Carnegie Mellon
School of Computer Science

Deep Reinforcement Learning and Control

Introduction to Deep Reinforcement Learning and Control

Spring 2019, CMU 10-403

Katerina Fragkiadaki

Course Logistics

- Course website : all you need to know is there
- Homework assignments and a final project, 60%/40% for the final grade
- Homework assignments will be both implementation and question/answering
- Final project: a choice between three different topics, e.g., object manipulation, maze navigation or Atari game playing
- Resources: AWS for those that do not have access to GPUs
- Prerequisites: We will assume comfort with deep neural network architectures, modeling and training, using tensorflow or another deep learning package
- People can audit the course, unless there are no seats left in class
- The readings on the schedule are **required**

Goal of the Course: Learning behaviors

Building agents that **learn** to act and accomplish **goals** in **dynamic** environments

Goal of the Course: Learning behaviours

Building agents that **learn** to act and accomplish **goals** in **dynamic** environments

…as opposed to agents that execute **preprogrammed** behaviors in a **static** environment…

Motor control is Important

"The brain evolved, not to think or feel, but to control movement."

Daniel Wolpert, nice TED talk

Daniel Wolpert: The real reason for brains | TED Talk | TED.com

https://www.ted.com/talks/daniel_wolpert_the_real_reason_for_brains

Motor control is Important

The brain evolved, not to think or feel, but to control movement.

Daniel Wolpert, nice TED talk

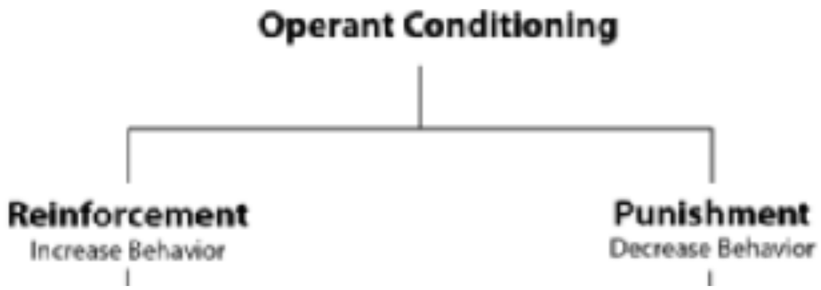
Sea squirts digest their own brain when they decide not to move anymore

Learning behaviours through reinforcement

Behavior is primarily shaped by reinforcement rather than free-will.

- behaviors that result in praise/pleasure tend to repeat,
- behaviors that result in punishment/pain tend to become extinct.

B.F. Skinner
1904-1990
Harvard psychology

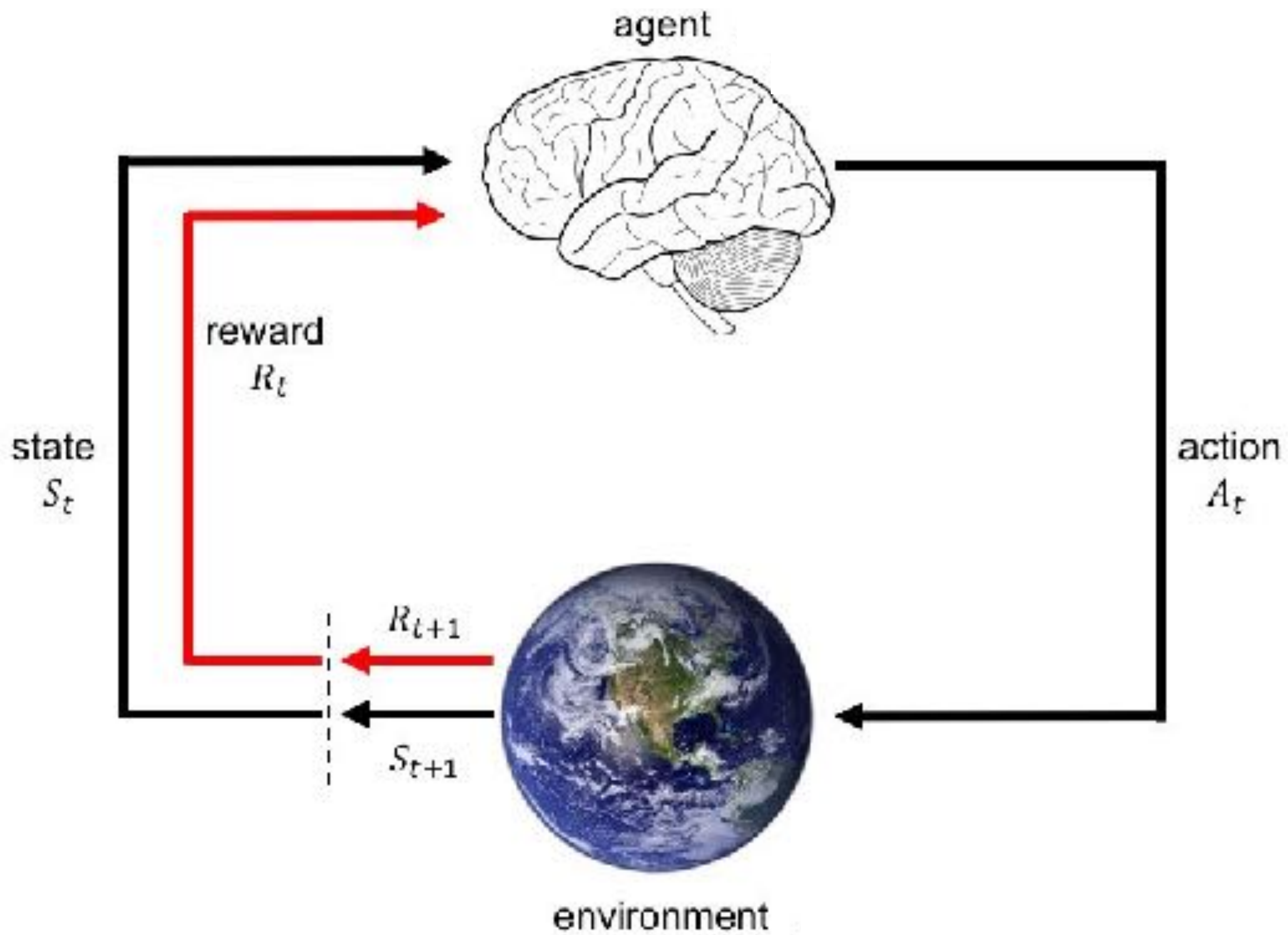


[Video on RL of behaviors in pigeons]

We will use similar shaping mechanism for learning behaviours in artificial agents

[Wikipedia]

Reinforcement learning



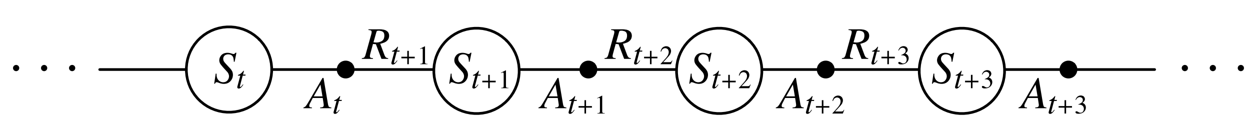
Agent and environment interact at discrete time steps: $t = 0, 1, 2, 3, \ldots$

Agent observes state at step t: $S_t \in \mathcal{S}$

produces action at step t: $A_t \in \mathcal{A}(S_t)$

gets resulting reward: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

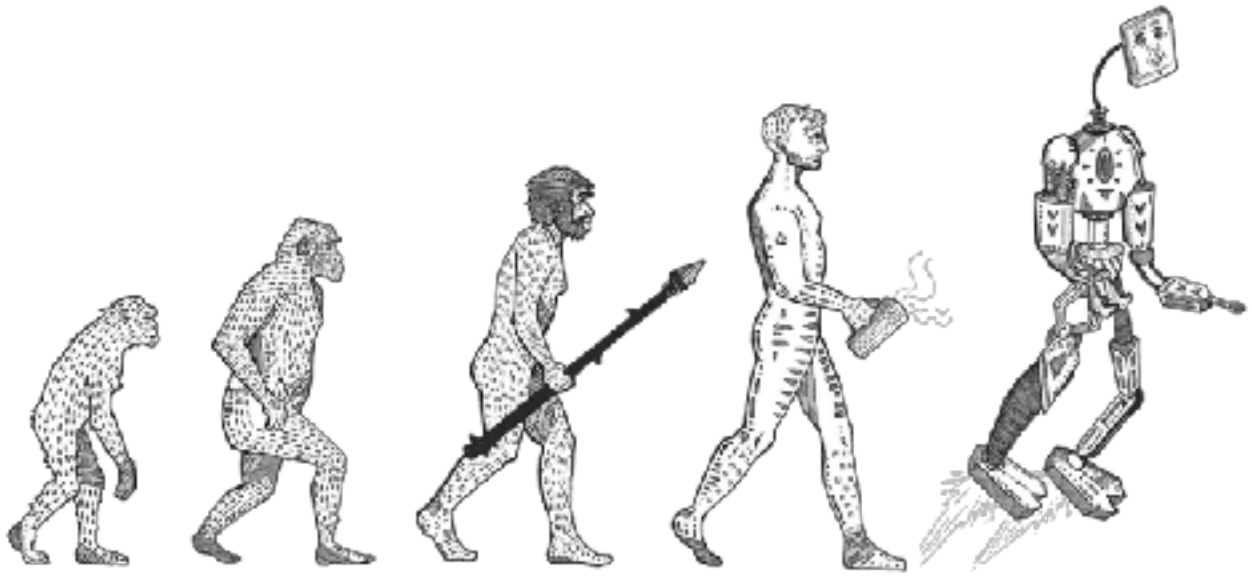
and resulting next state: $S_{t+1} \in \mathcal{S}^+$



Agent

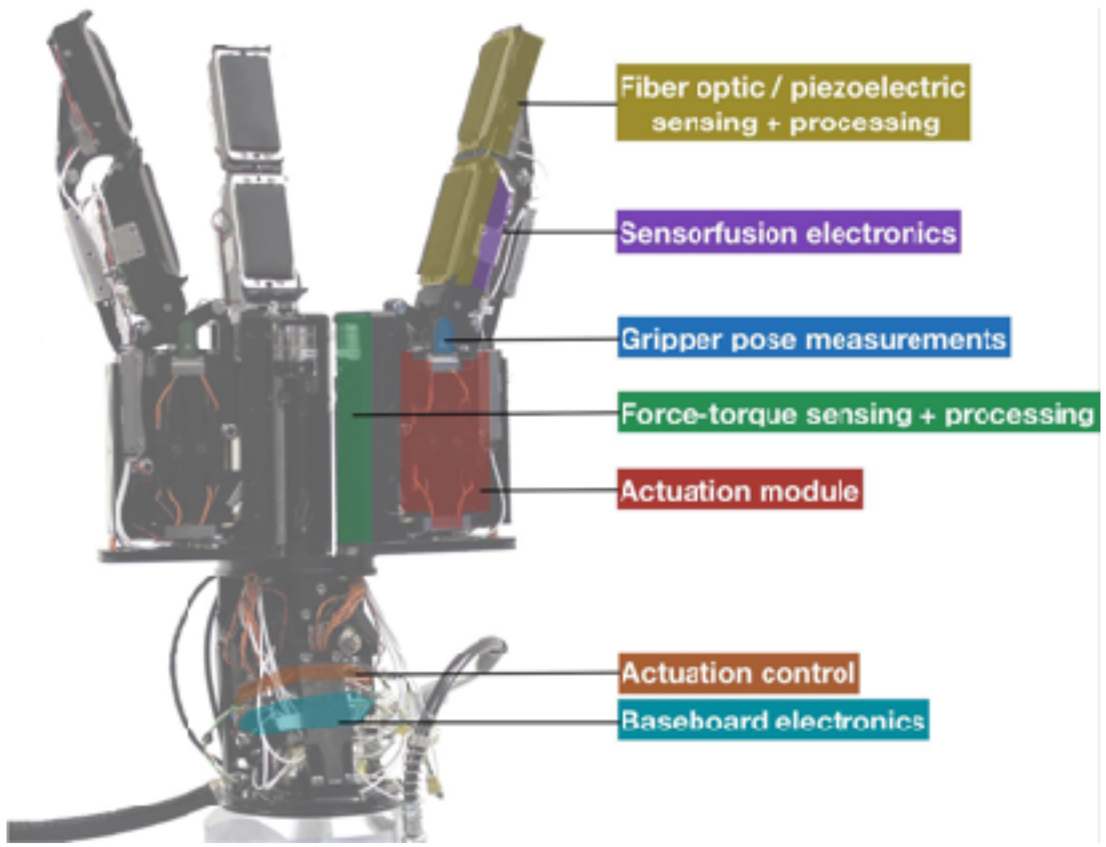
An entity that is equipped with

- **sensors**, in order to sense the environment,
- **end-effectors** in order to act in the environment, and
- **goals** that she wants to achieve



Actions A_t

They are used by the agent to interact with the world. They can have many different temporal granularities and abstractions.



