Lecture 1: Introduction to RL

Zhi Wang & Chunlin Chen

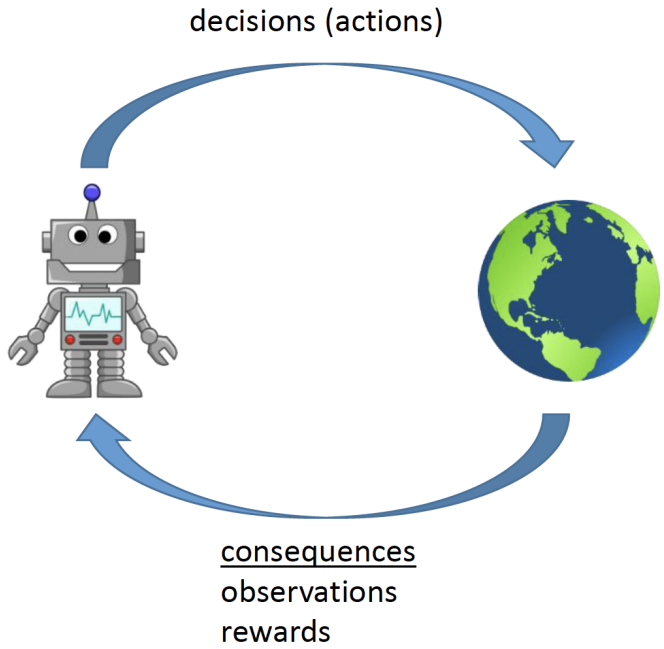
Department of Control and Systems Engineering
Nanjing University

Sept. 24th, 2020

Table of Contents

What is reinforcement learning (RL)?

- Mathematical formalism for learning-based decision making
- Approach for sequential decision-making and control **from experience**



How is RL different from other machine learning topics?

- Standard supervised learning:
  - given $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$
  - learn to predict y from $\boldsymbol{x}$, $f(\boldsymbol{x}) \approx y$
- Usually assumes:
  - independent and identically distributed (i.i.d.)
  - known ground truth outputs in training

- RL:
  - Data is not i.i.d., previous outputs influence future inputs
  - Ground truth answer is not known, only known if we succeed or failed (more generally, we know the reward)

Markov Decision Process (MDP)

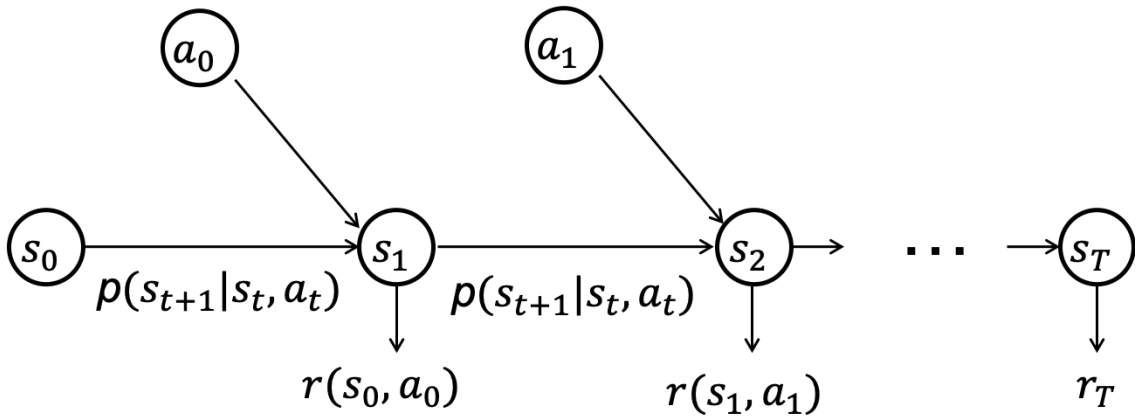
$M = \langle S, A, T, R \rangle$

S: State space — state $s \in S$ (discrete/continuous)

A: Action space — action $a \in A$ (discrete/continuous)

T: Transition operator — $T_{i,j,k} = p(s_{t+1} = j | s_t = i, a_t = k)$

R: Reward function — $R_{i,j,k} = r(s_{t+1} = j | s_t = i, a_t = k)$

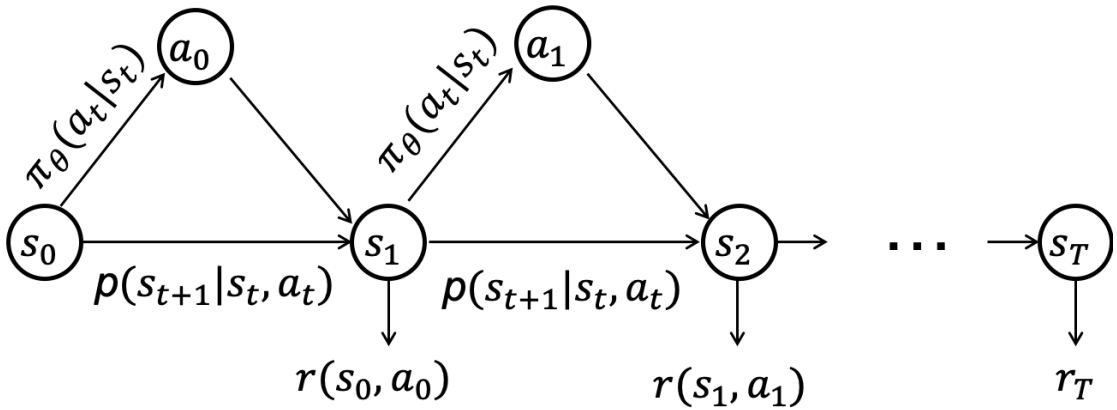


The Goal of RL

- Find **optimal policies** to maximize cumulative reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} r(s_t, a_t) \right]$$

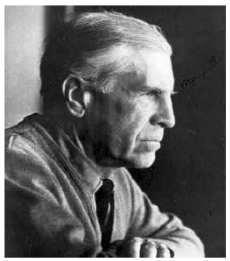
  - In a **trial-and-error** manner
  - A general **optimization** framework for sequential decision-making

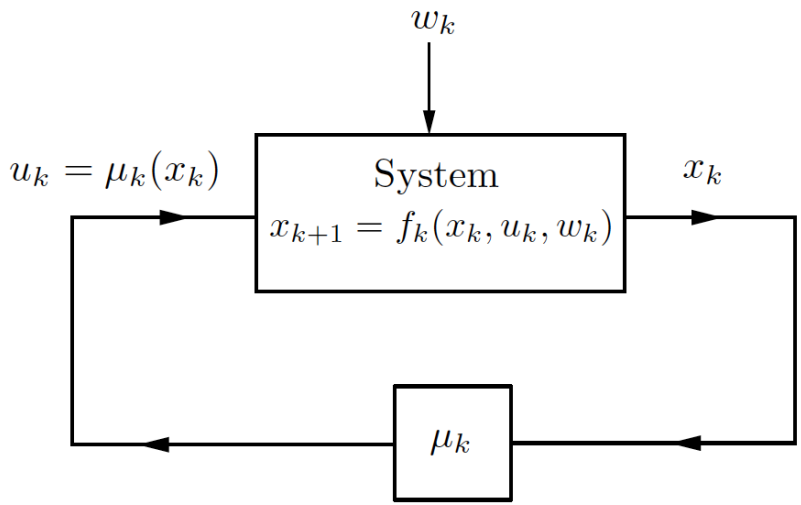


RL - A Control Perspective

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action





- w_k: random disturbance, system:

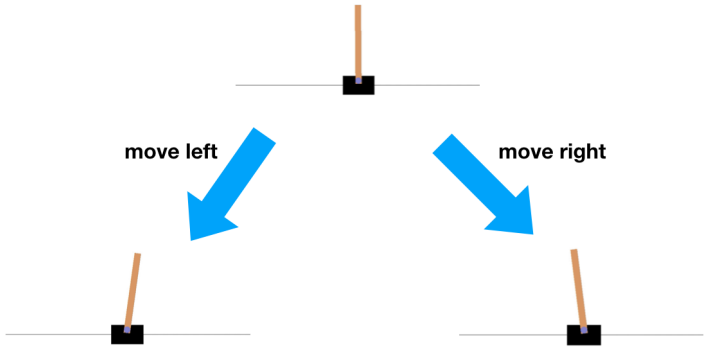
$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, ..., N-1$$

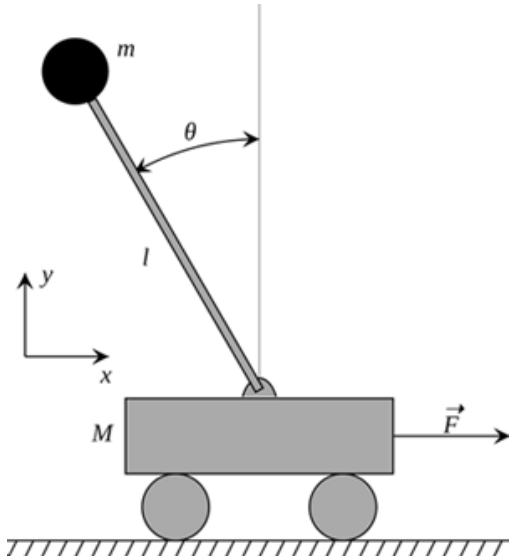
- Cost function:

$$\mathbb{E}\left[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k, w_k)\right]$$

- Optimization over feedback policies $\{\mu_0, ..., \mu_{N-1}\}$: Rules that specify the control $\mu_k(x_k)$ to apply at each possible state x_k that can occur

