CMPS 4660/6660 Reinforcement Learning

Prof. Zizhan Zheng

Fall 2020

Agenda

- Course Logistics
- Introduction to Reinforcement Learning

Personal

- Instructor: Prof. Zizhan Zheng
  - Office: Stanley Thomas 307B
  - Email: zzheng3@tulane.edu
  - Research Interests: Networks, Machine learning, Cybersecurity
  - Office hours (virtual): Wed 10-11 am and by appointment

Meeting Plan

- Hybrid class (60% on ground, 40% online)
  - Aug & Sep: Tu – on-ground, Th – online
  - Will meet more in person in Oct and Nov
- Class meeting time: Tu & Th 12:25pm-1:35pm
  - Reading materials and discussions will be assigned on Canvas to compensate for the shortened class time
  - All classes will be recorded
- Zoom links
  - Lecture: https://tulane.zoom.us/j/99301709427
  - Office hour: https://tulane.zoom.us/j/97536228925

Attendance Policy

- All students should follow the hybrid class meeting schedule unless
  - you don’t feel well or have COVID symptoms
  - you have been approved for remote learning
- If you cannot attend class for any reason, you are responsible for communicating with me to make up any work you may miss
- If you are sick or told to quarantine, you can contact your case manager, and have your case management contact me directly

Course Overview

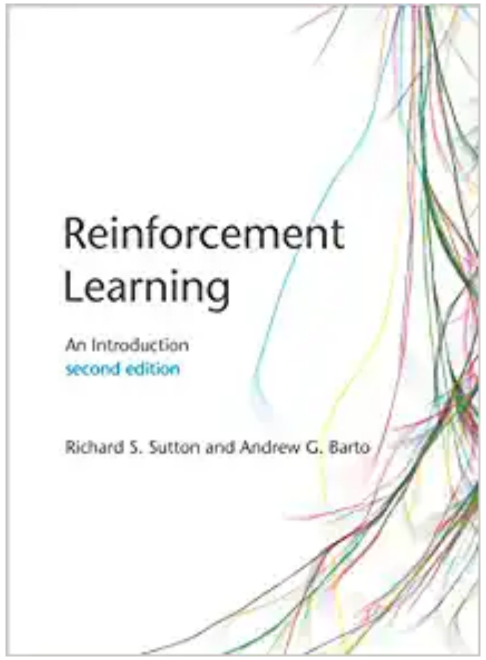
- Objectives
  - Introduce both the classic results and state-of-the-art research in RL.
  - Focus on both the theoretical foundation and the application of RL
- Prerequisites
  - Knowledge of discrete probabilities and algorithms
  - Comfortable with rigorous mathematical reasoning
  - Programming skills

Topics

- Markov Decision Processes
- Dynamic Programming
- Model-Free Prediction and Control
- Value Function Approximation
- Policy Gradient Methods
- Planning and Learning
- Exploration and Exploitation
- Imitation Learning, Inverse RL
- Transfer Learning, Meta-learning
- Multi-Agent Reinforcement Learning
- …

Textbook and References

- Textbook: Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction (2nd edition)*, A Bradford Book, 2018.
- References
  - Dimitri Bertsekas, *Reinforcement Learning and Optimal Control*, Athena Scientific, 2019.
  - Csaba Szepesvári, *Algorithms for Reinforcement Learning*, 2010.
  - Aleksandrs Slivkins, *Introduction to Multi-Armed Bandits*, 2019.
- Research papers, tutorials, etc.



Grading Policy

- Problem Sets - 20%
- Labs - 15%
- Midterm - 20%
- Final Project - 35%
- Class participation - 10%

A >= 90%; B >= 80%; C >= 70%; D >= 60%; F < 60%

+/- grades will be given for borderline cases.

- All grades will be posted on Canvas.