Actions can be defined to be

- The instantaneous torques applied on the gripper
- The instantaneous gripper translation, rotation, opening
- Instantaneous forces applied to the objects
- Short sequences of the above

State estimation: from observations to states

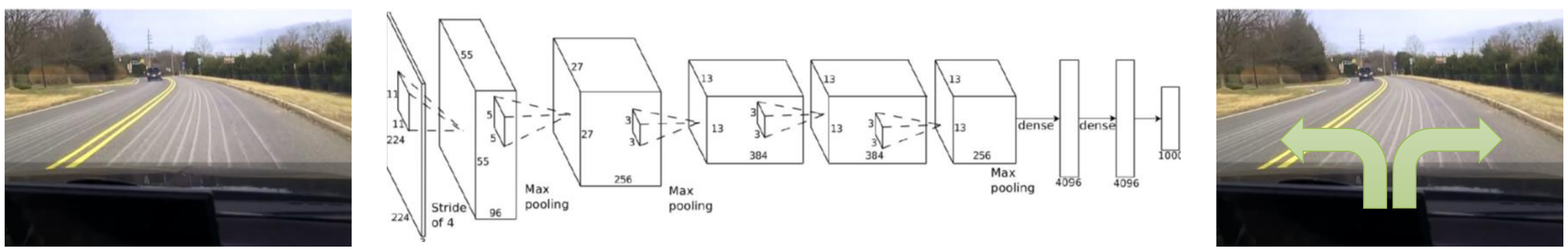
- An observation a.k.a. sensation: the (raw) input of the agent's sensors, images, tactile signal, waveforms, etc.
- A state captures whatever information is available to the agent at step t about its environment. The state can include immediate "sensations," highly processed sensations, and structures built up over time from sequences of sensations, memories etc.

Policy π

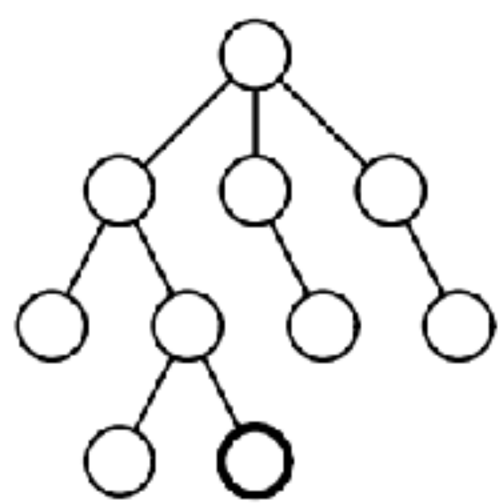
A mapping function from states to actions of the end effectors.

$$\pi(a|s) = \mathbb{P}[A_t = a | S_t = s]$$

It can be a shallow or deep function mapping,

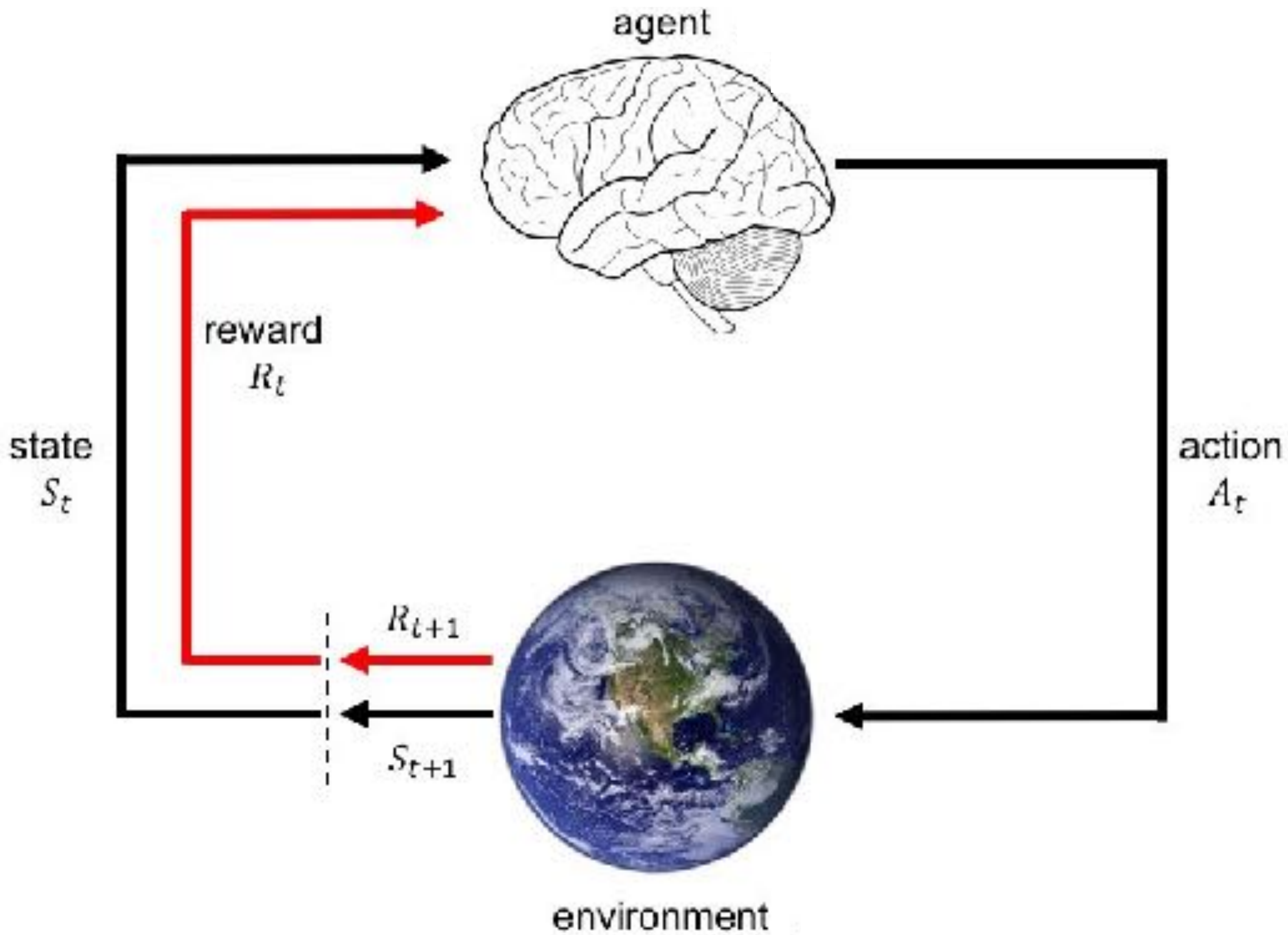


or it can be as complicated as involving a tree look-ahead search



Reinforcement learning

Learning policies that maximize a reward function by interacting with the world

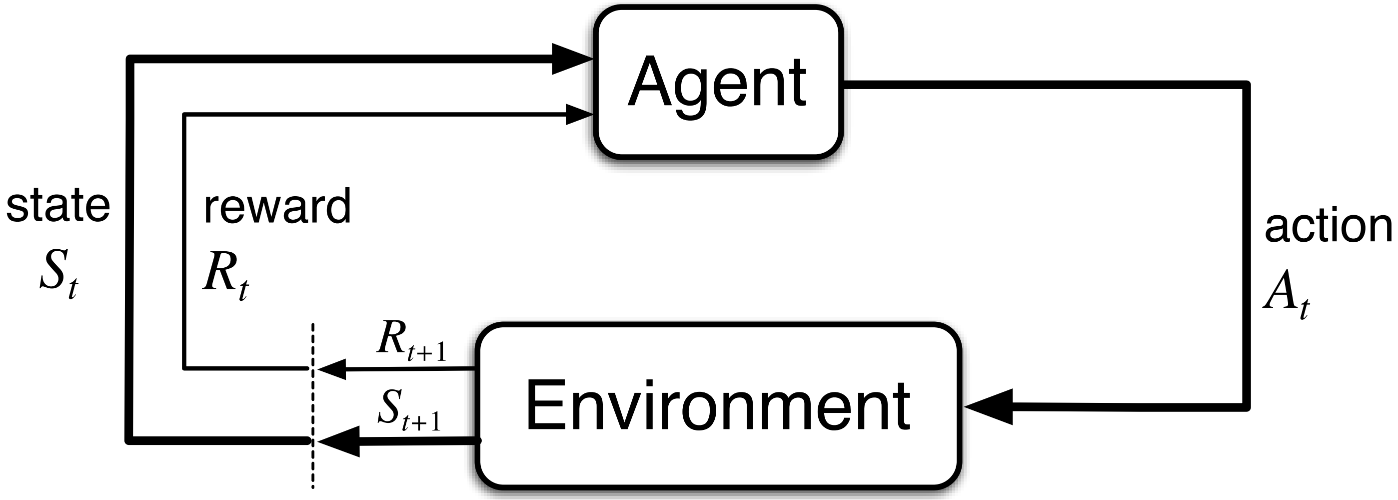


Note: Rewards can be intrinsic, i.e., generated by the agent and guided by its curiosity as opposed to an external task

Closed loop sensing and acting

Imagine an agent that wants to pick up an object and has a policy that predicts what the actions should be for the next 2 secs ahead. This means, for the next 2 secs we switch off the sensors, and just execute the predicted actions. In the next second, due to imperfect sensing, the object is about to fall over!

Sensing is always imperfect. Our excellent motor skills are due to continuous sensing and updating of the actions. So this loop is in fact extremely short in time.