Example: Inverted Pendulum



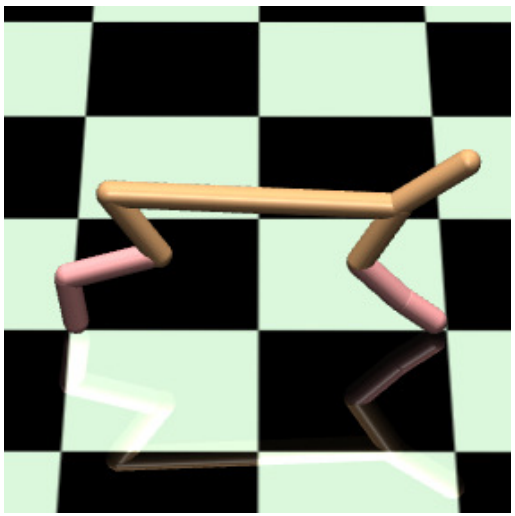


- x: position along the x-axis
- θ: angle of the pendulum

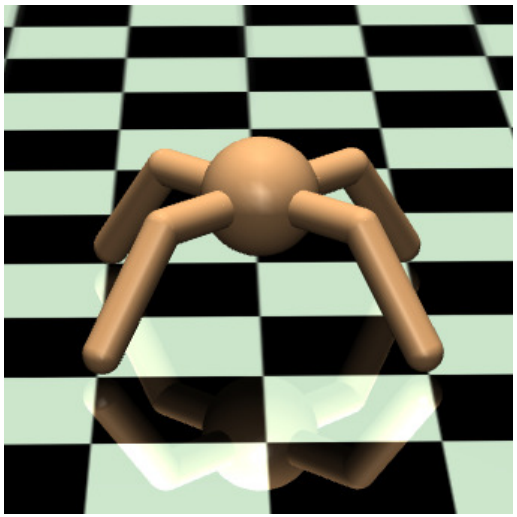
- State: $(x, \dot{x}, \theta, \dot{\theta})$
- Action: $\{-1, 0, +1\}$ or $[-1, +1]$
- Reward: $+1$ if stand up, -0.01 otherwise

Example: Robot locomotion

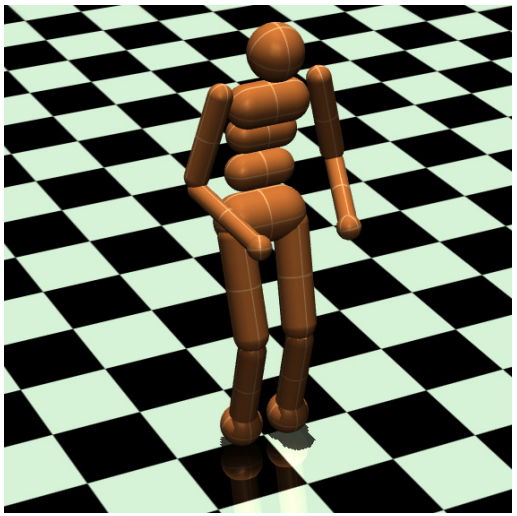
- Make the robots run forward, or navigation
- Observations/states: positions, velocities, angular velocities
- Actions: torques at the joint
- Rewards: velocities, goal



Half Cheetah



Ant

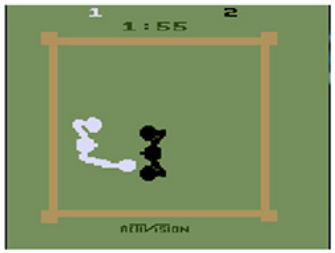


Humanoid

Example: Atari games

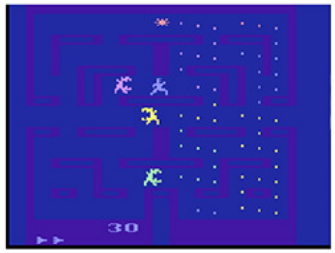
- Observations/states: raw images
- Actions: control signals
- Rewards: win/lose


Space Invaders



Boxing

Skiing

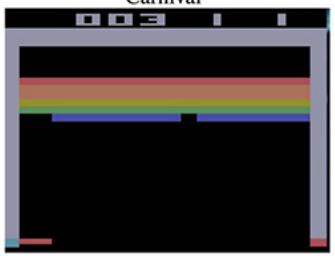

Alien

Carnival


Pong


River Raid


Montezuma's Revenge


Breakout

Kung-Fu Master


Enduro


Ms. PacMan

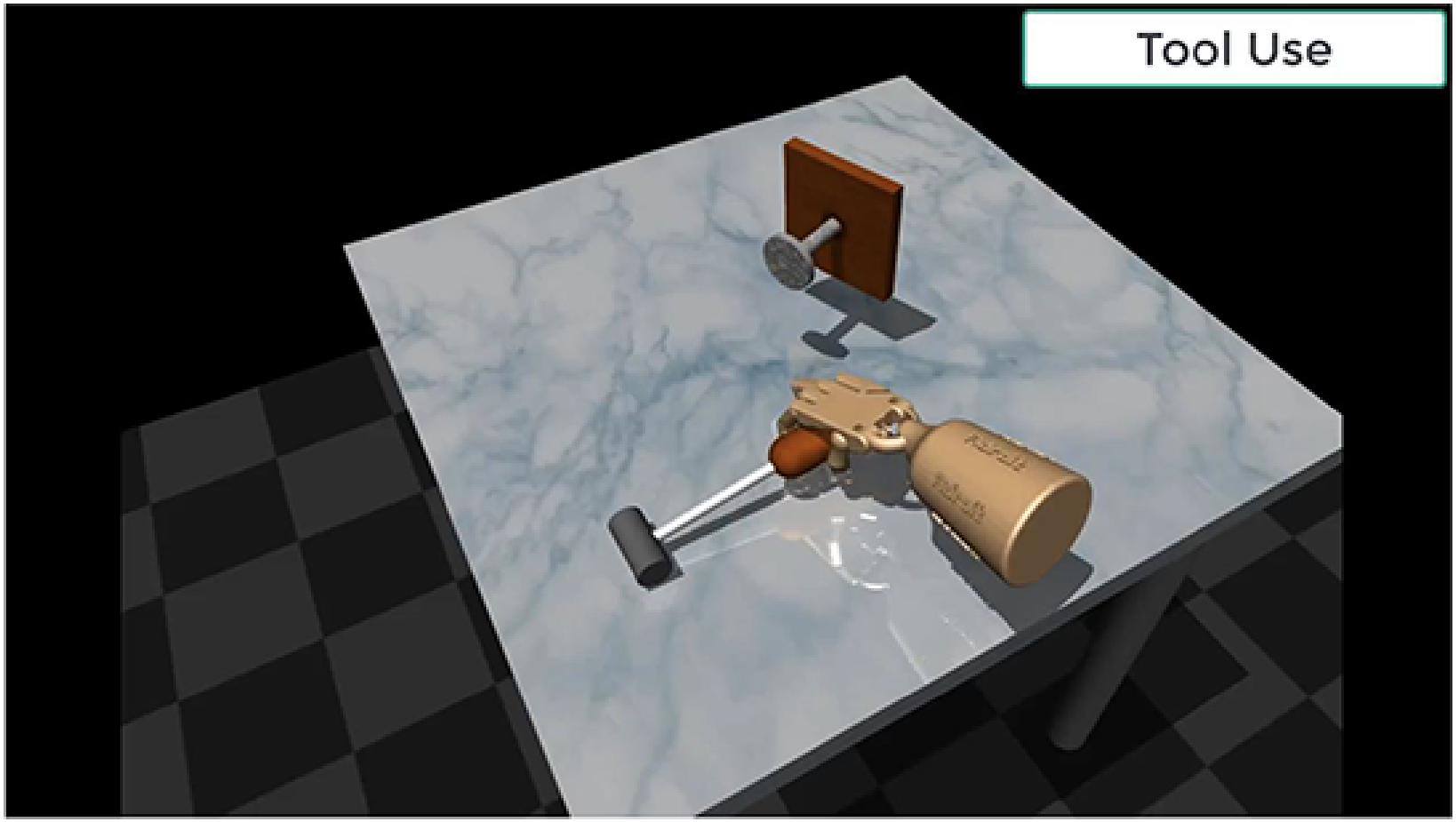
In general...