Problem Sets (20%)

- Given once every one or two weeks
  - Due in the following week
  - Some questions are for graduate students only
  - Undergraduate will get extra points for trying those harder problems

Labs (15%)

- 3 lab assignments
  - Each student should work on Lab 1 individually
  - Each project group can work on Labs 2 & 3 together: document contributions clearly
- OpenAI Gym will be used for some of the labs
- GPU access may be needed for Lab 3 and the final project
  - Google Colab

Final Project (35%)

- Two students per group
  - Group formation due Sep 1
- Presentations (tentative)
  - Proposal (15 min - Sep 24)
  - Progress update (15 min - Oct 27)
  - Mini-lecture (30 min - Nov 10 & Nov 12)
  - Final presentation (30 min - Nov 30 – Dec 5)
- Final report (Dec 6)

Class Participation (10%)

- Q&A in class & during office hours
- Pop quizzes
- Homework solution presentation
- Online discussions

Late Policy

- 6 grace days that may be applied to homework/lab assignments
- No more than 2 grace days on any single assignment
  - assignment submitted > 2 days past the deadline (or no late day credit left) will get zero credit
- No late days for the final presentation and report

- Course Webpage
  - http://www.cs.tulane.edu/~zzheng3/teaching/cmps6660/fall20
  - Used to post weekly schedule, assignments, lecture slides, reading material, etc.
- Canvas
  - Used to post grades, discussions, and reading material, etc.

Introduction

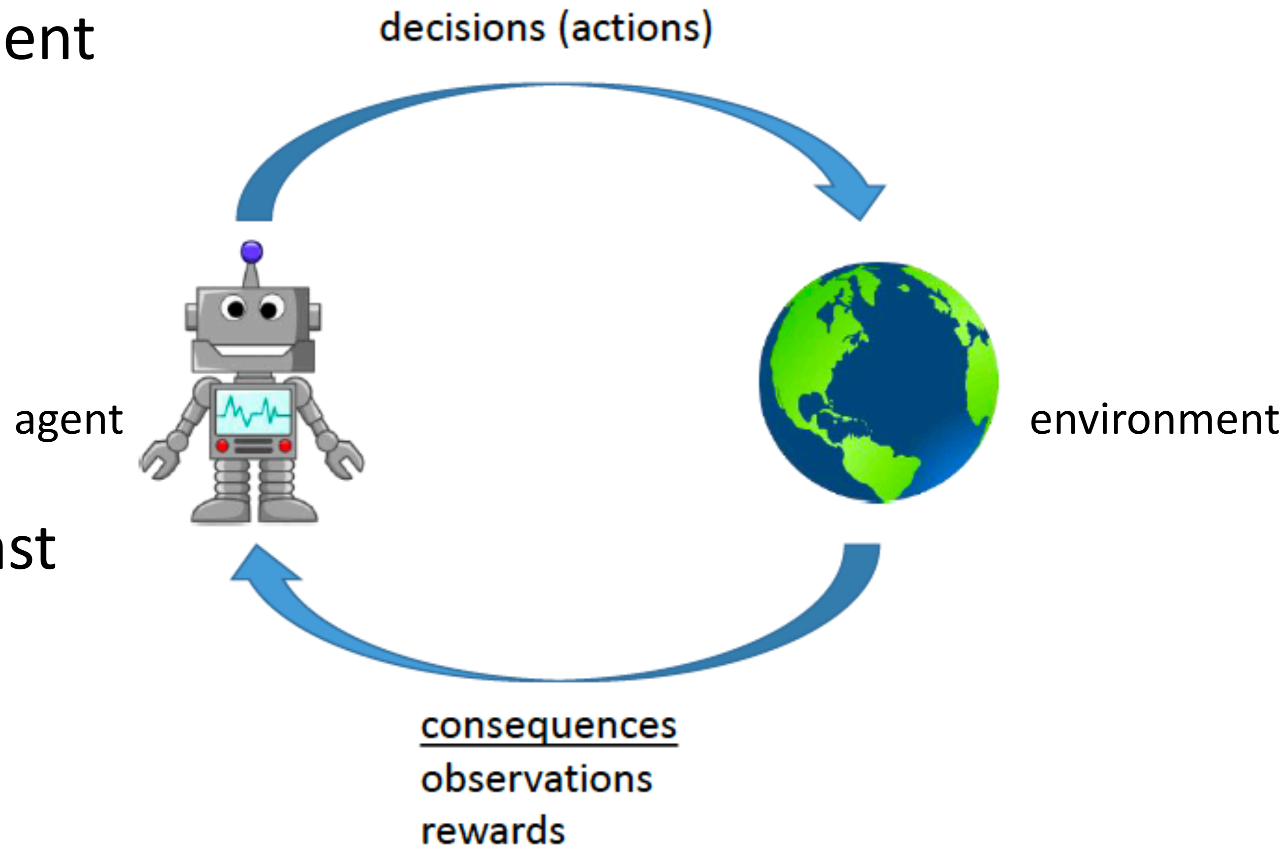
CMPS 4660/6660: Reinforcement Learning

Acknowledgement: slides adapted from David Silver's RL course and Berkeley CS285.