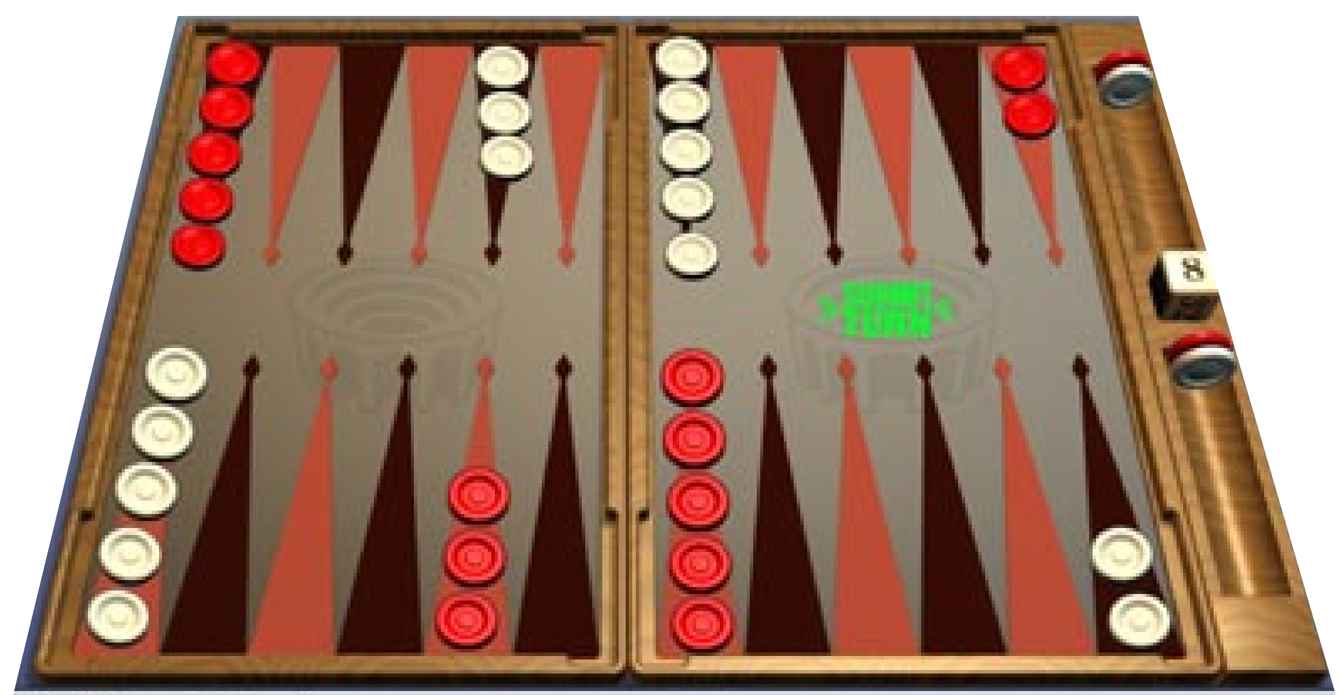
Rewards R_t

They are scalar values provided by the environment to the agent that indicate whether goals have been achieved, e.g., ***1 if goal is achieved, 0 otherwise, or -1 for overtime step the goal is not achieved***

- Rewards specify what the agent needs to achieve, not how to achieve it.
- The simplest and cheapest form of supervision, and surprisingly general: All of what we mean by goals and purposes can be well thought of as the maximization of the cumulative sum of a received scalar signal (reward)

Backgammon

- States: Configurations of the playing board (≈1020)
- Actions: Moves
- Rewards:
  - win: +1
  - lose: –1
  - else: 0

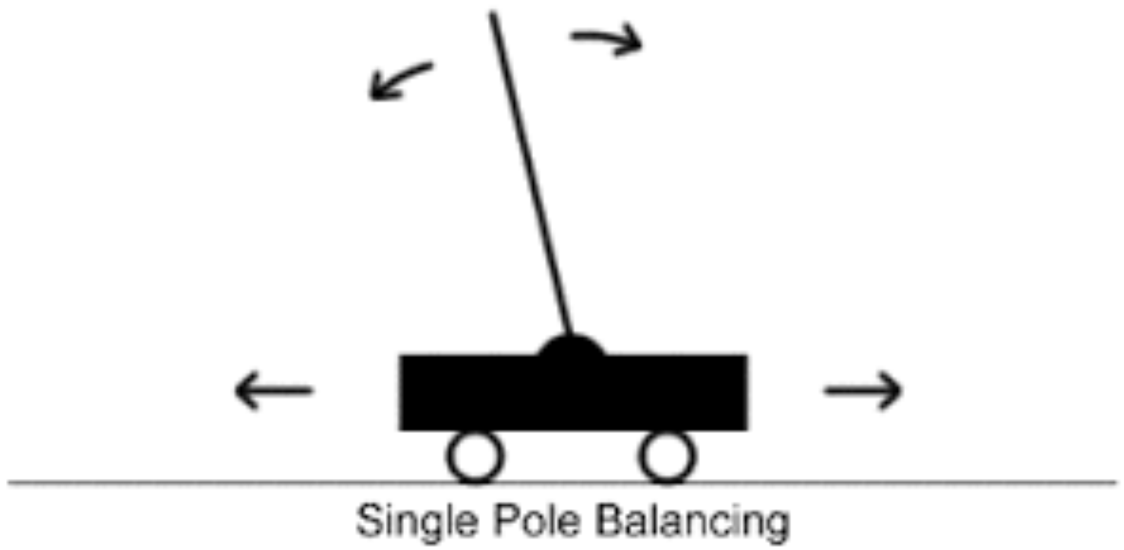


Learning to Drive

- States: Road traffic, weather, time of day
- Actions: steering wheel, break
- Rewards:
  - +1 reaching goal not over-tired
  - -1: honking from surrounding drivers
  - -100: collision

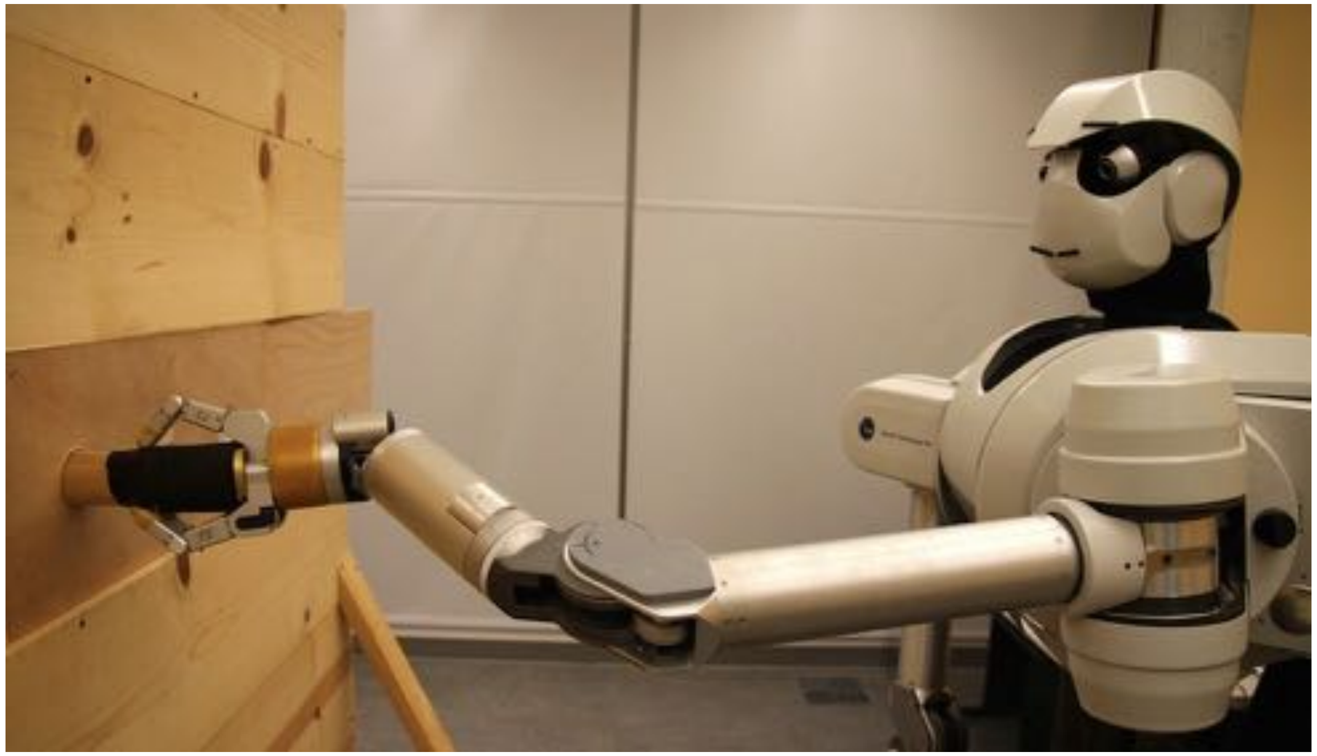
Cart Pole

- States: Pole angle and angular velocity
- Actions: Move left right
- Rewards:
  - 0 while balancing
  - -1 for imbalance



Peg in Hole Insertion Task

- States: Joint configurations (7DOF)
- Actions: Torques on joints
- Rewards: Penalize jerky motions, inversely proportional to distance from target pose



Returns G_t

Goal-seeking behavior of an agent can be formalized as the behavior that seeks maximization of the expected value of the ***cumulative sum of (potentially time discounted) rewards, we call it return.***

We want to maximize returns.

$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T$$

Dynamics p a.k.a. the Model

- How the states and rewards change given the actions of the agent

$$\mathrm{p}(s', r \mid s, a) = \mathbb{P}\{S_t = s', \mathrm{R_t} = r \mid \mathrm{S}_{t-1} = s, \mathrm{A}_{t-1} = a\}$$

- Transition function or next step function:

$$\mathrm{T}(s' \mid s, a) = \mathrm{p}(s' \mid s, a) = \mathbb{P}\{S_t = s' \mid \mathrm{S}_{t-1} = s, \mathrm{A}_{t-1} = a\} = \sum_{r \in \mathbb{R}} \mathrm{p}(s', r \mid s, a)$$

The Model

“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

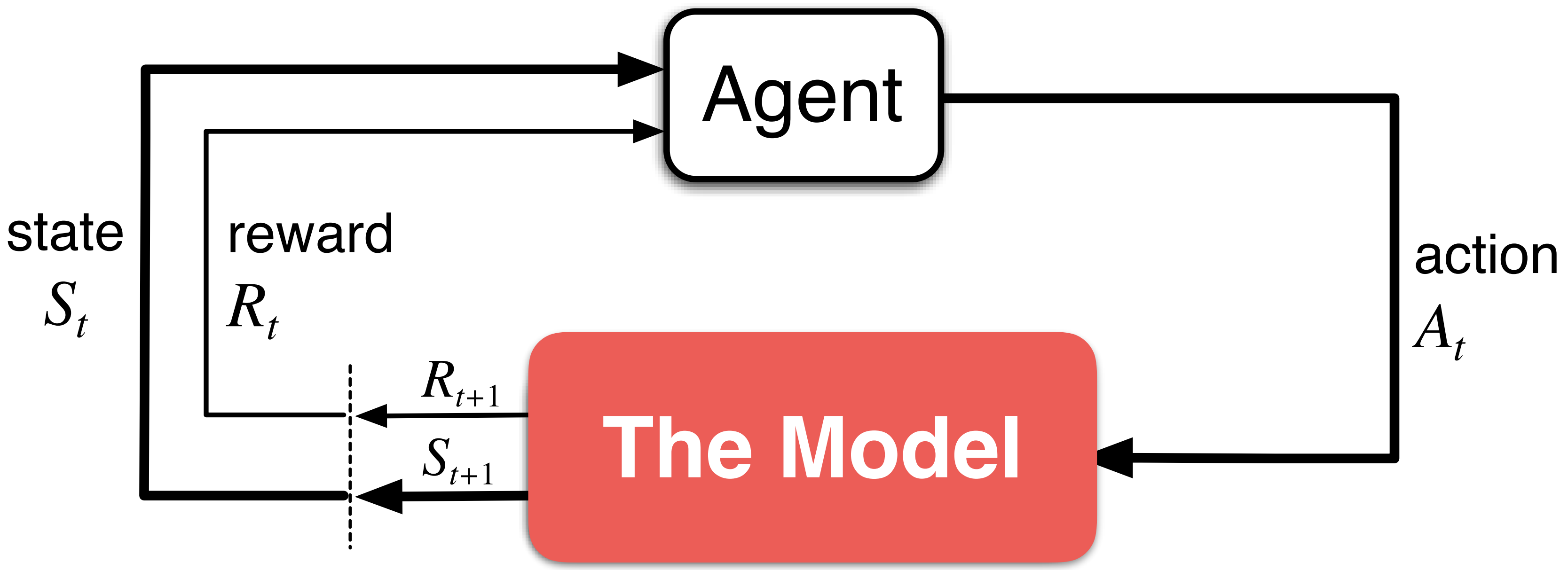
Rajesh P. N. Rao[1] and Dana H. Ballard[2]

slide borrowed from Sergey Levine

Planning

Planning: unrolling (querying) a model forward in time and selecting the best action sequence that satisfies a specific goal