Actions: muscle contractions
Observations: sight, smell
Rewards: food

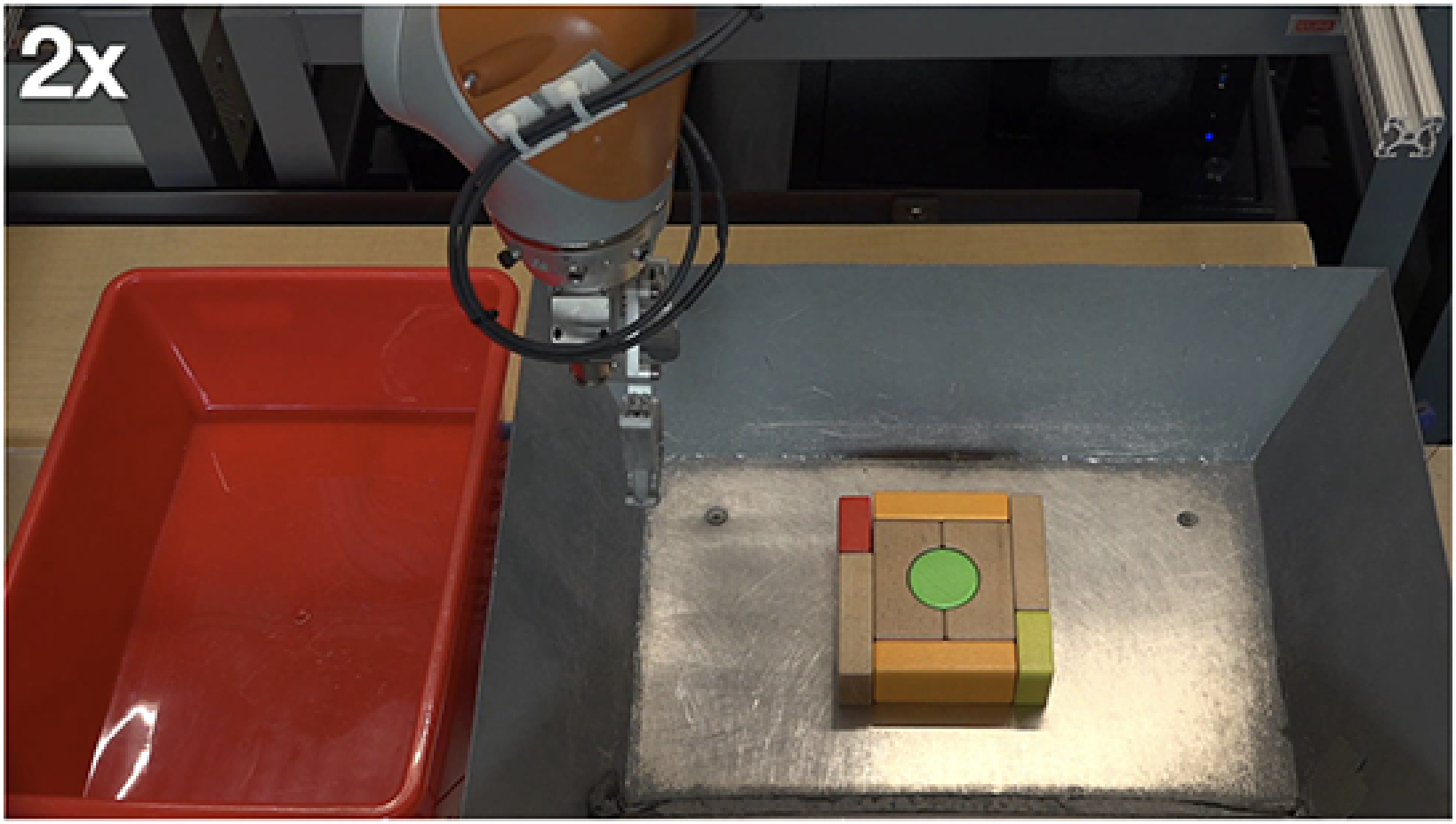
Actions: motor current or torque
Observations: camera images
Rewards: task success measure (e.g., running speed)

Actions: what to purchase
Observations: inventory levels
Rewards: profit

Complex physical tasks



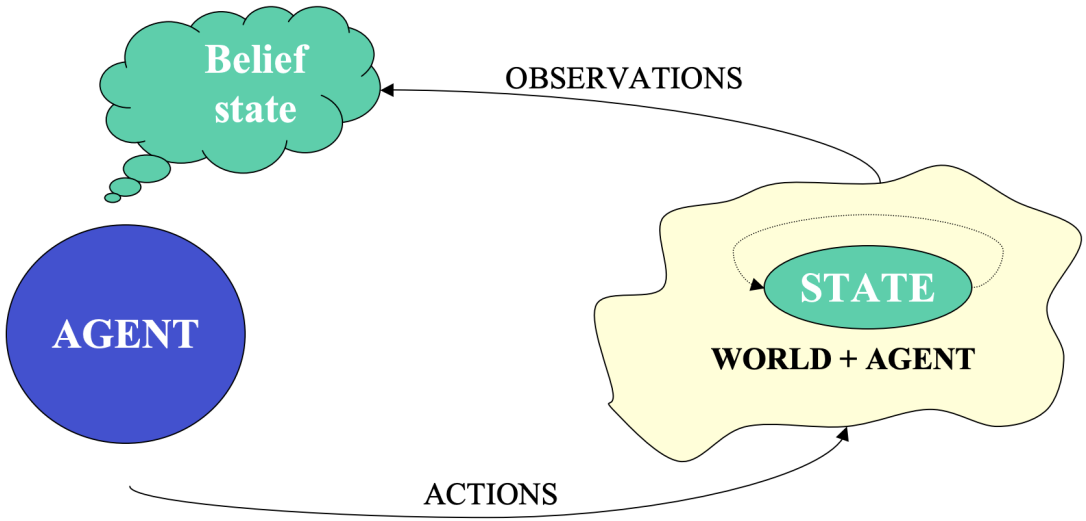
Real-world scenarios



Partially Observable MDP (POMDP)

POMDP: Uncertainty

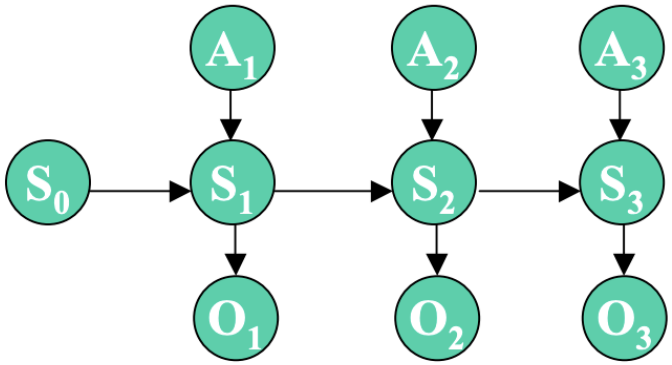
- Case 1: Uncertainty about the action outcome
- Case 2: Uncertainty about the world state due to imperfect (partial) information



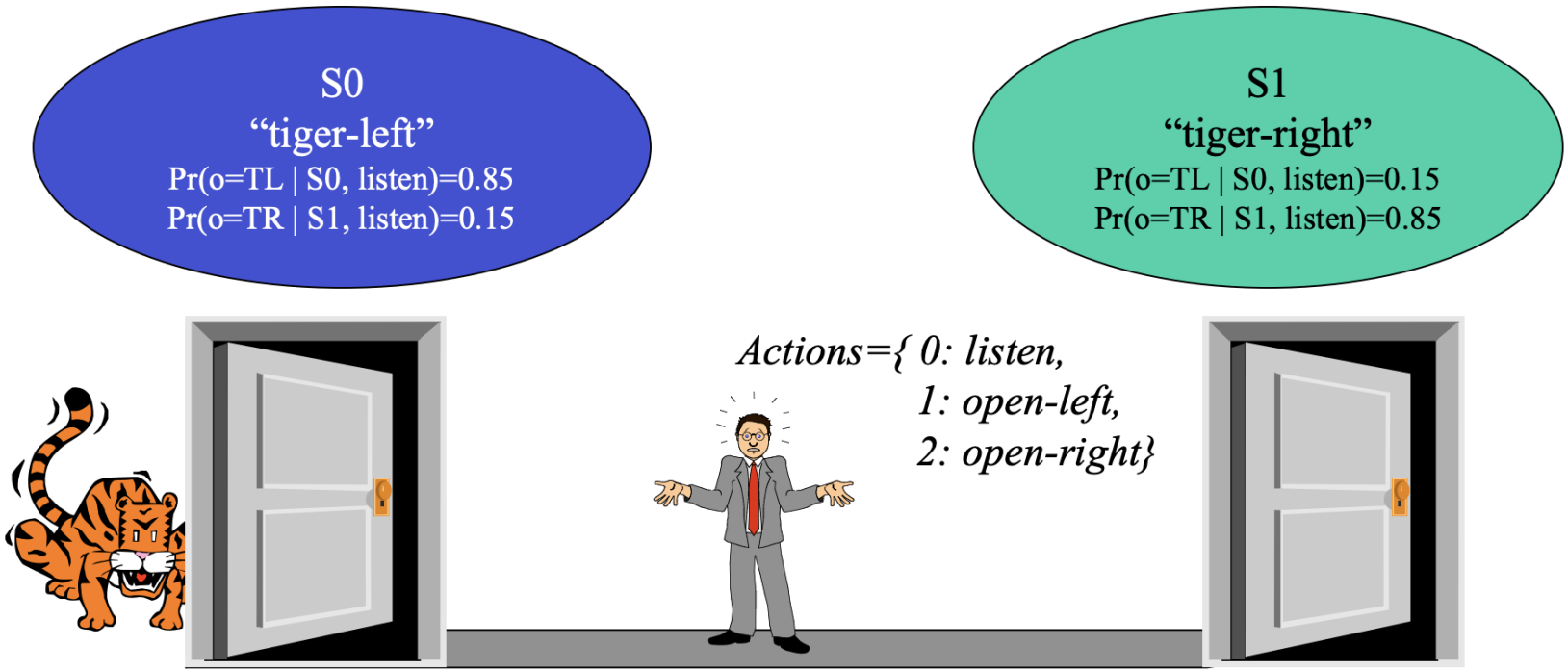
GOAL = Selecting appropriate actions

Partially Observable MDP (POMDP)

- A generalization of an MDP
  - the agent cannot directly observe the underlying state
  - it must maintain a probability distribution over the set of possible states, based on a set of observations and observation probabilities
- $M = \langle S, A, T, R, \Omega, O \rangle$
  - $\Omega = \{o_1, ..., o_k\}$ is a set of observations
  - $O(o|s', a)$ is a set of conditional observation probabilities



A POMDP example: the tiger problem



Reward Function

- *Penalty for wrong opening: -100*
- *Reward for correct opening: +10*
- *Cost for listening action: -1*