Reinforcement Learning

- Learn to make good sequences of decisions in an uncertain environment
  - Goal directed
  - Sequence of actions
  - Learn from interaction

- Learn a policy = a mapping from past experience to action



Example: Inventory Management

- Actions: what to purchase
- Observations: inventory level
- Rewards: profit - storage cost

Example: Robotics

Actions: motor current or torque

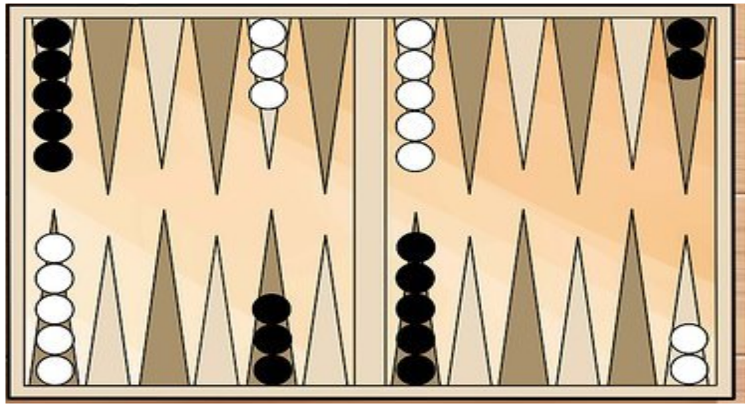
Observations: images

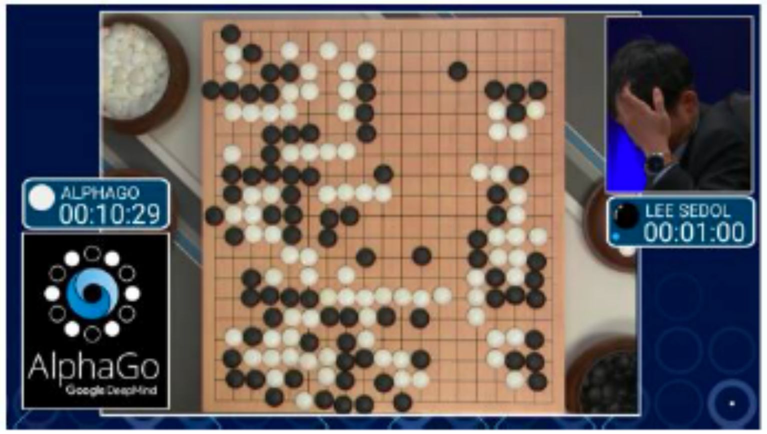
- State estimation

Rewards: task success measure

Example: Classic Games

- Actions: a legal game action
  - Simple, known rules
- Observations: board position
  - perfect information: Chess, Checkers, Go
  - imperfect information: Poker
- Rewards
  - win (+1), lose (-1), otherwise (0)
  - Delayed feedback