Plan: a sequence of actions



Value Functions are Expected Returns

The *state-value function* $\mathrm{v}_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$\mathrm{v}_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

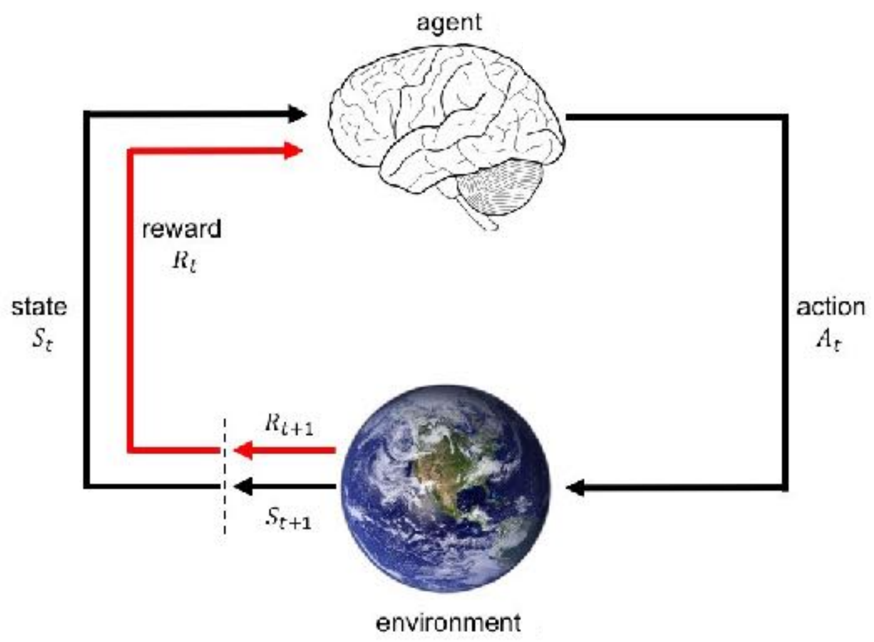
The *action-value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

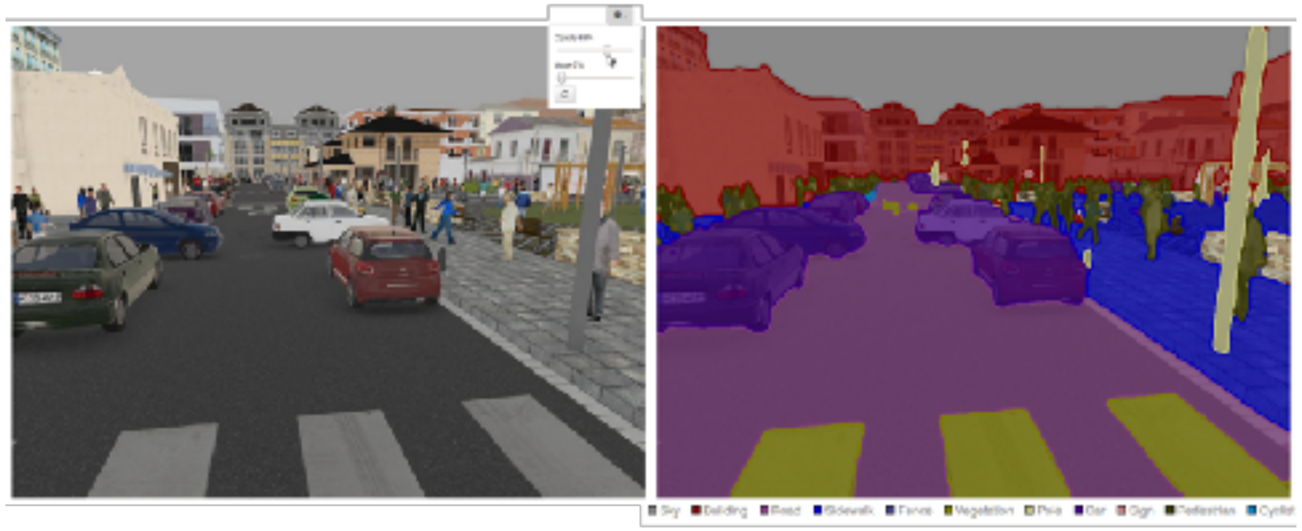
Reinforcement learning-and why we like it

Learning policies that maximize a reward function by interacting with the world

- It is considered the most biologically plausible form of learning
- It addresses the full problem of making artificial agents that act in the world end-to-end, so it is driven by the right loss function

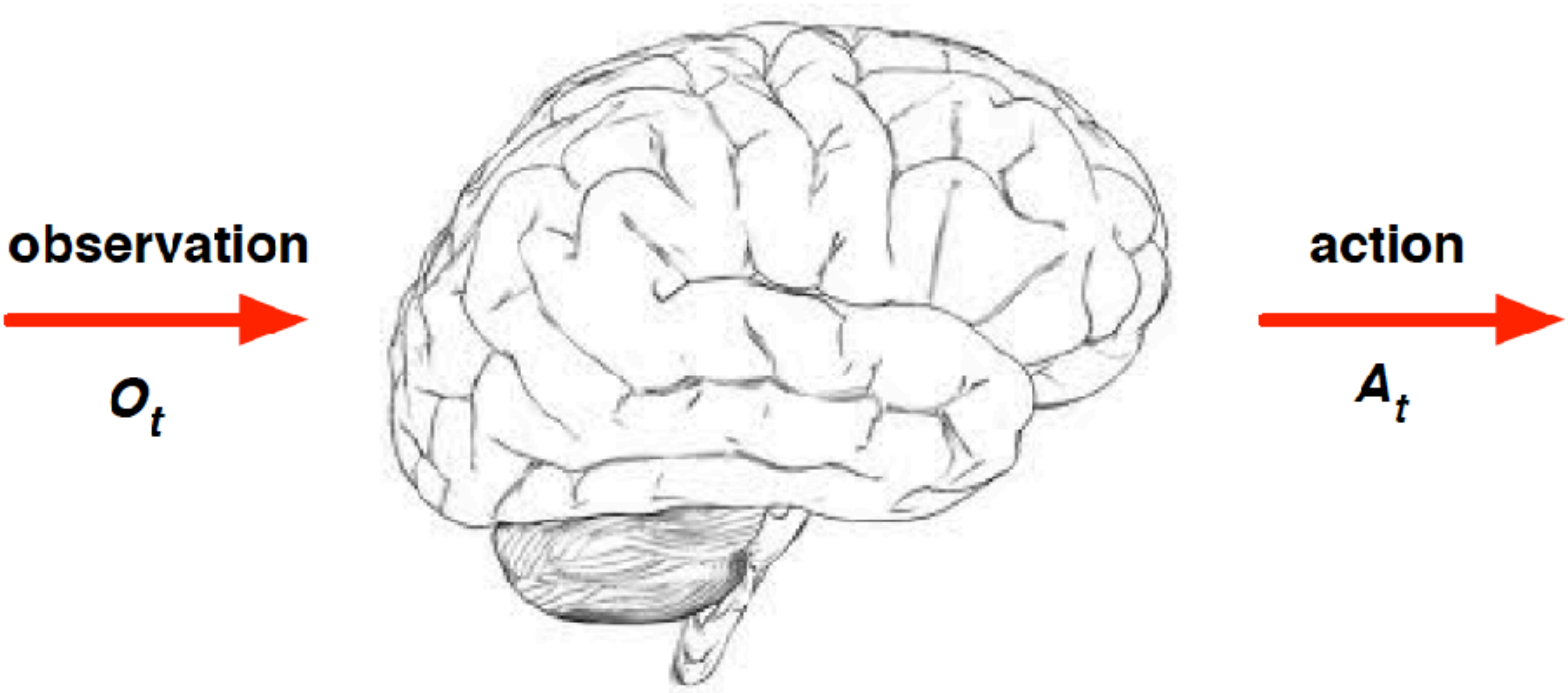


…in contrast to, for example, pixel labelling



Learning to Act

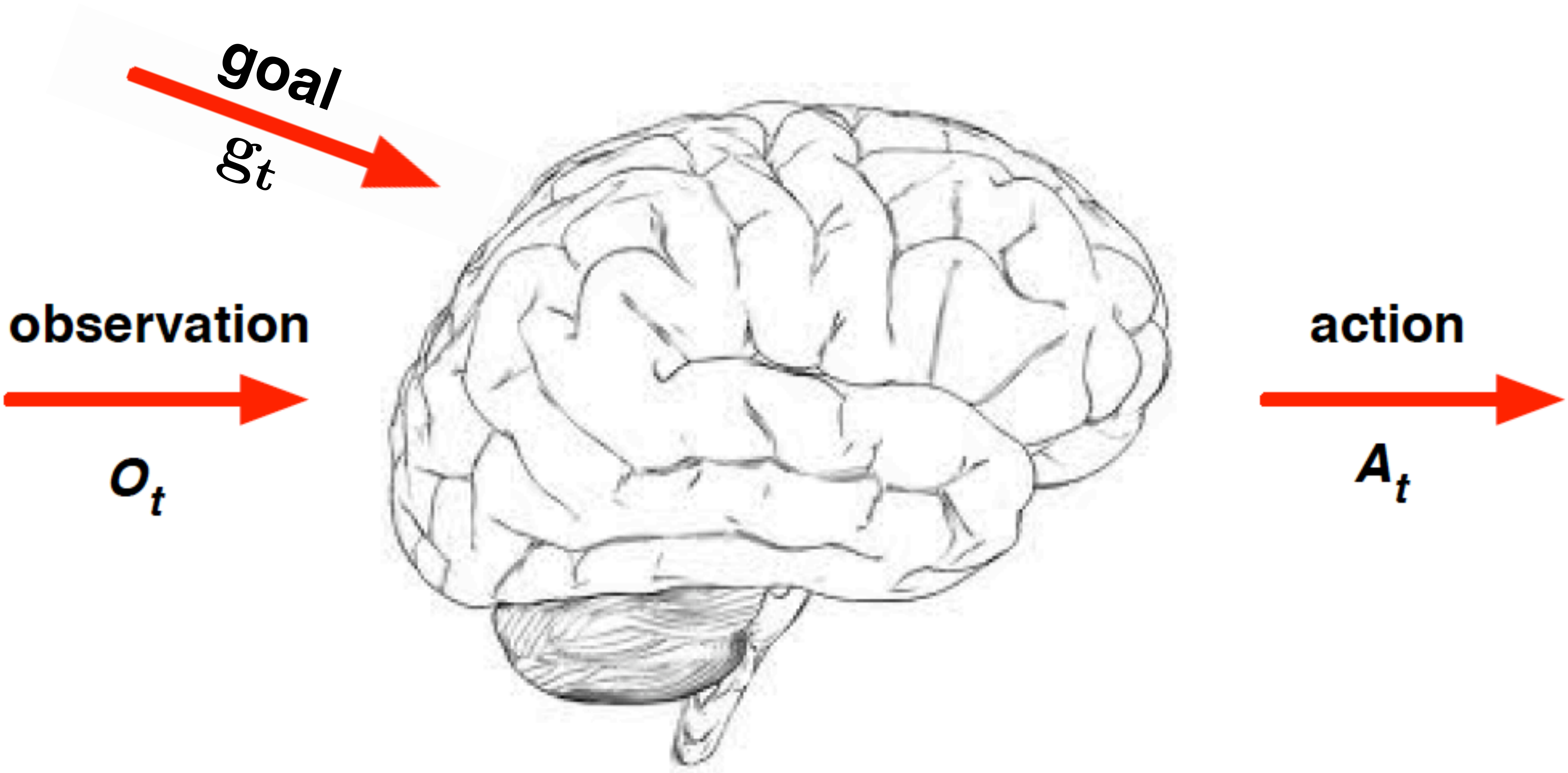
Learning to map sequences of observations to actions