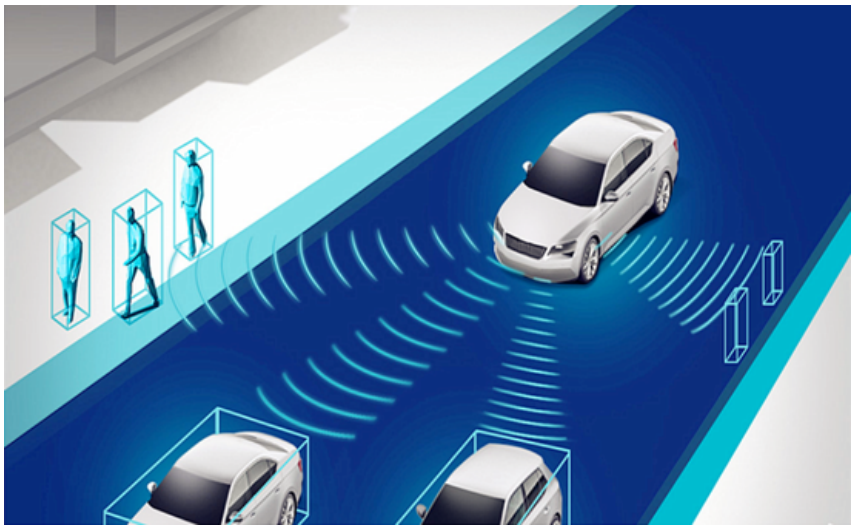
Observations

- *to hear the tiger on the left (TL)*
- *to hear the tiger on the right(TR)*

Table of Contents

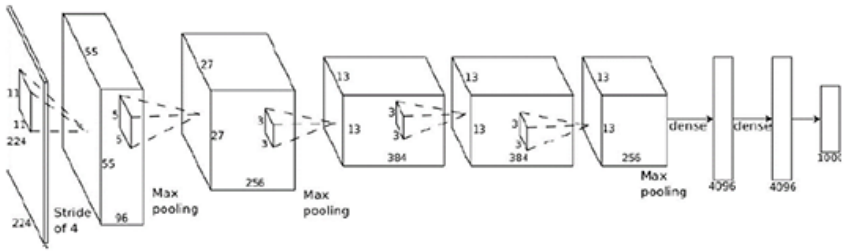
Intelligent machines

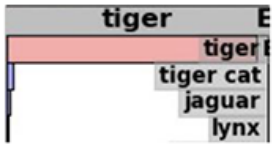
Intelligent machines must able to adapt

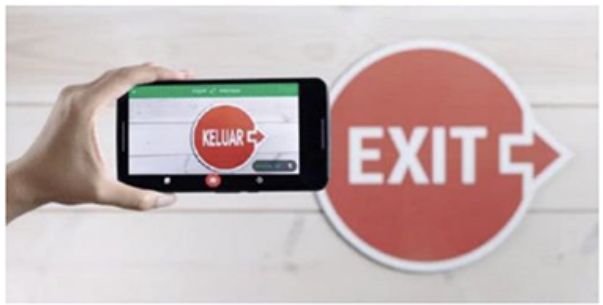


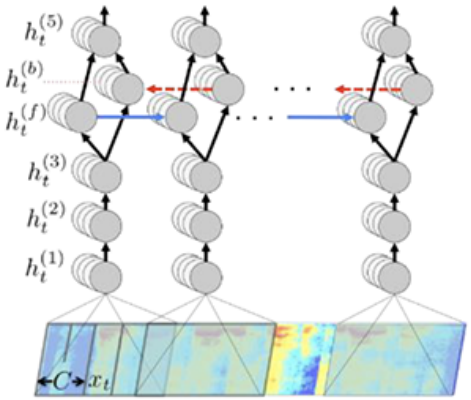
Deep learning

- Deep learning helps us handle *unstructured environments*



