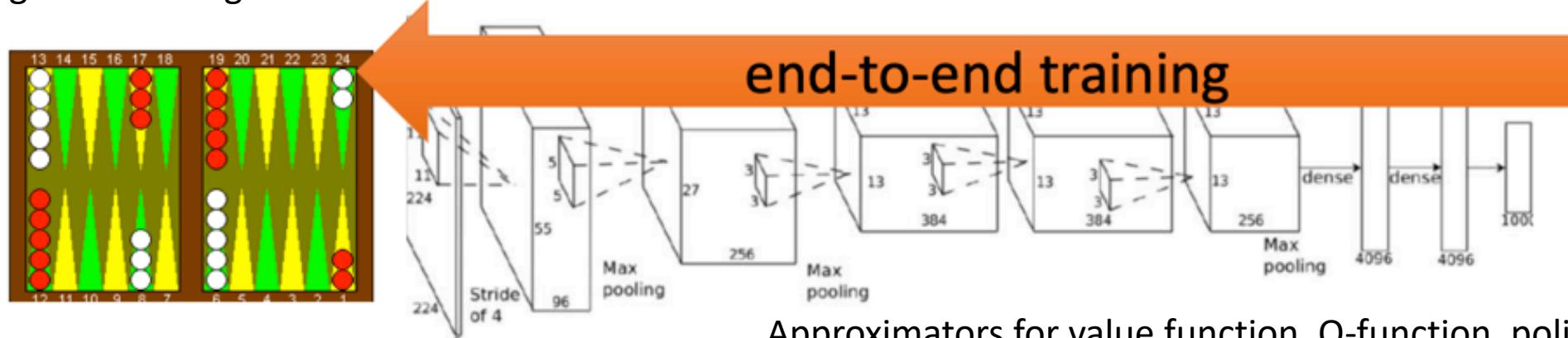
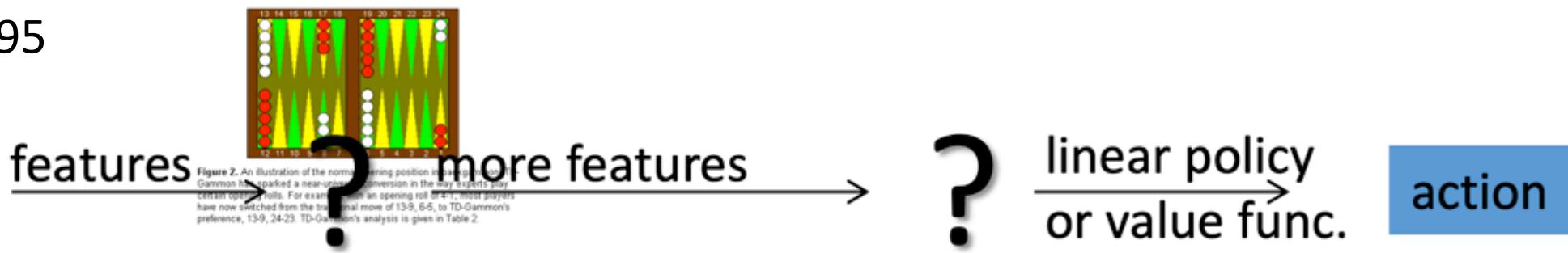
Why deep reinforcement learning?

- Standard RL and deep RL

TD-Gammon, 1995



game of backgammon



Why RL works now?

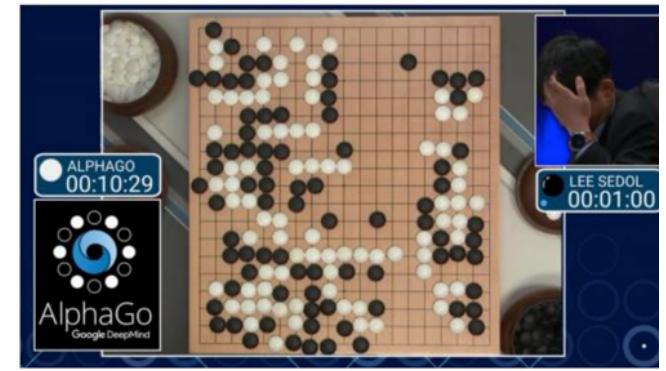
- One of the most exciting areas in machine learning



Game playing



Robotics



Beating best human player

[Playing Atari with Deep Reinforcement Learning](#)

[Mastering the game of Go without Human Knowledge](#)

Why RL works now?