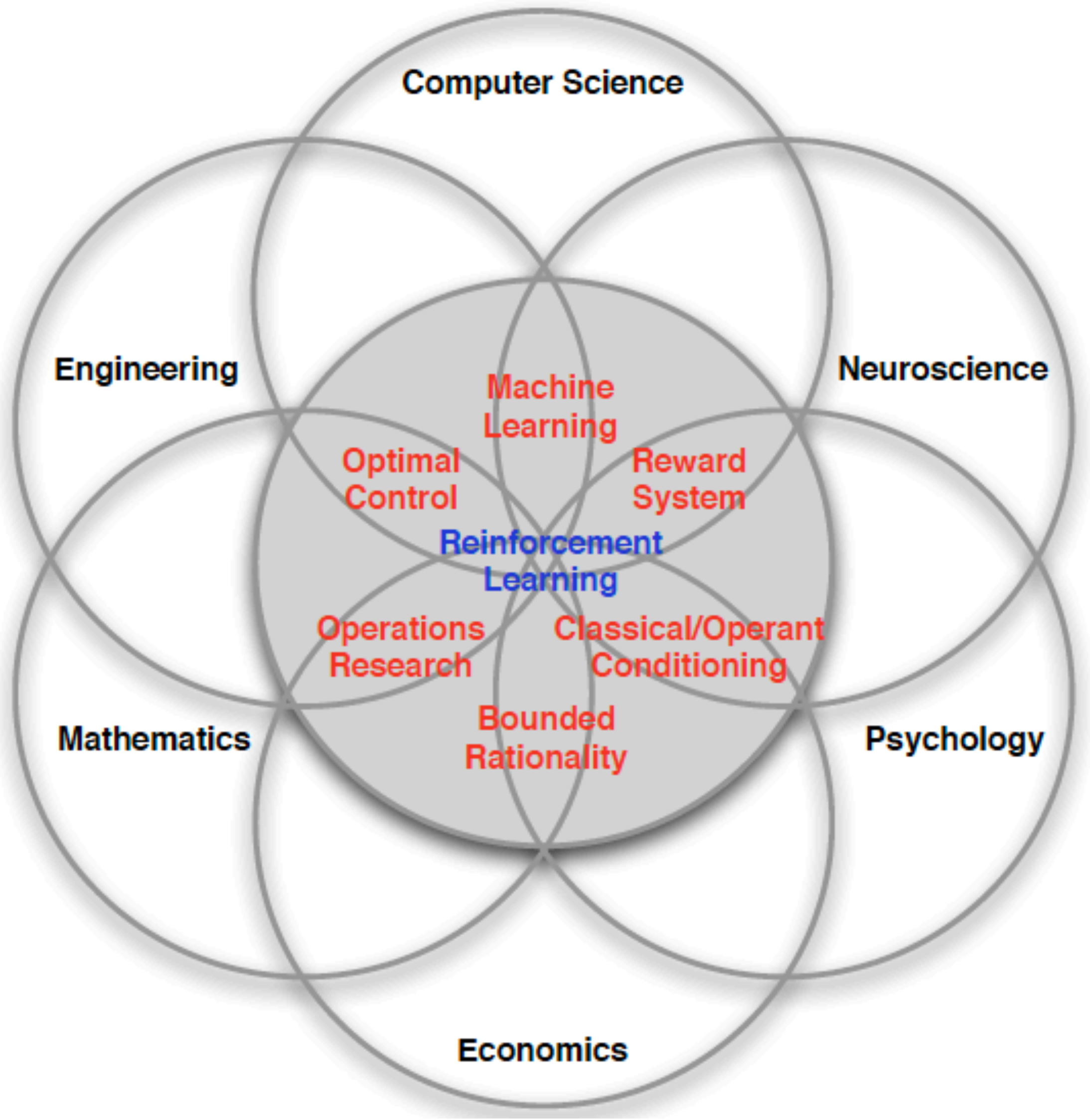


Types of learning task

- Supervised learning
  - Learn to predict an output when given an input vector
- Unsupervised learning
  - Discover a good internal representation of the input
- Reinforcement learning
  - Learn to select an action to maximize payoff