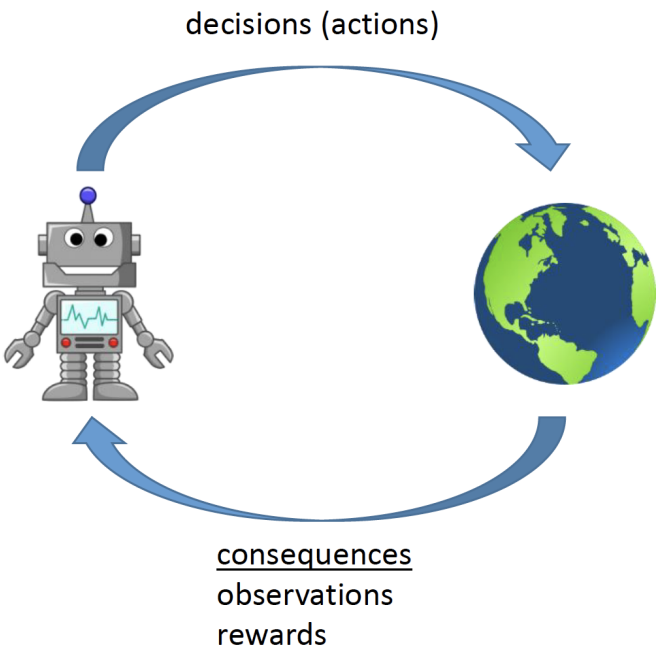


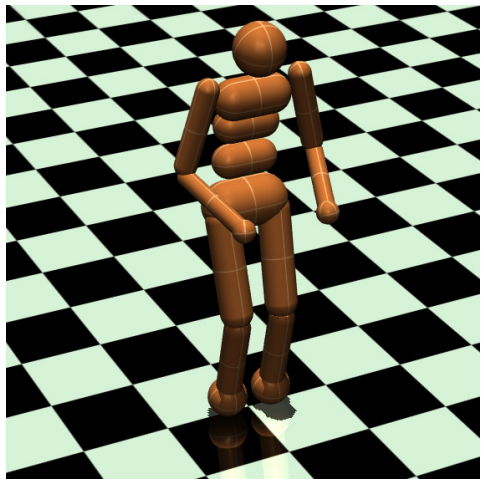


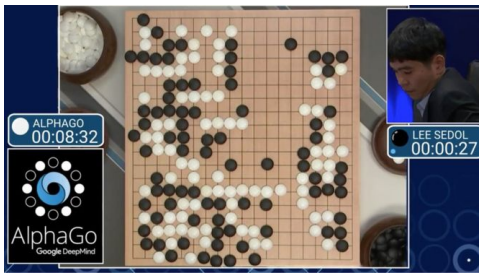


RL

- RL provides a formalism for behavior, decision-making

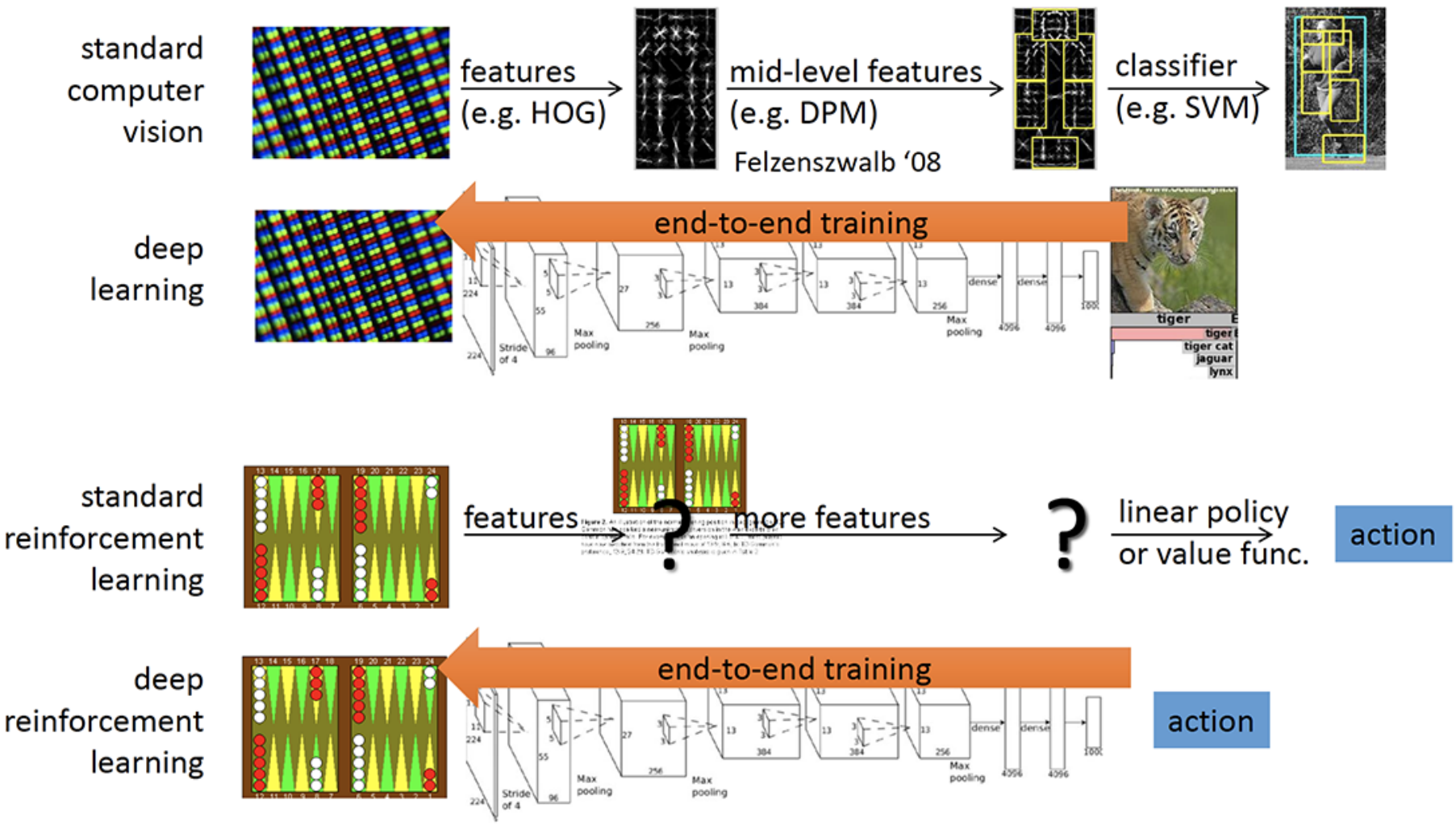




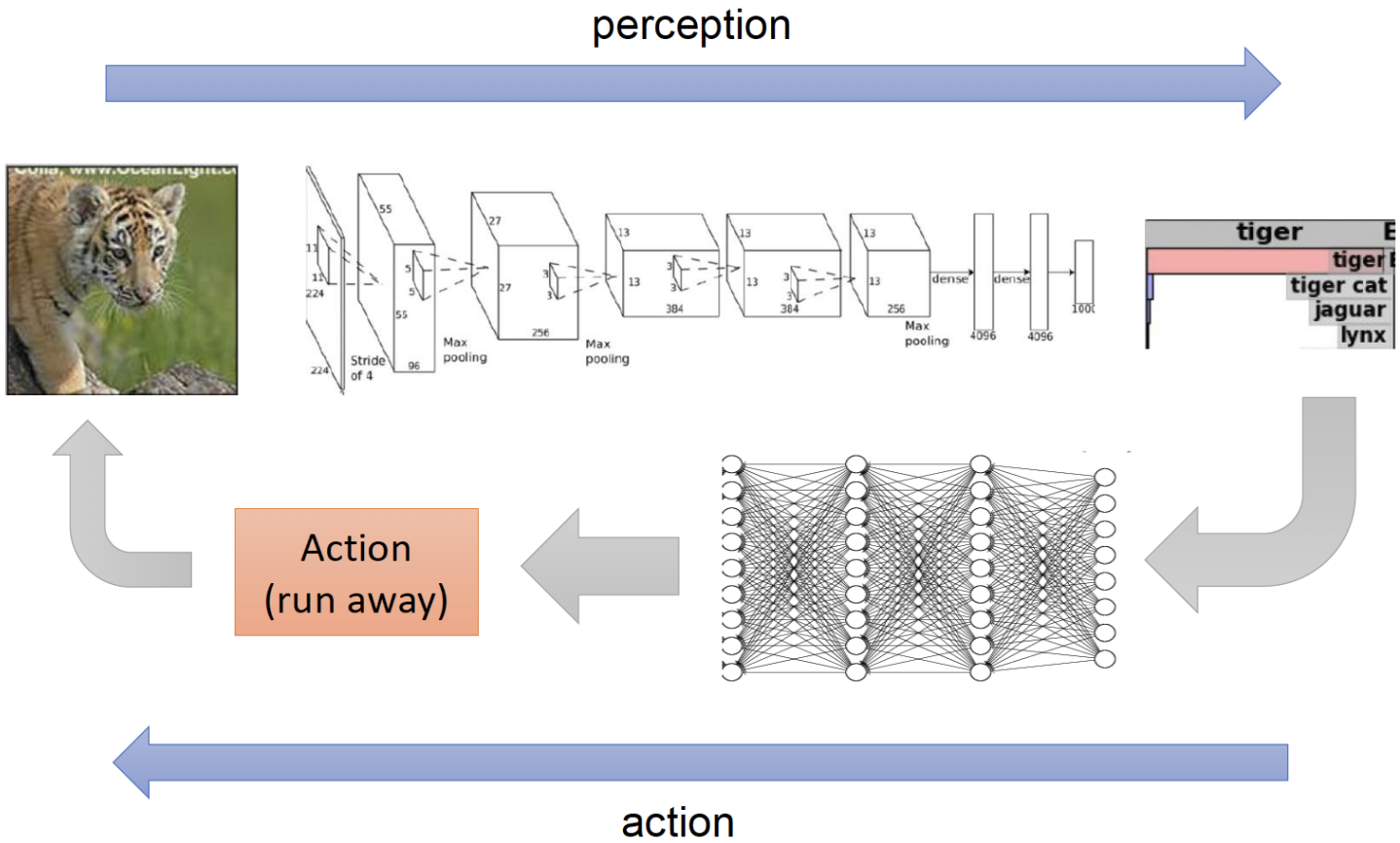


Space Invaders

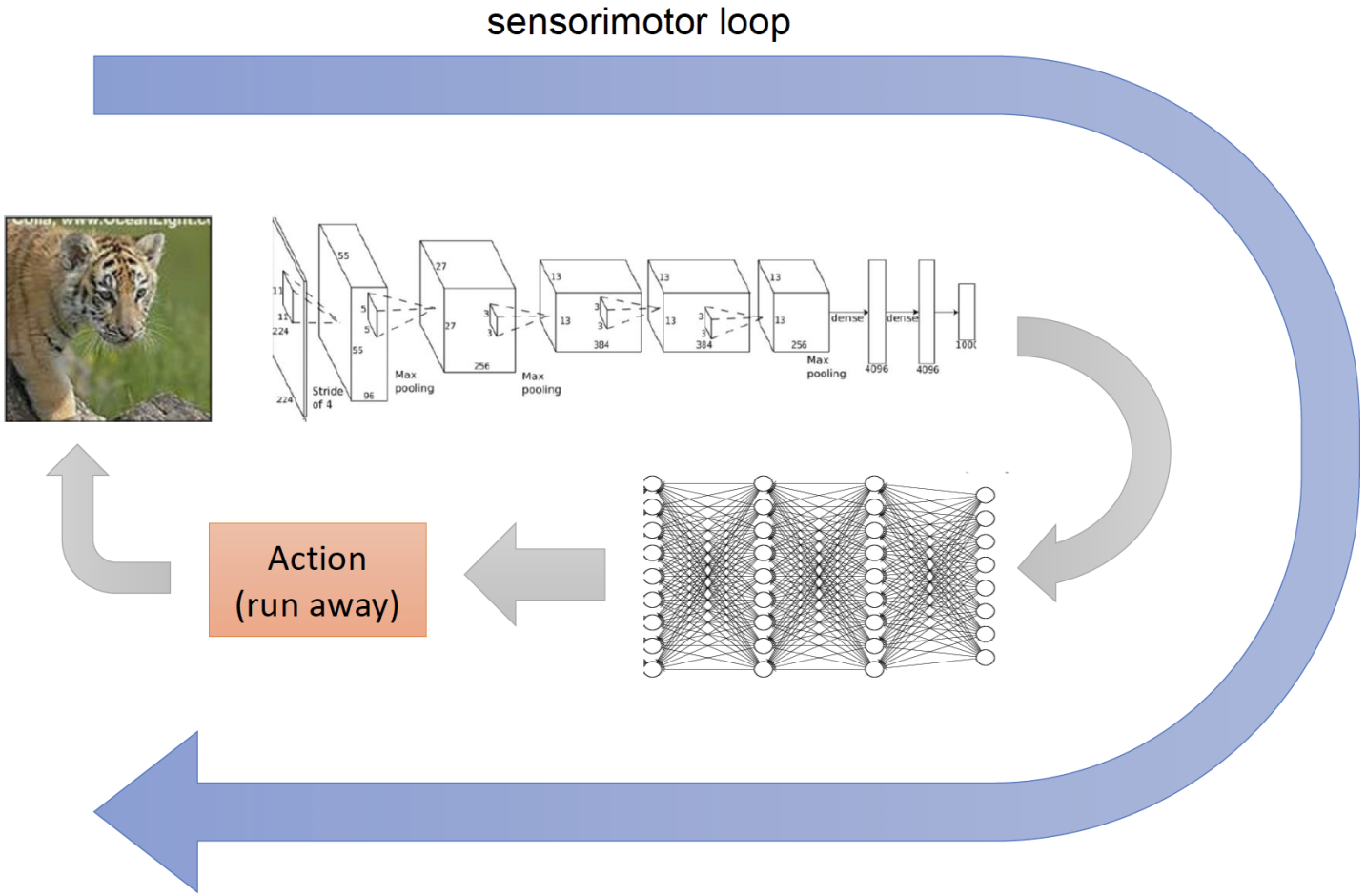
Deep RL = deep learning + RL



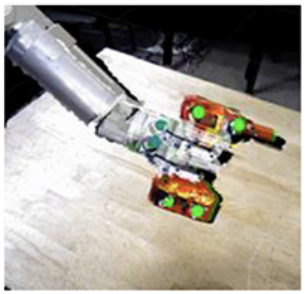
End-to-end learning for sequential decision-making