observations: inputs from our sensor

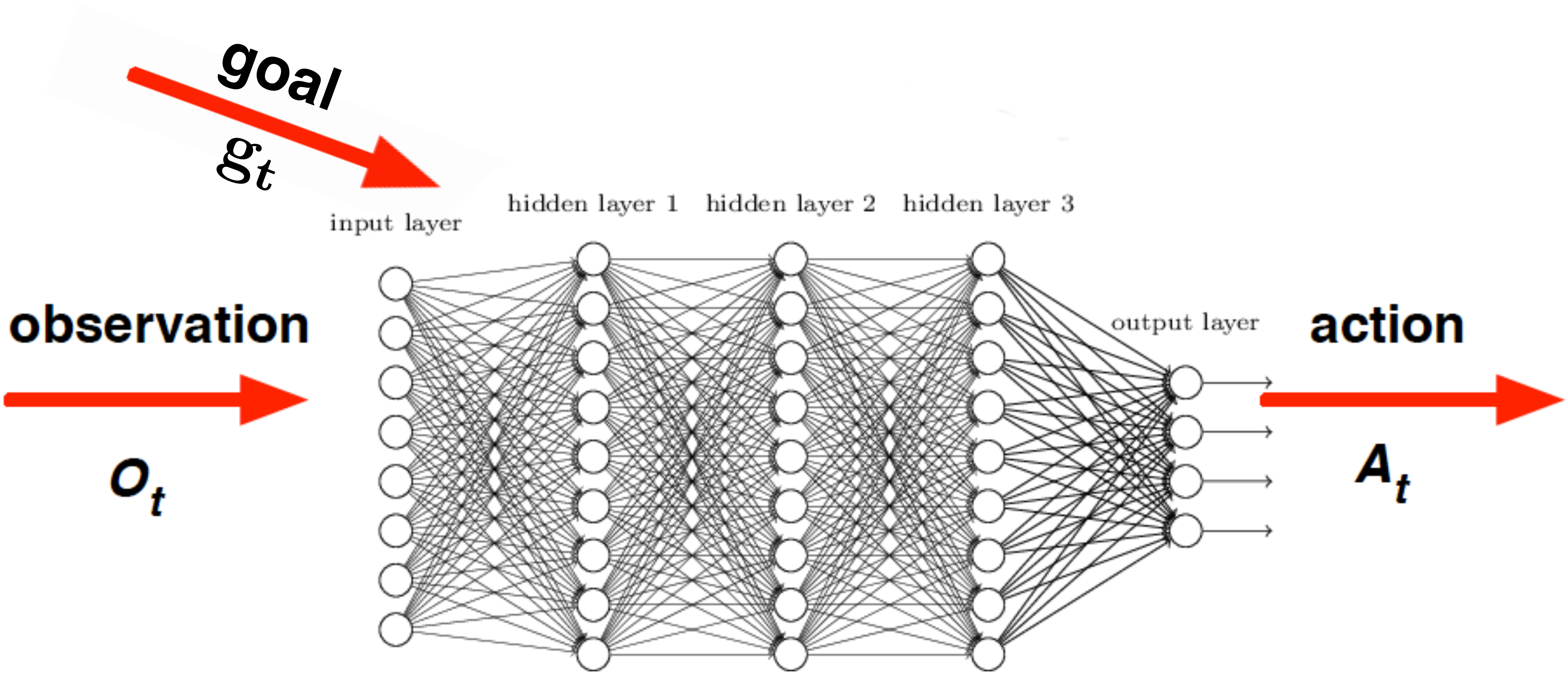
Learning to Act

Learning to map sequences of observations to actions, for a particular goal



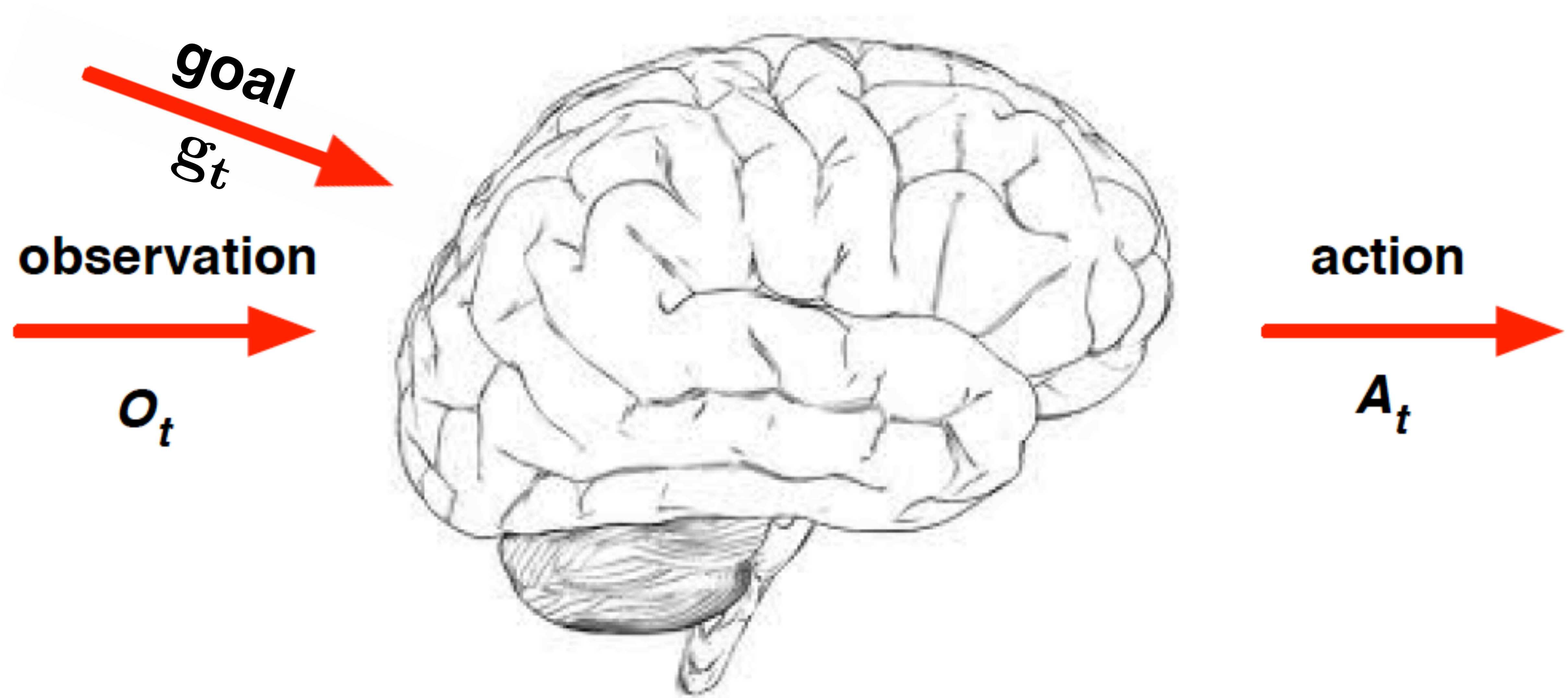
Learning to Act

Learning to map sequences of observations to actions, for a particular goal



Learning to Act

Learning to map sequences of observations to actions, for a particular goal



The mapping from sensory input to actions can be quite complex, much beyond a feedforward mapping of ~30 layers! It may involve mental evaluation of alternatives, unrolling of a model, model updates, closed loop feedback, retrieval of relevant memories, hypothesis generation, etc. .

Limitations of Learning by Interaction

- Can we think of goal directed behavior learning problems that cannot be modeled or are not meaningful using the MDP framework and a trial-and-error Reinforcement learning framework?
- The agent should have the chance to try (and fail) enough times
- This is impossible if episode takes too long, e.g., reward="obtain a great Ph.D."
- This is impossible when safety is a concern: we can't learn to drive via reinforcement learning in the real world, failure cannot be tolerated

Q: what other ways humans use to learn to act in the world?

Value Functions reflect our knowledge about the world

We are social animals and learn from one another: We imitate and we communicate our value functions to one another through natural language

“don’t play video games else your social skills will be impacted”

Value functions capture the knowledge of the agent regarding how good is each state for the goal he is trying to achieve.

Other forms of supervision for learning behaviours?

In this course, we will also visit the first two forms of supervision.

1. Learning from rewards
2. Learning from demonstrations
3. Learning from specifications of optimal behavior

Behavior: High Jump

scissors

Fosbury flop

1. Learning from **rewards**

   Reward: jump as high as possible: It took years for athletes to find the right behavior to achieve this