-- Hinton: "The Next Generation of Neural Networks", SIGIR'20

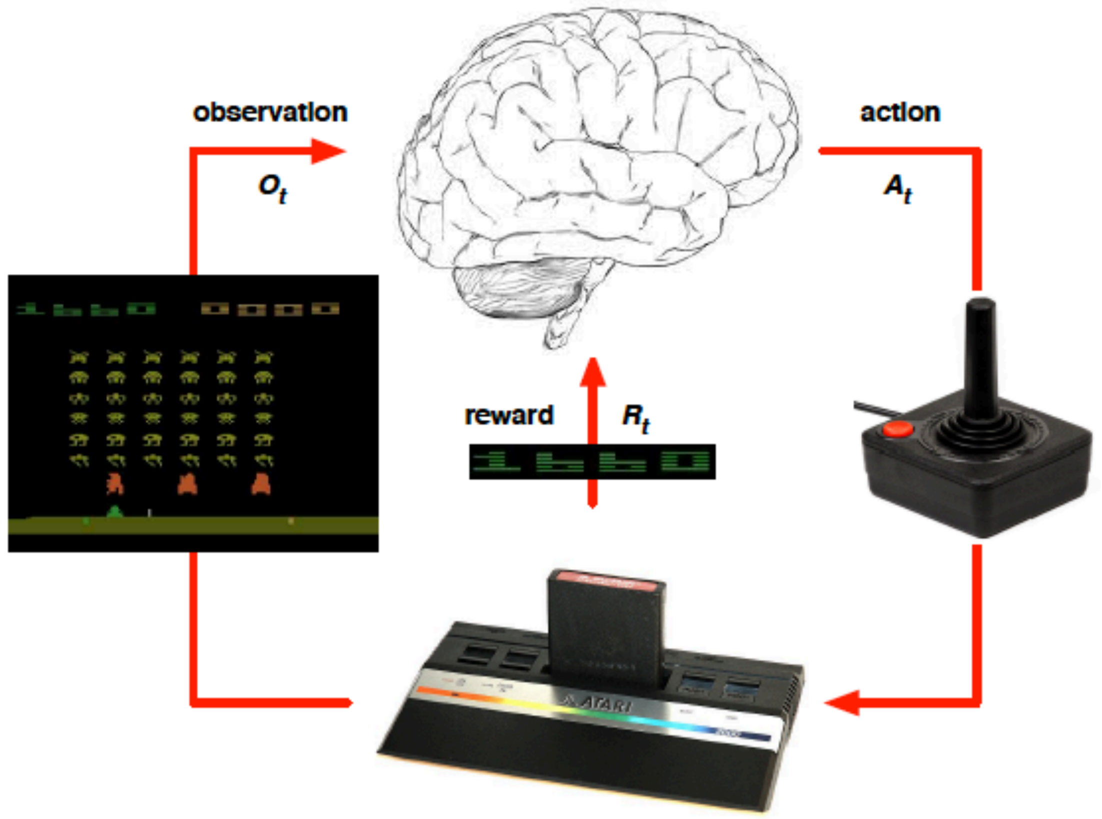
Many Faces of Reinforcement Learning



Learning and Planning

- Two fundamental problems in sequential decision making
- Reinforcement Learning:
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- Planning:
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy
  - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

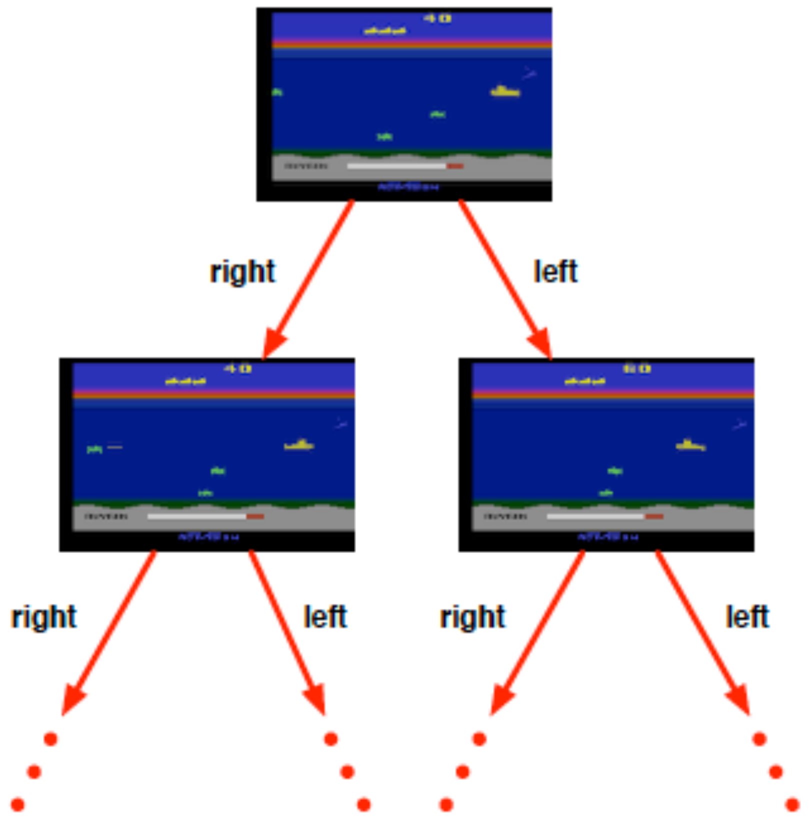
Atari Example: Reinforcement Learning



- Rules of the game are unknown
- Learn directly from interactive game-play
- Pick actions on joystick, see pixels and scores