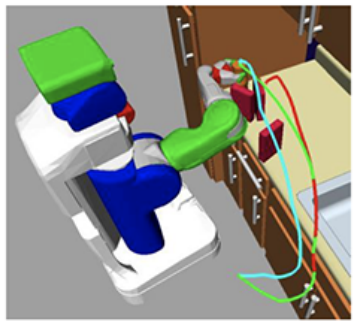


End-to-end learning for sequential decision-making

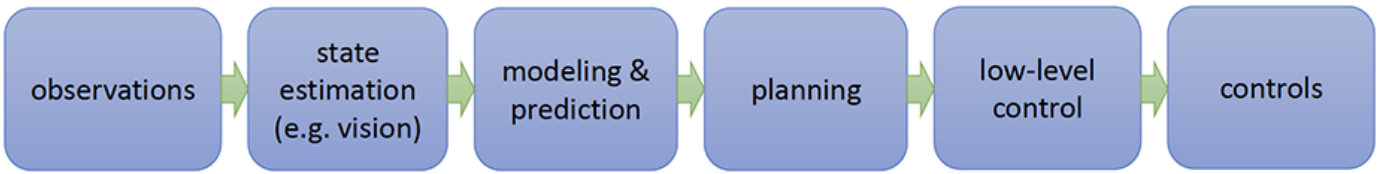


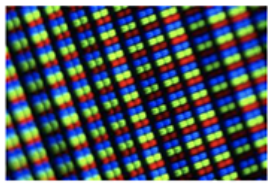
Example: Robotics

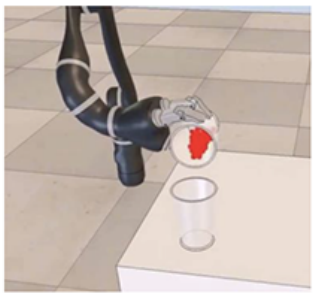


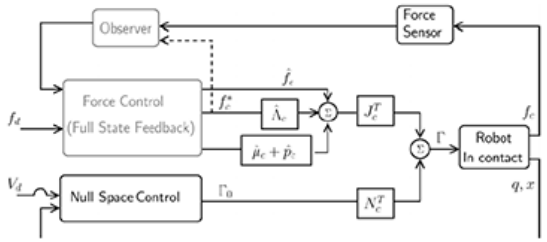


robotic control pipeline

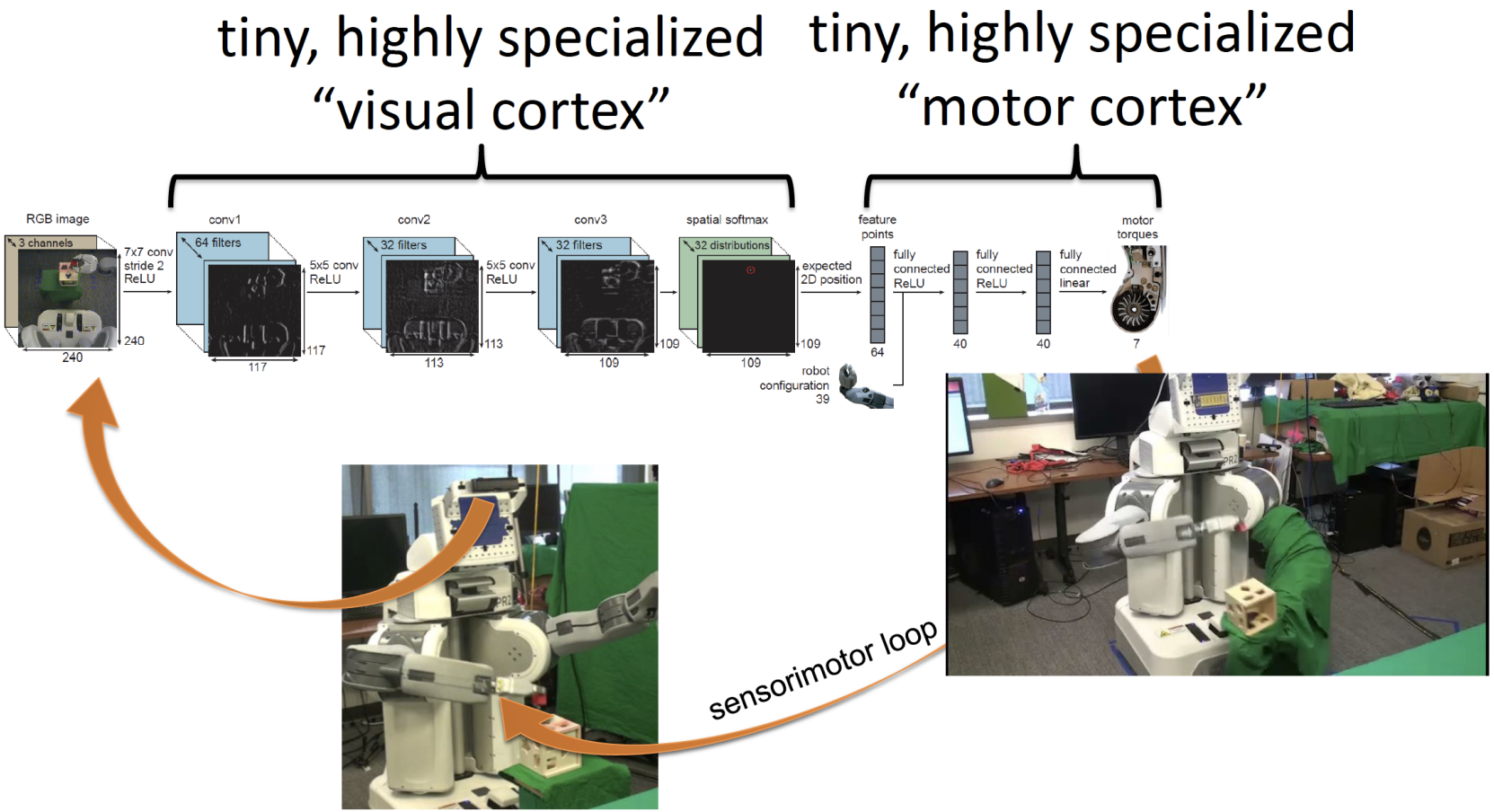




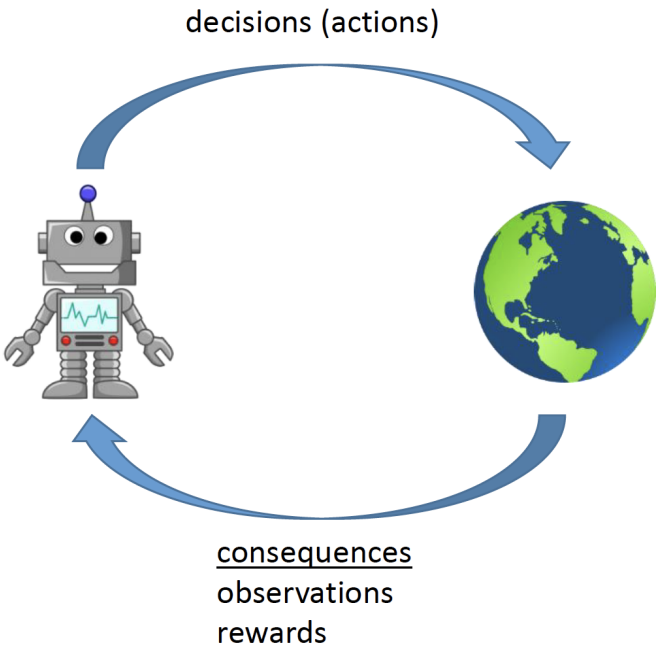


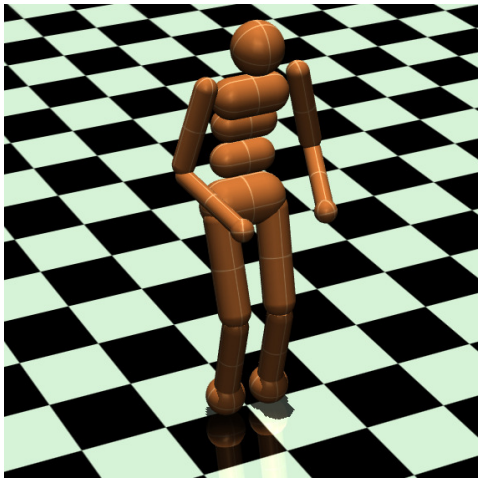


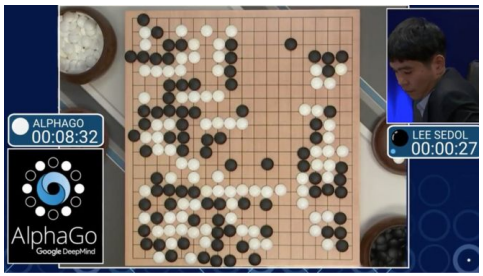
End-to-end learning



The RL problem is the AI problem





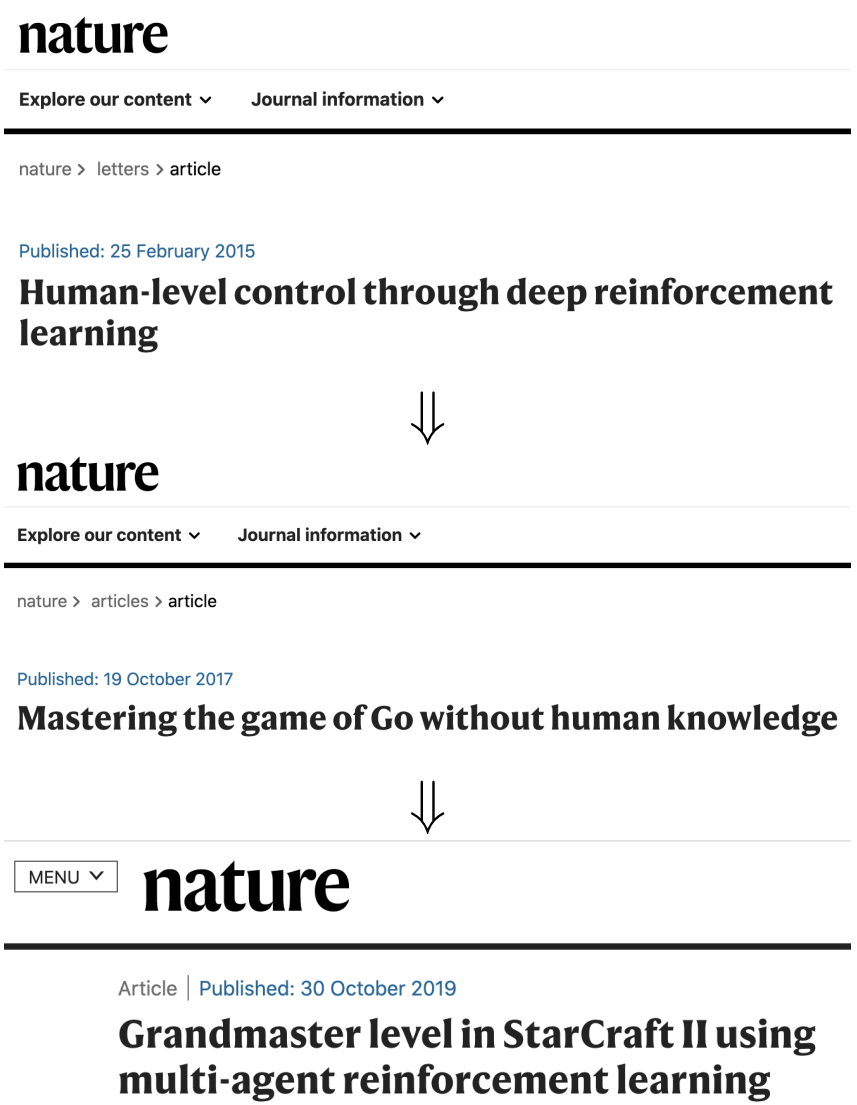


Space Invaders

- Deep models are what allow RL algorithms to solve complex problems end-to-end!