- Computation power: many GPUs to do trial-and-error rollout
- End-to-end training, features and policy are jointly optimized toward the end goal
 - Acquire the high degree of proficiency in domains governed by simple, known rules



Game playing



Robotics

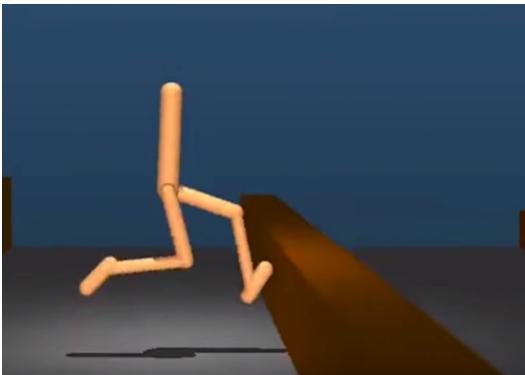


Beating best human player

What are the applications of RL?

Some interesting examples:

Learning to walk



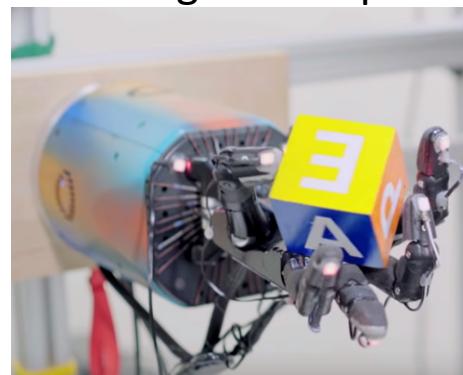
<https://www.youtube.com/watch?v=gn4nRCC9TwQ>

Learning to grasp



<https://ai.googleblog.com/2016/03/deep-learning-for-robots-learning-from.html>

Learning to manipulate



<https://www.youtube.com/watch?v=jwSbzNHGfIM>

Learning to dress



<https://www.youtube.com/watch?v=ixmE5nt2o88>

Learning to walk



Learning to grasp

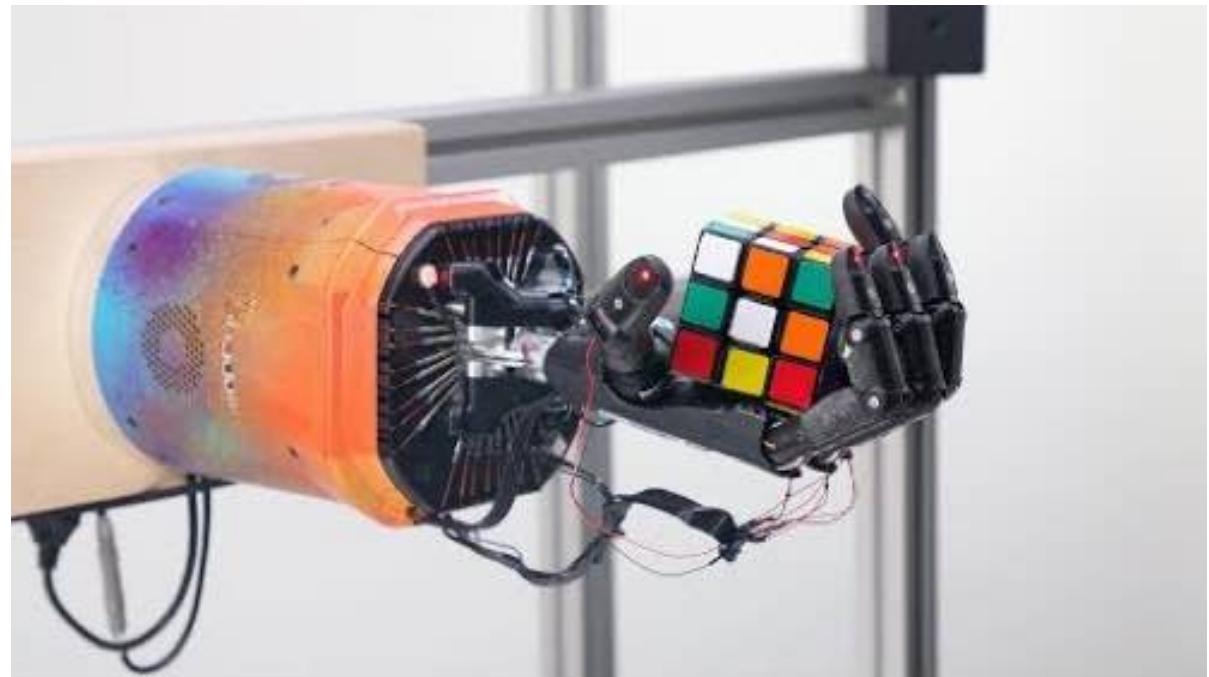


Learning to move fingers

2018



2019



OpenAI

Learning to dress



Learning to dress: synthesizing human dressing motion via deep reinforcement learning.

<https://ckllab.stanford.edu/learning-dress-synthesizing-human-dressing-motion-deep-reinforcement-learning>

Learning to race



Break

- Next lecture:
 - RL basics
 - coding with RL