2. Learning from **demonstrations**

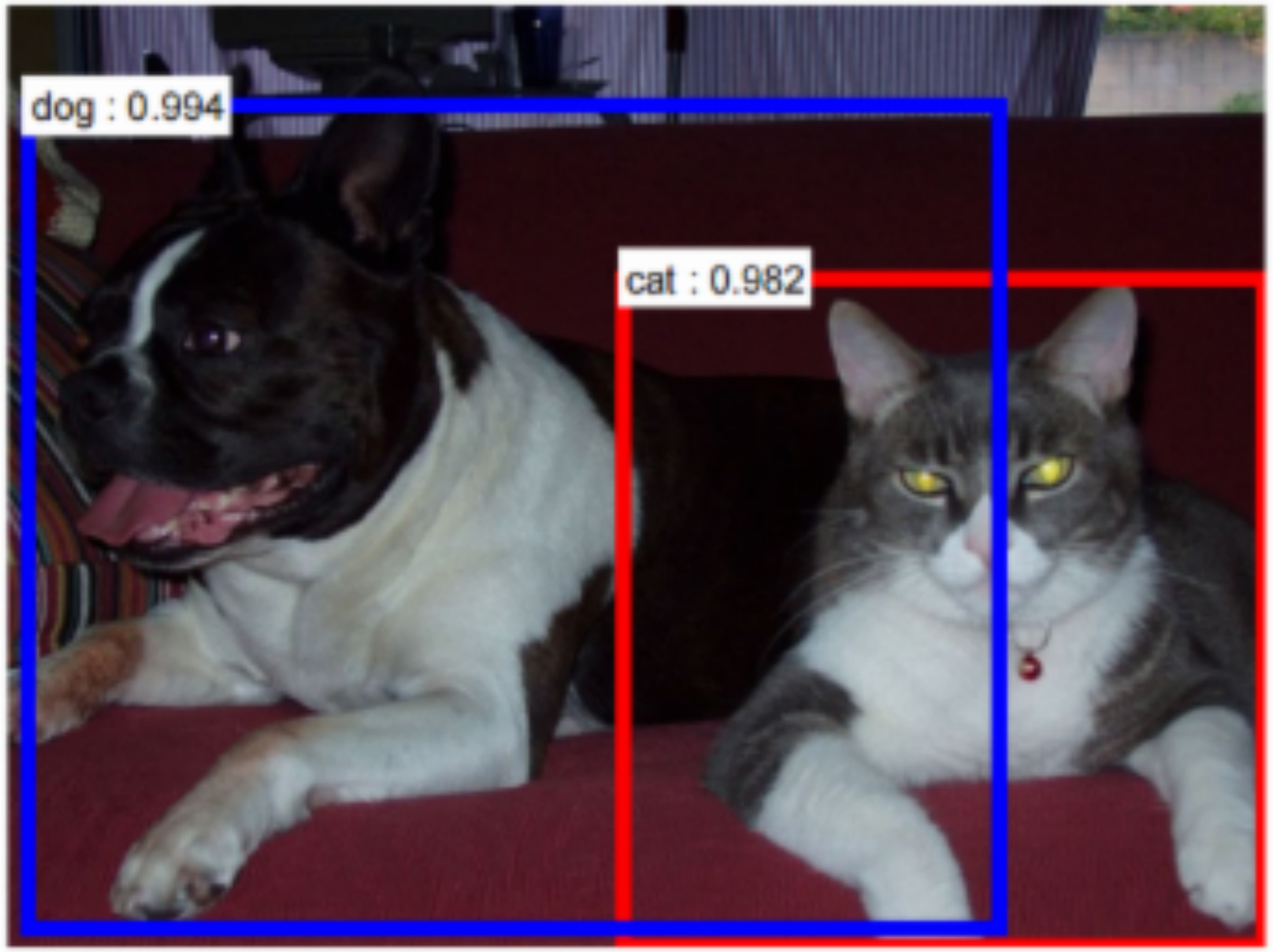
   It was way easier for athletes to perfection the jump, once someone showed the right general trajectory

3. Learning from **specifications of optimal behavior**

   For novices, it is much easier to replicate this behavior if additional guidance is provided based on specifications: where to place the foot, how to time yourself etc.

RL Versus ML

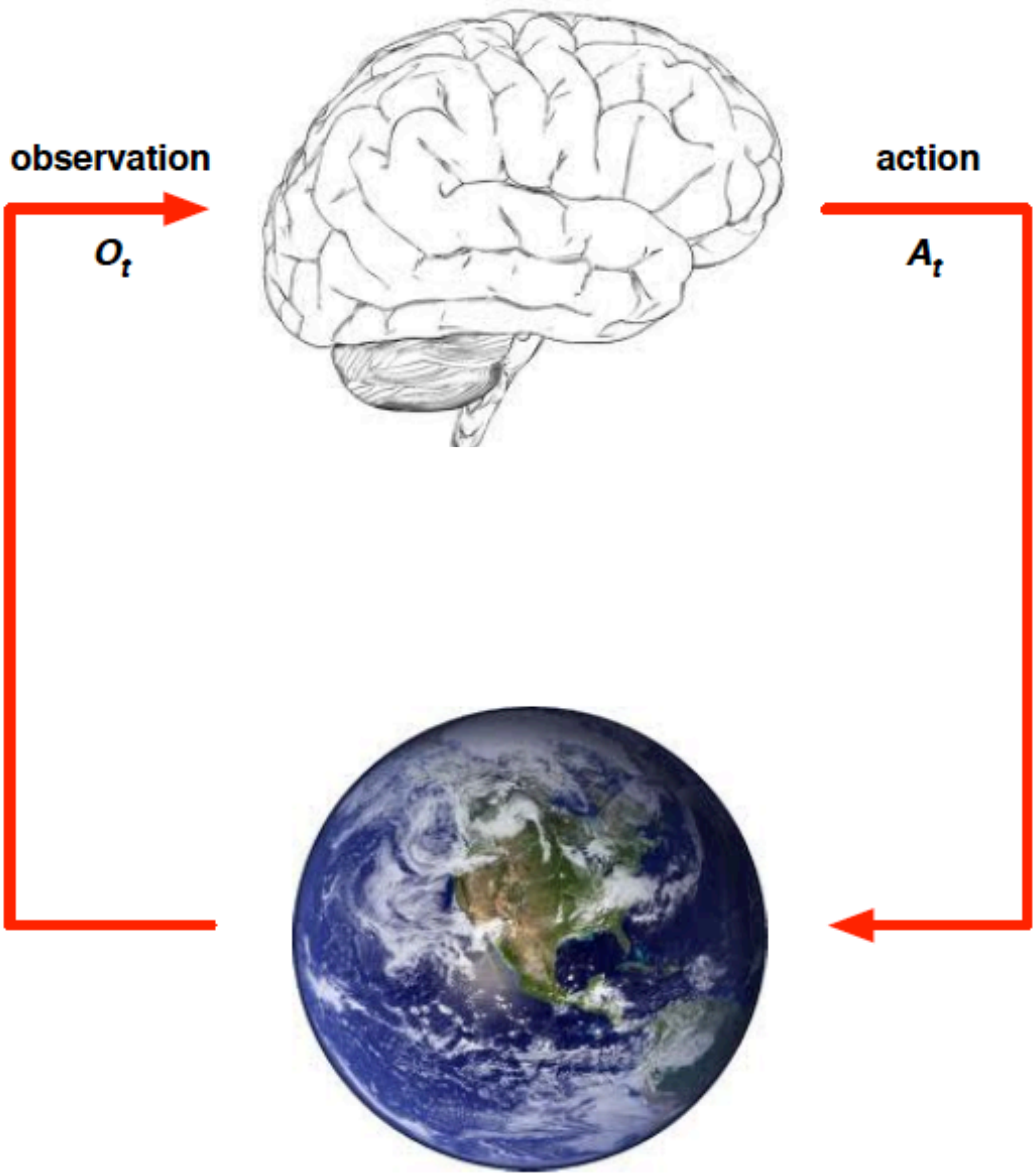
How learning to act is different than other machine learning paradigms, e.g., object detection?



RL Versus ML

How learning to act is different than other machine learning paradigms?

- The agent’s actions affect the data she will receive in the future


observation
O_t
action
A_t

Learning to Act

How learning behaviors is different than other machine learning paradigms?

- The agent’s actions affect the data she will receive in the future:
  - The data the agent receives are sequential in nature, not i.i.d. (independent and identically distributed)
  - Bad policies will never lead you to collect better data.

Learning to Act

How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
   - Temporal credit assignment: which actions were important and which were not, is hard to know

Learning to Act

How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
3) Actions take time to carry out in the real world, we want to minimize the amount of interaction

Learning to Act

How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
3) Actions take time to carry out in the real world, we want to minimize the amount of interaction

Reminds of active learning! we want to ask humans for labels and we want to choose the queries carefully to minimize human involvement
A lecture by Marc Toussaint that shows how those problems are interrelated

Learning to Act

How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
3) Actions take time to carry out in the real world, we want to minimize the amount of interaction
    1) We can use simulated experience and tackle the sim2real transfer

Learning to Act

How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
3) Actions take time to carry out in the real world, and thus this may limit the amount of experience
    - We can use simulated experience and tackle the sim2real transfer
    - We can have robots working 24/7

Supersizing Self-Supervision

Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours, Pinto and Gupta

Learning to Act

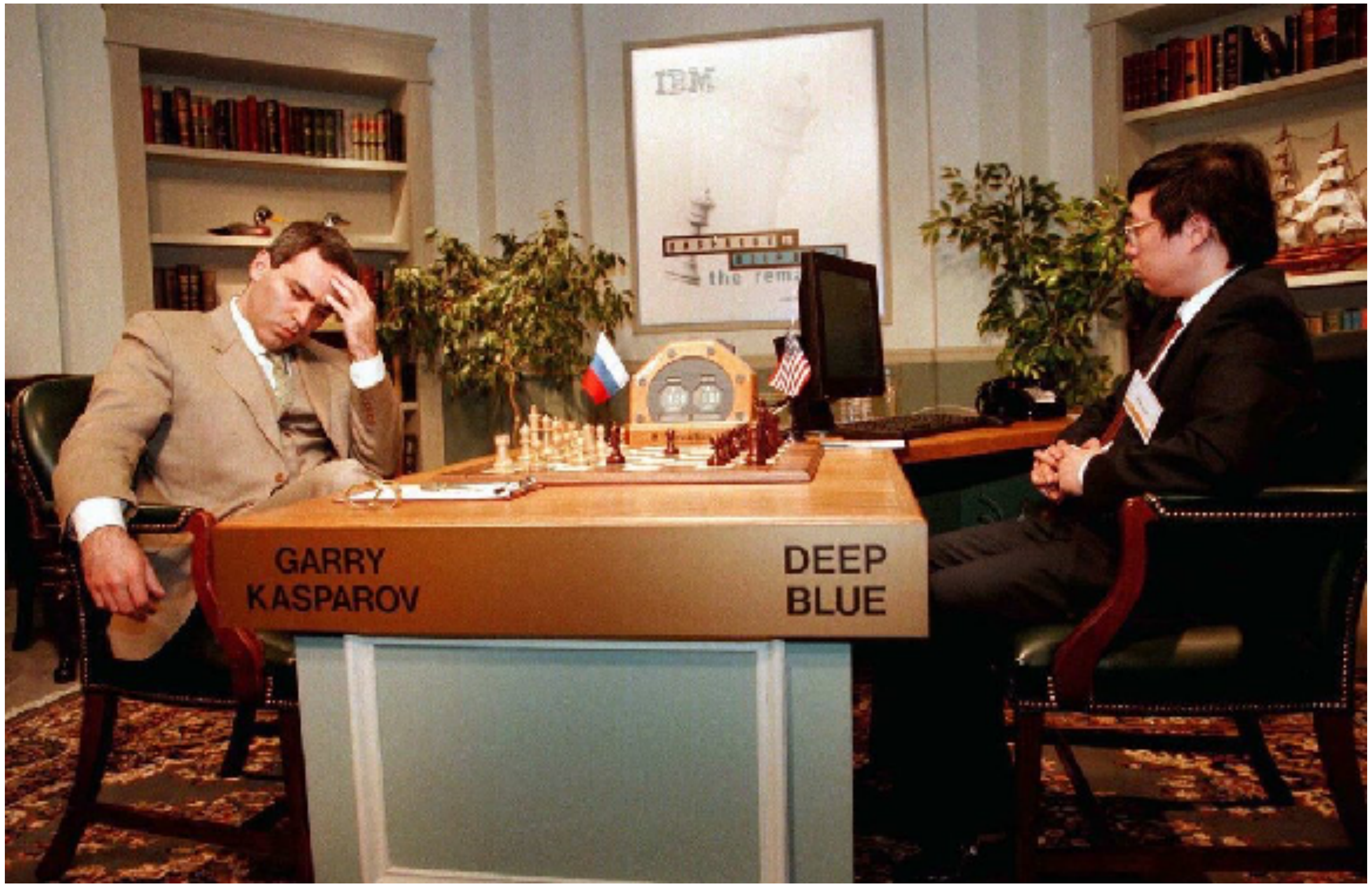
How learning behaviors is different than other machine learning paradigms?