Why should we study this now?

- Advances in deep learning
- Advances in RL
- Advances in computational capability



Why should we study this now?


Atari games:

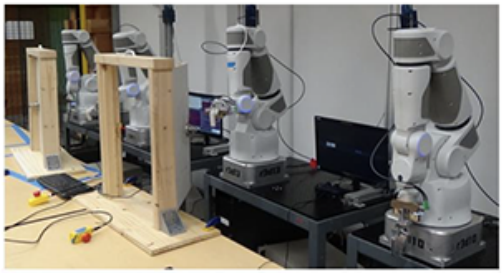
Q-learning:

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:

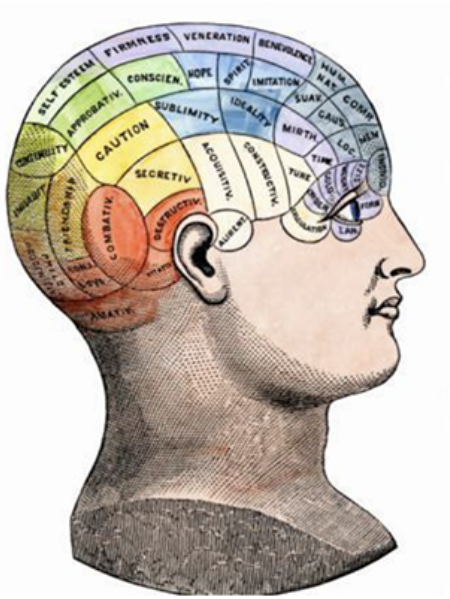
Supervised learning + policy gradients + value functions + Monte Carlo tree search:

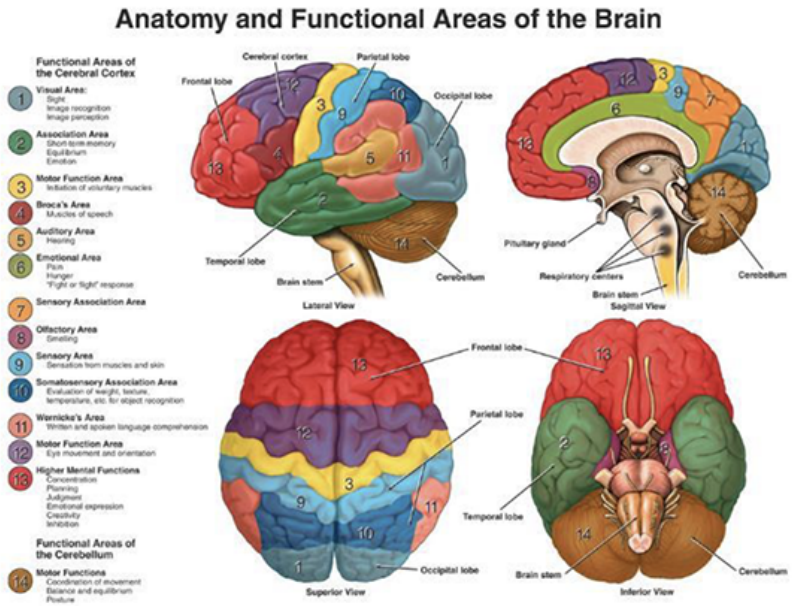
D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, et al. "Mastering the game of Go with deep neural networks and tree search". Nature (2016).

Table of Contents

How do we build intelligent machines?

- Imagine you have to build an intelligent machine, where do you start?



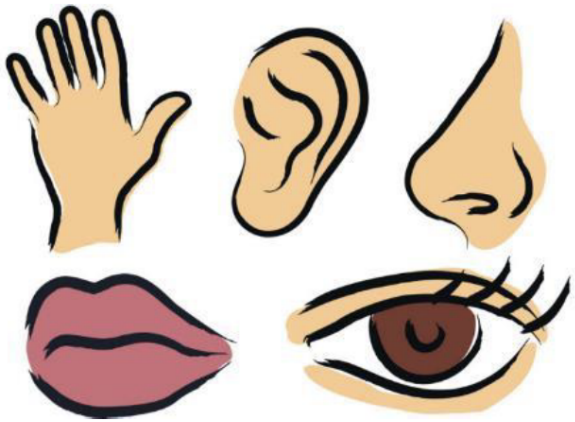


Learning as the basis of intelligence

- Some things we can all do (e.g. walking)
- Some things we can only learn (e.g. driving a car)
- We can learn a huge variety of things, including very difficult things
- Therefore, our learning mechanism(s) are likely powerful enough to do everything we associate with intelligence
    - But it may still be very convenient to "hard code" a few really important bits

What must a single algorithm do?

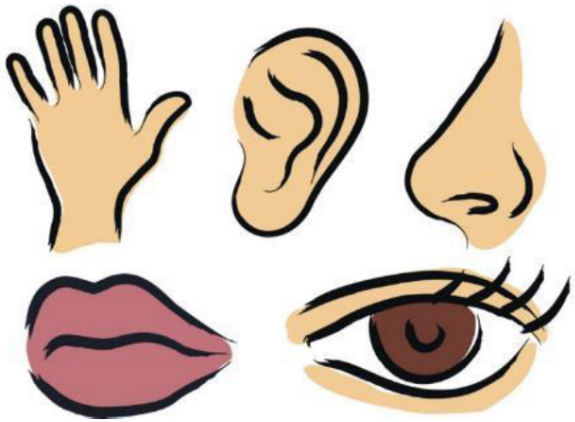
- Interpret (rich) sensory inputs



- Choose (complex) actions

Why (deep) RL?

- Deep = can process complex sensory input
  - and also compute really complex functions



- RL = can choose complex actions

Some evidence for RL

Reinforcement learning in the brain

Yael Niv
Psychology Department & Princeton Neuroscience Institute, Princeton University

- Basal ganglia appears to be related to reward system
- Model-free RL-like adaptation is often a good fit for experimental data of animal adaptation (but not always)

Some evidence for RL

Quantum reinforcement learning during human decision-making

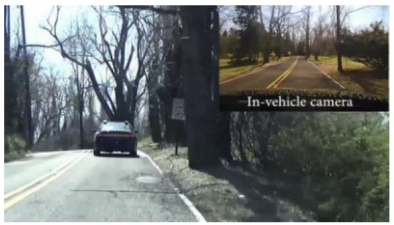
Ji-An Li, Daoyi Dong, Zhengde Wei, Ying Liu, Yu Pan, Franco Nori & Xiaochu Zhang

Nature Human Behaviour **4**, 294–307(2020) | Cite this article

- RL has been widely applied in neuroscience and psychology
- Value-based decision-making can be illustrated by quantum RL at both the behavioral and neural levels

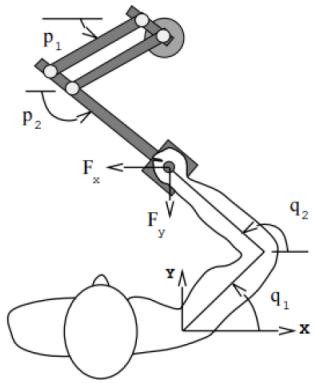
What can (deep) RL do?

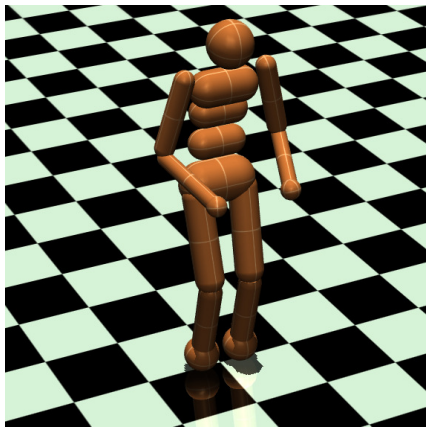
- Acquire high degree of proficiency in domains governed by simple, known rules
- Learn simple skills with raw sensory inputs, given enough experience
- Learn from imitating enough human provided expert behavior



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

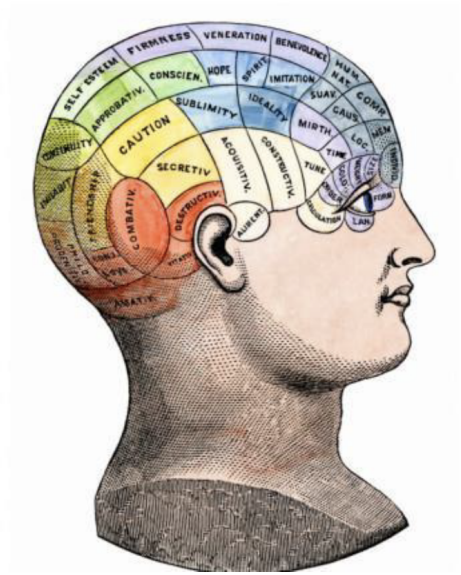




RL, decision making, human behavior

Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.