Atari Example: Planning

- Rules of the game are known
- Can query emulator
  - perfect model inside agent's brain
- If I take action a from state s:
  - what would the next state be?
  - what would the score be?
- Plan ahead to find optimal policy
  - e.g. tree search



Exploration and Exploitation

- Reinforcement learning is like trial-and-error learning
- The agent should discover a good policy
- From its experiences of the environment
- Without losing too much reward along the way

Exploration and Exploitation

- *Exploration* finds more information about the environment, e.g., *try a new restaurant*
- *Exploitation* exploits known information to maximize reward, e.g., *go to your favorite restaurant*
- A fundamental tradeoff in any sequential decision making under uncertainty
- Multi-armed bandit: a canonical model to study the tradeoff

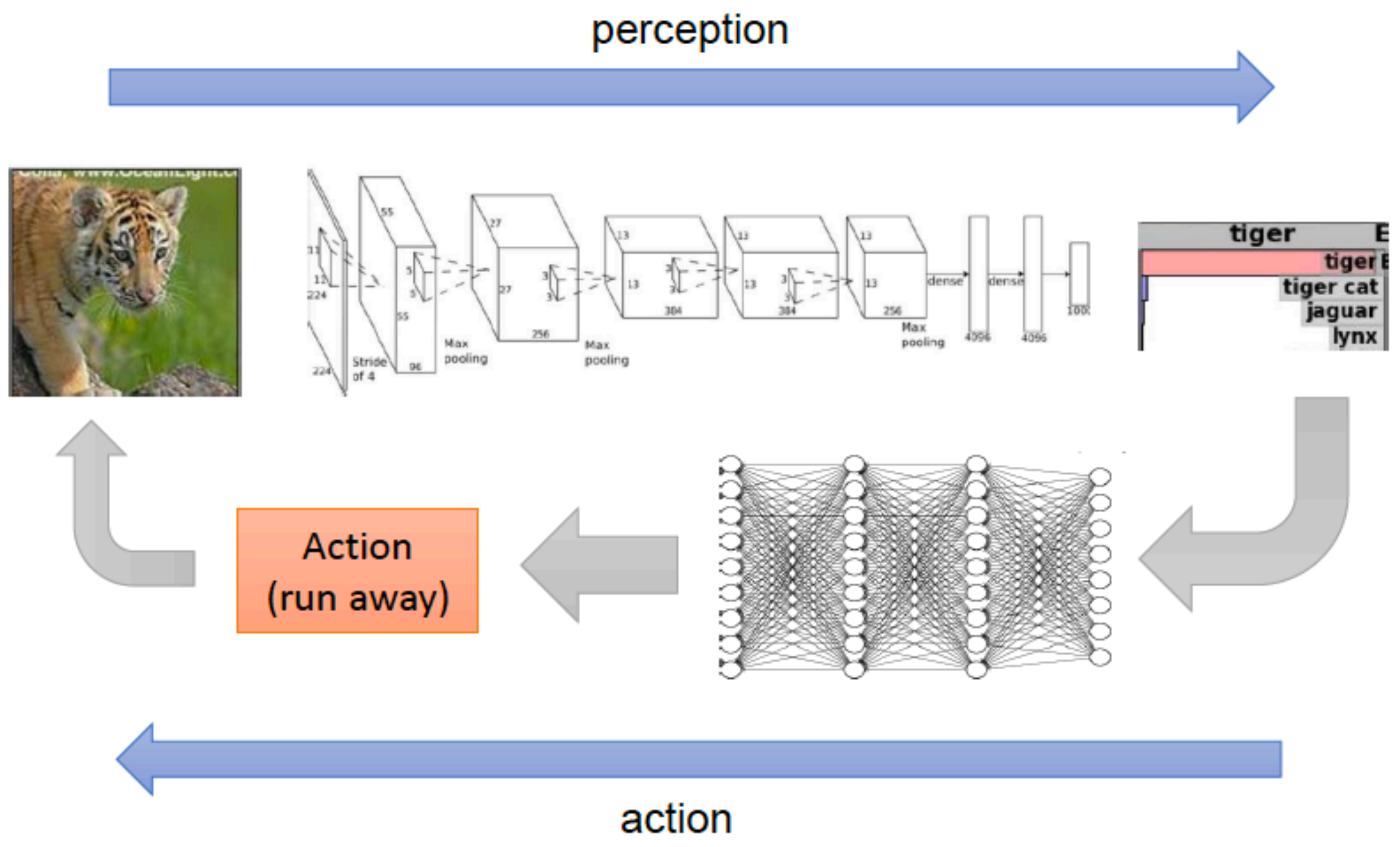
Generalization

- Reinforcement learning can be used to solve *large* problems, e.g.
  - Backgammon: 10^{20} states
  - Computer Go: 10^{170} states
  - Helicopter: continuous state space
- Function approximation
  - Generalize from seen states to unseen states