1) The agent's actions affect the data she will receive in the future
2) The reward (whether the goal of the behavior is achieved) is far in the future:
3) Actions take time to carry out in the real world, and thus this may limit the amount of experience
   - We can use simulated experience and tackle the sim2real transfer
   - We can have robots working 24/7
   - We can buy many robots

Google's Robot Farm

Successes so far

Deep Blue



- Q1: Is this a machine learning achievement?
- Q2: What is machine learning / artificial intelligence?
- A2: The discipline that develops agents that learn and improve with experience (Tom Mitchell)
- A1: No, it is not. Brute-force manual development of a board evaluation function

Backgammon

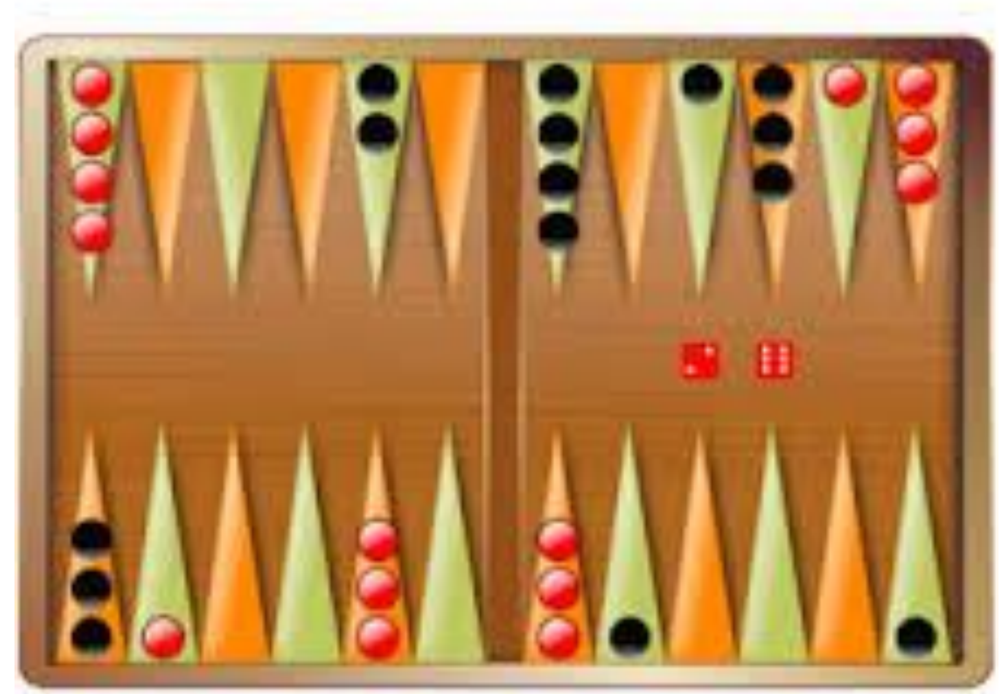
Backgammon

How is it different than chess?

Backgammon

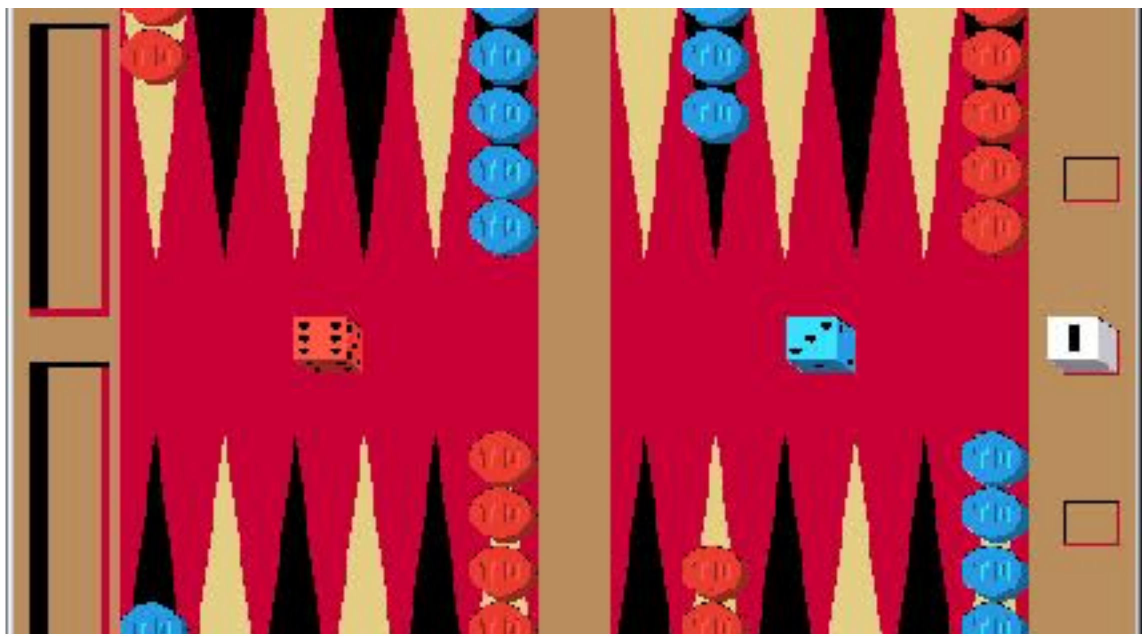
High branching factor due to dice roll prohibits brute force deep searches such as in chess

Neuro-Gammon



- Developed by Gerald Tesauro in 1989 in IBM's research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

TD-Gammon

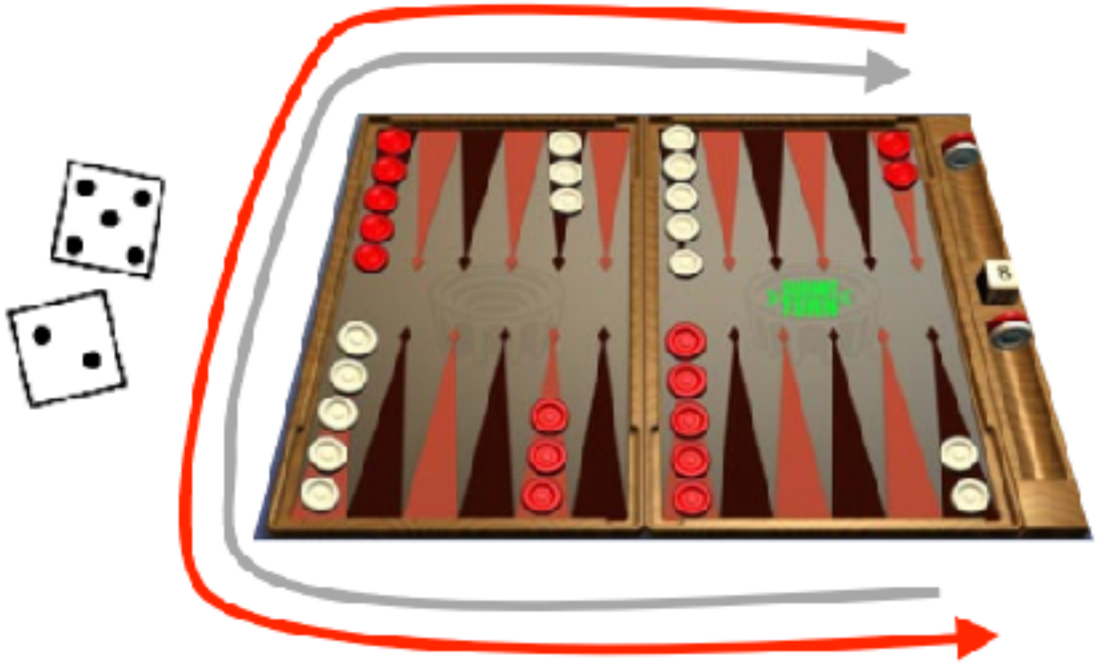


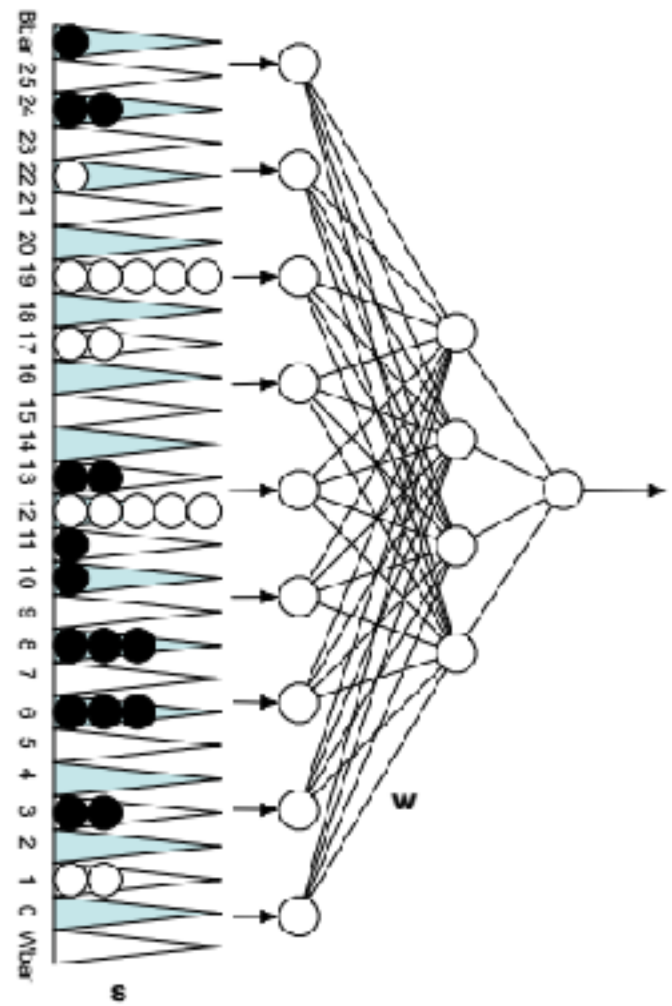
- Developed by Gerald Tesauro in 1992 in IBM’s research center
- A neural network that trains itself to be an **evaluation function** by playing against itself starting from random weights
- Achieved performance close to top human players of its time

Neuro-Gammon

- Developed by Gerald Tesauro in 1989 in IBM’s research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