- Alan Turing



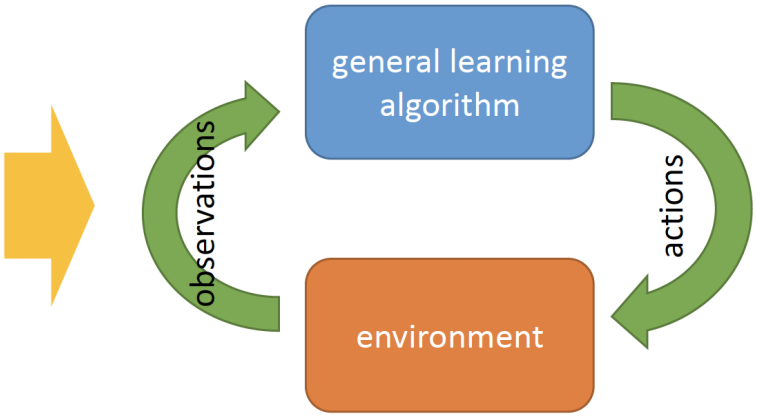
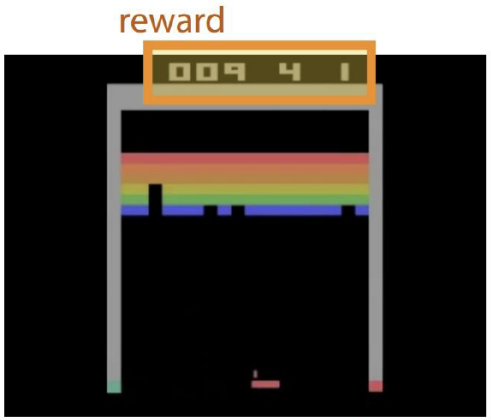


Table of Contents

To enable real-world sequential decision making – Beyond learning from reward

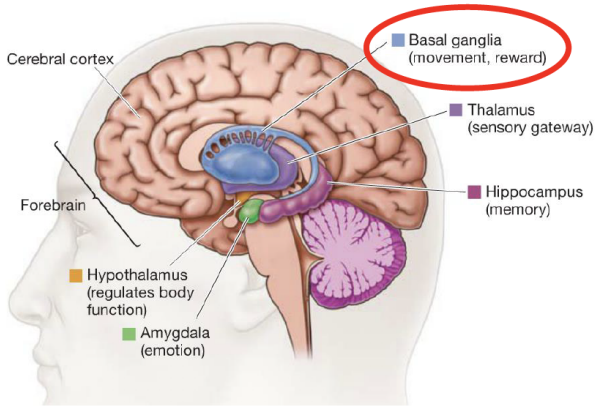
- Basic RL deals with maximizing rewards
- This is not the only problem that matters for sequential decision making!
- More advanced topics
  - Transferring knowledge between domains (transfer learning, meta learning)
  - Learning to predict and using prediction to act (model-based RL)
  - Learning reward functions from example (inverse RL)

What do rewards come from?



Mnih et al. '15

reinforcement learning agent

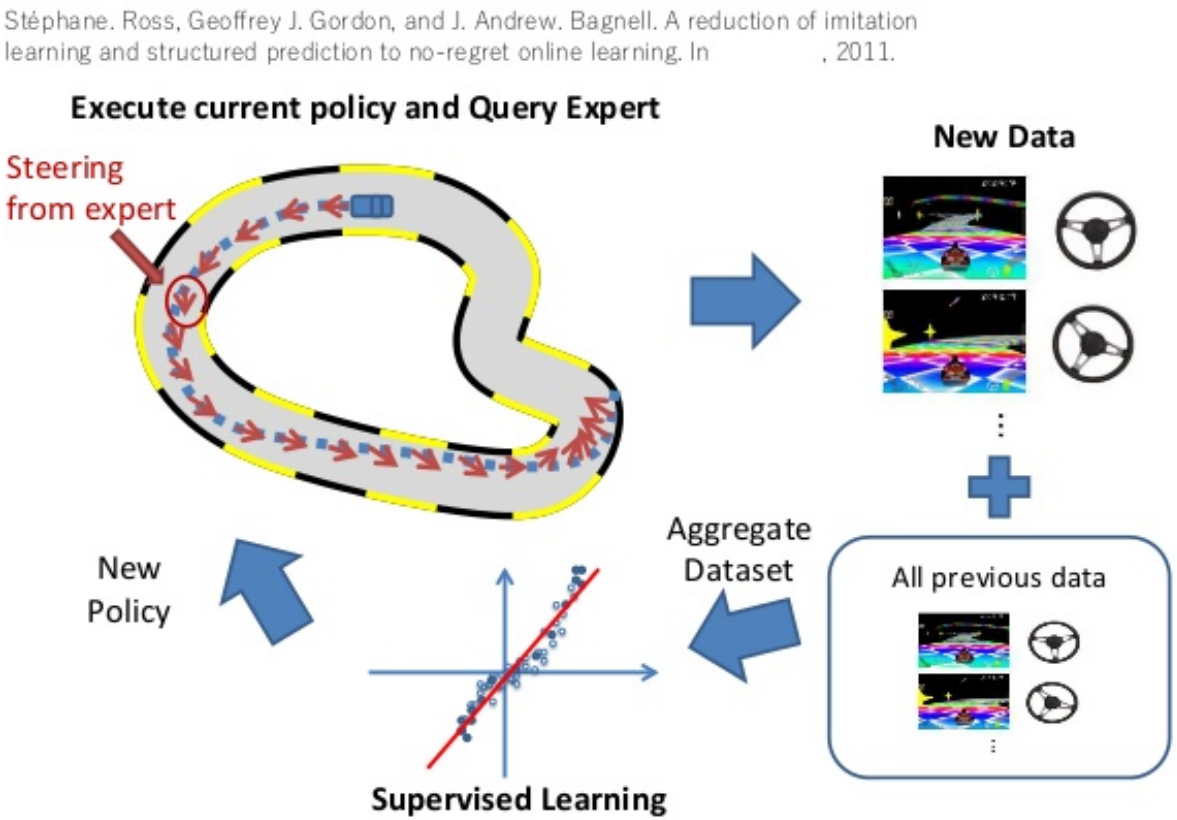


- As human beings, we are accustomed to operating with rewards that are so sparse that we only experience them once or twice in a lifetime, if at all.

Are there other forms of supervision?

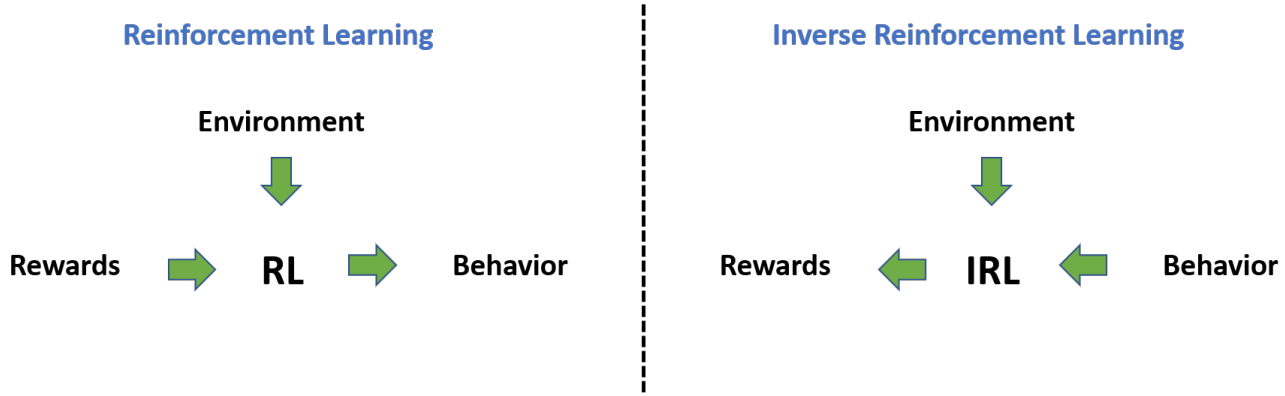
- Learning from demonstrations
  - Directly copying observed behavior
  - Inferring rewards from observed behavior (inverse RL)
- Learning from observing the reward
  - Learning to predict
  - Unsupervised learning
- Learning from other tasks
  - Transfer learning
  - Meta-learning: learning-to-learn

Imitation learning



- Given: demonstrations or demonstrator
- Goal: train a policy to mimic demonstrations

Inverse reinforcement learning



- Using the reward function to find a policy π^*
- Modeling reward can be easier. Simple reward function can lead to complex policy.

Prediction

"the idea that we **predict the consequences of our motor commands** has emerged as an important theoretical concept in all aspects of sensorimotor control"

Prediction Precedes Control in Motor Learning

J. Randall Flanagan,[1,*] Philipp Vetter,[2] Roland S. Johansson,[3] and Daniel M. Wolpert[2]

Procedures for details). Figure 1 shows, for a single subject, the hand path (top trace) and the grip (middle)

Predicting the Consequences of Our Own Actions: The Role of Sensorimotor Context Estimation

Sarah J. Blakemore, Susan J. Goodbody, and Daniel M. Wolpert

Sobell Department of Neurophysiology, Institute of Neurology, University College London, London WC1N 3BG,

Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects

Rajesh P. N. Rao[1] and Dana H. Ballard[2]

Learning objectives of this lecture

You should be able to...

- Understand the basic concepts about sequential decision-making, the differences between supervised learning and RL
- How RL provides a formalism for behavior and decision-making

References

- Lectures 1 of CS285 at UC Berkeley, *Deep Reinforcement Learning, Decision Making, and Control*
  - `http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-1.pdf`

THE END