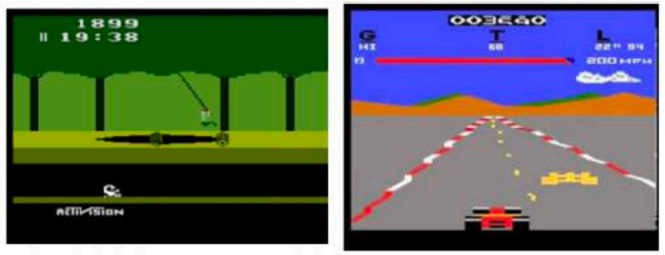
Deep Reinforcement Learning

- Reinforcement Learning => a framework for sequential decision making
- Deep learning => models that can handle unstructured environment
- Deep RL => algorithms that can solve very complex decision-making problems

End-to-end decision making via Deep RL



Why should we study this now?



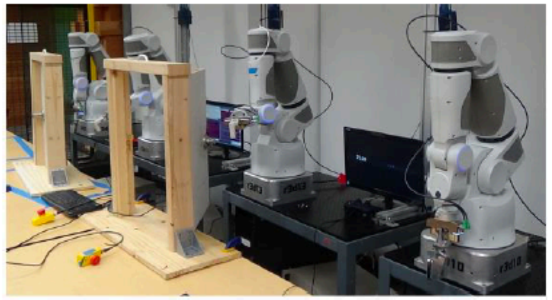
Atari games:

Q-learning:

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, et al. "Playing Atari with Deep Reinforcement Learning". (2013).

Policy gradients:

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. "Trust Region Policy Optimization". (2015).
V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, et al. "Asynchronous methods for deep reinforcement learning". (2016).



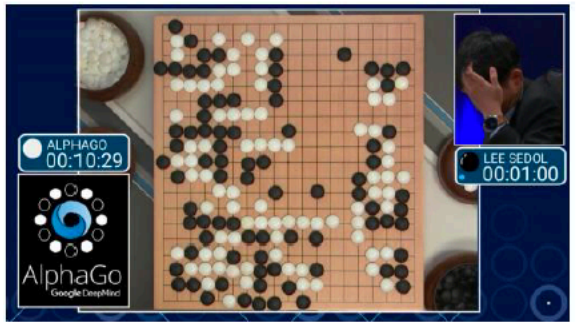
Real-world robots:

Guided policy search:

S. Levine*, C. Finn*, T. Darrell, P. Abbeel. "End-to-end training of deep visuomotor policies". (2015).

Q-learning:

D. Kalashnikov et al. "QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation". (2018).



Beating Go champions:

Supervised learning + policy gradients + value functions + Monte Carlo tree search:

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, et al. "Mastering the game of Go with deep neural networks and tree search". Nature (2016).

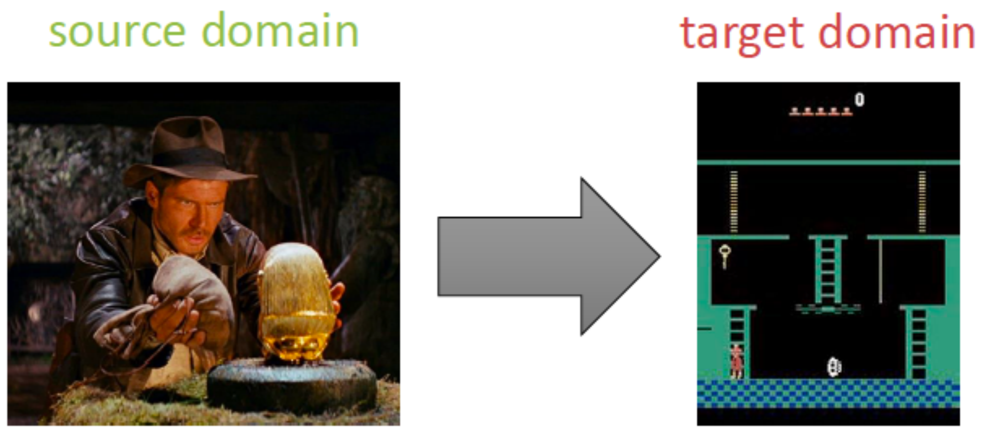
Beyond learning from reward

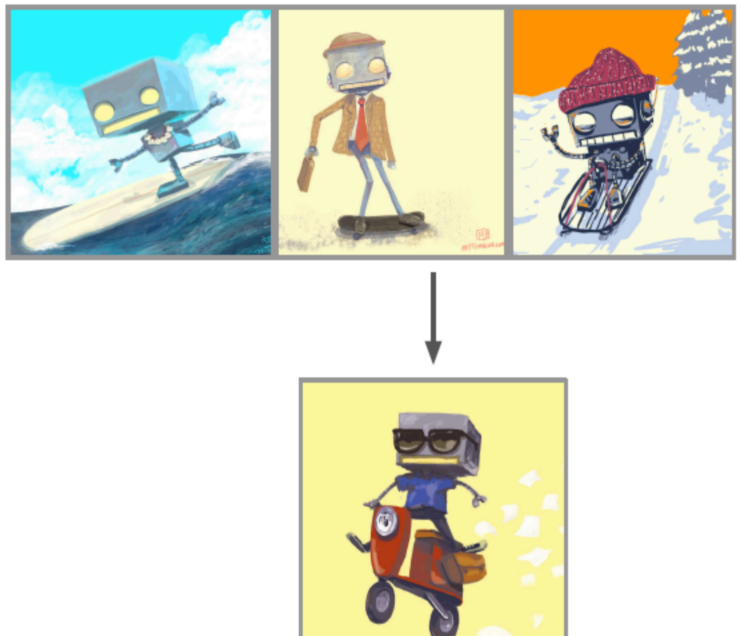
- Learning from demonstrations
  - Directly copying observed behavior (imitation learning)
  - Inferring rewards from observed behavior (inverse RL)

https://www.youtube.com/watch?v=wbhCXPSNNH0

Beyond learning from reward

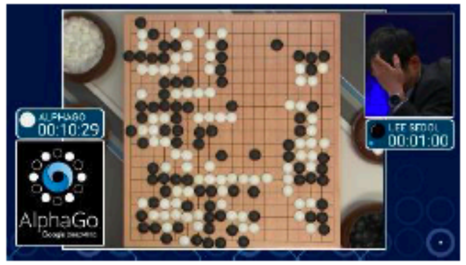
- Learning from other tasks
  - Transfer learning
  - Meta learning: learning to learn

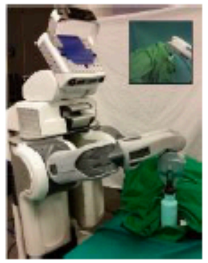
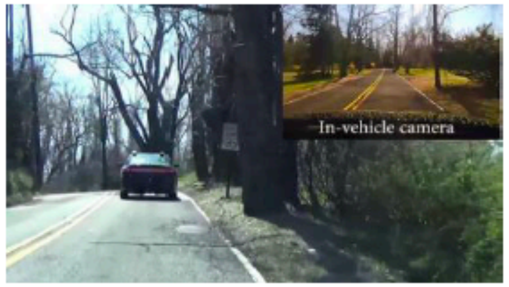




What can deep learning & RL do well now?

- Acquire high degree of proficiency in domains governed by simple, known rules
- Learn simple skills with raw sensory inputs, given enough experience
- Learn from imitating enough human-provided expert behavior



What has proven challenging so far?

- Humans can learn incredibly quickly
  - Deep RL methods are usually slow
- Humans can reuse past knowledge
  - Transfer learning in deep RL is an open problem
- Not clear what the reward function should be
- Not clear what the role of prediction should be

“There are no methods that are guaranteed to work for all or even most problems, but there are enough methods to try on a given challenging problem with a reasonable chance that one or more of them will be successful in the end.”

-- Dimitri P. Bertsekas