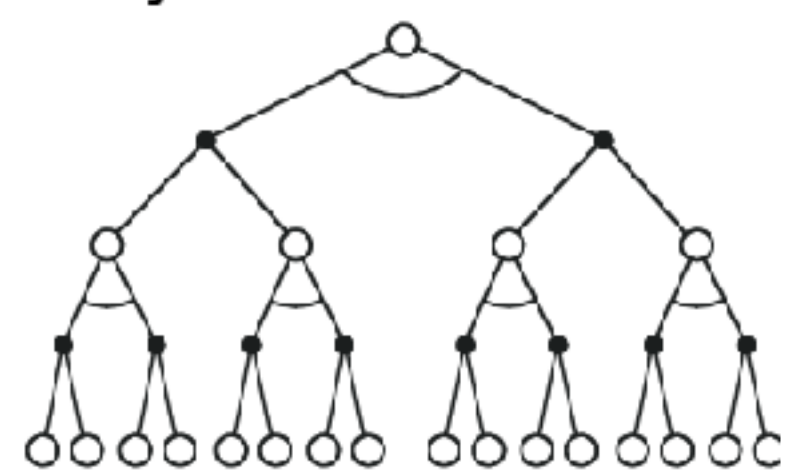
Evaluation function





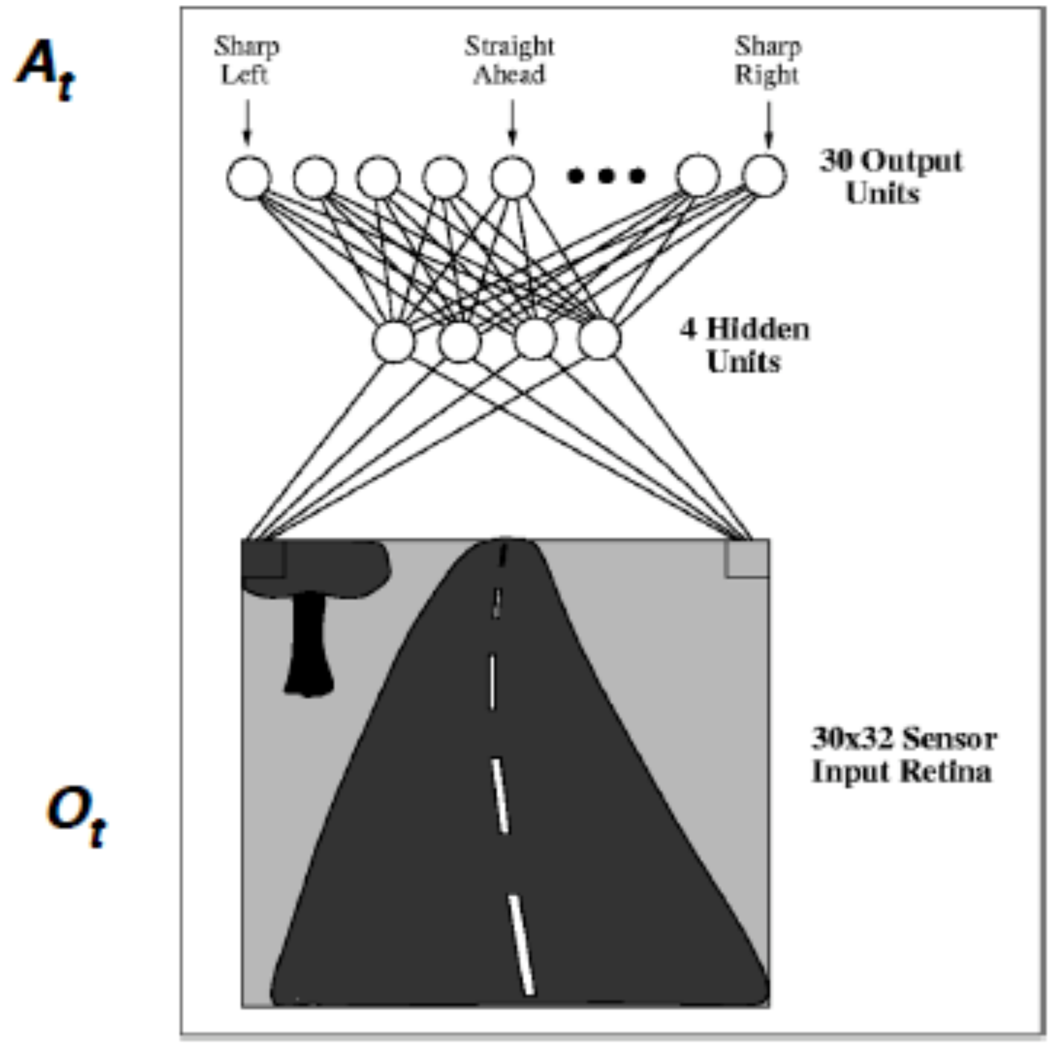
estimated state value
(≈ prob of winning)

Action selection
by a shallow search



Self-Driving Cars

Self-Driving Cars



Policy network π: mapping of observations to actions

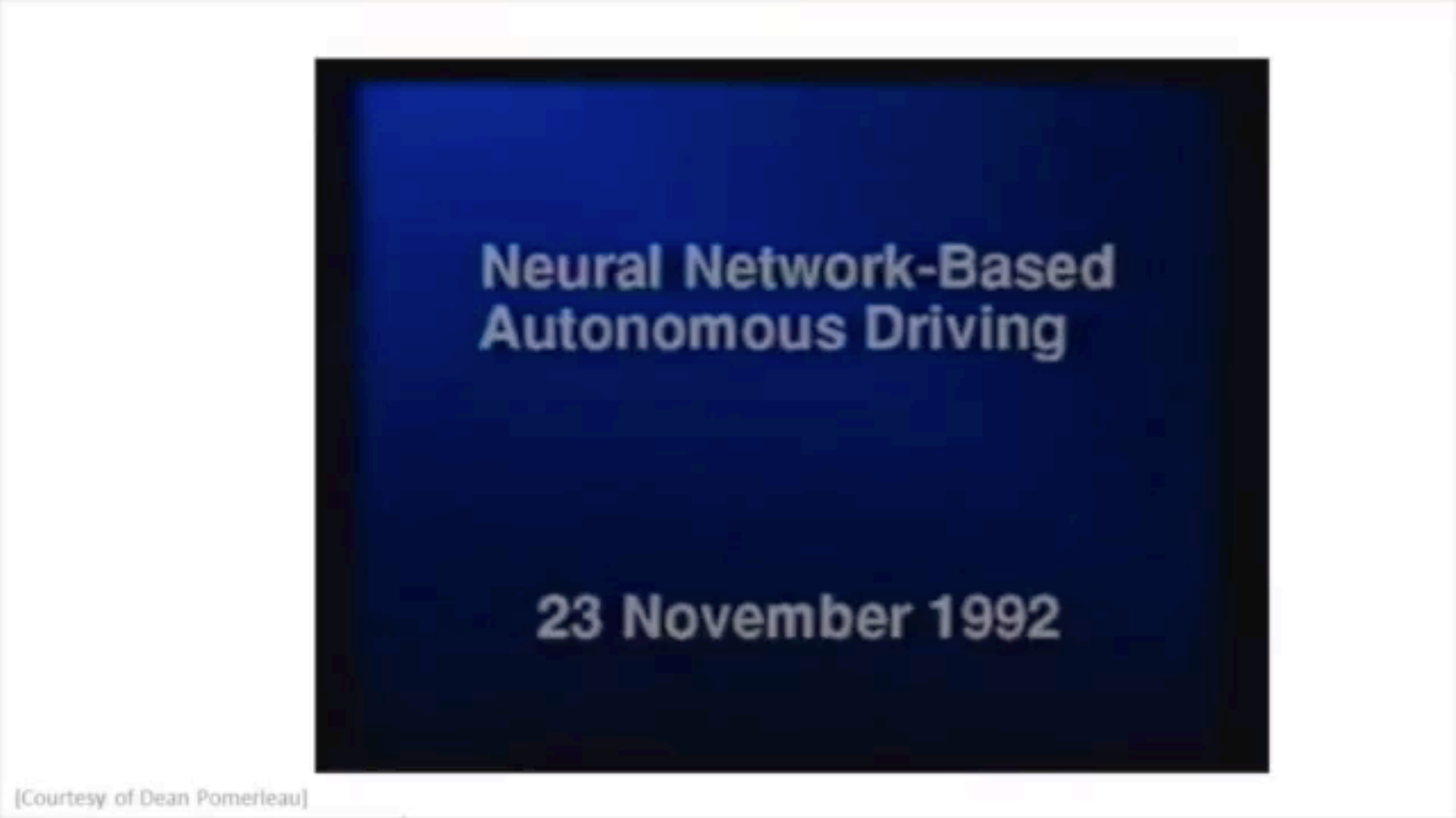
1989

ALVINN, an autonomous land vehicle in a neural network

Dean A. Pomerleau
Carnegie Mellon University

Self-Driving Cars

- ALVINN video



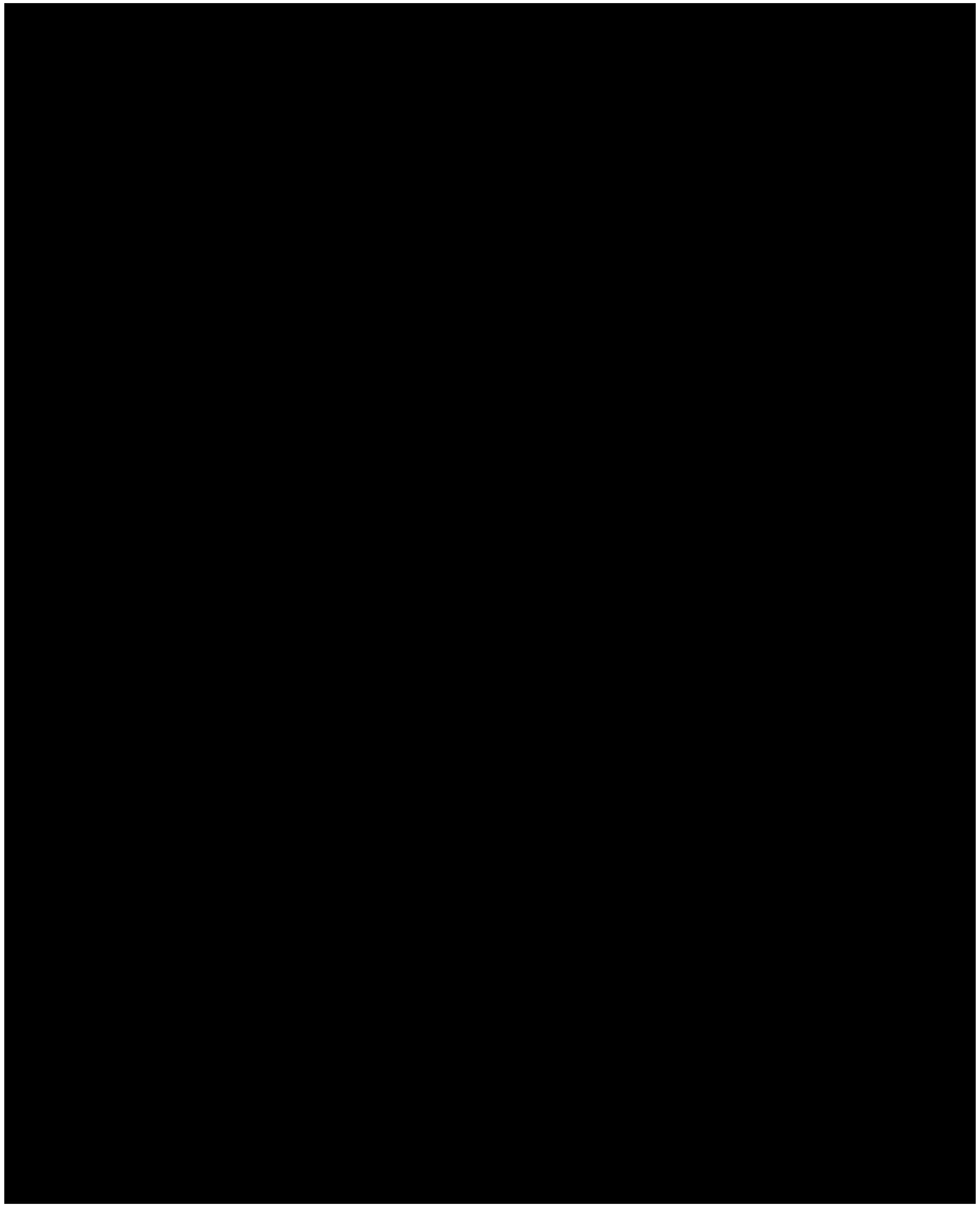
- behavior cloning- learning from the human driver
- data augmentation to deal with compounding errors

ALVINN (Autonomous Land Vehicle In a Neural Network), Efficient Training of Artificial Neural Networks for Autonomous Navigation, Pomerleau 1991

Self-Driving Cars

- Currently: much better computer vision front end: object detection, trajectory forecasting etc.
- Open problem: learning reward functions from humans on how to behave on intersections, crowds, traffic jams, etc. .

Atari



Deep Mind 2014+

Deep Q learning

Atari

[Montezuma's Revenge with Go-Explore]

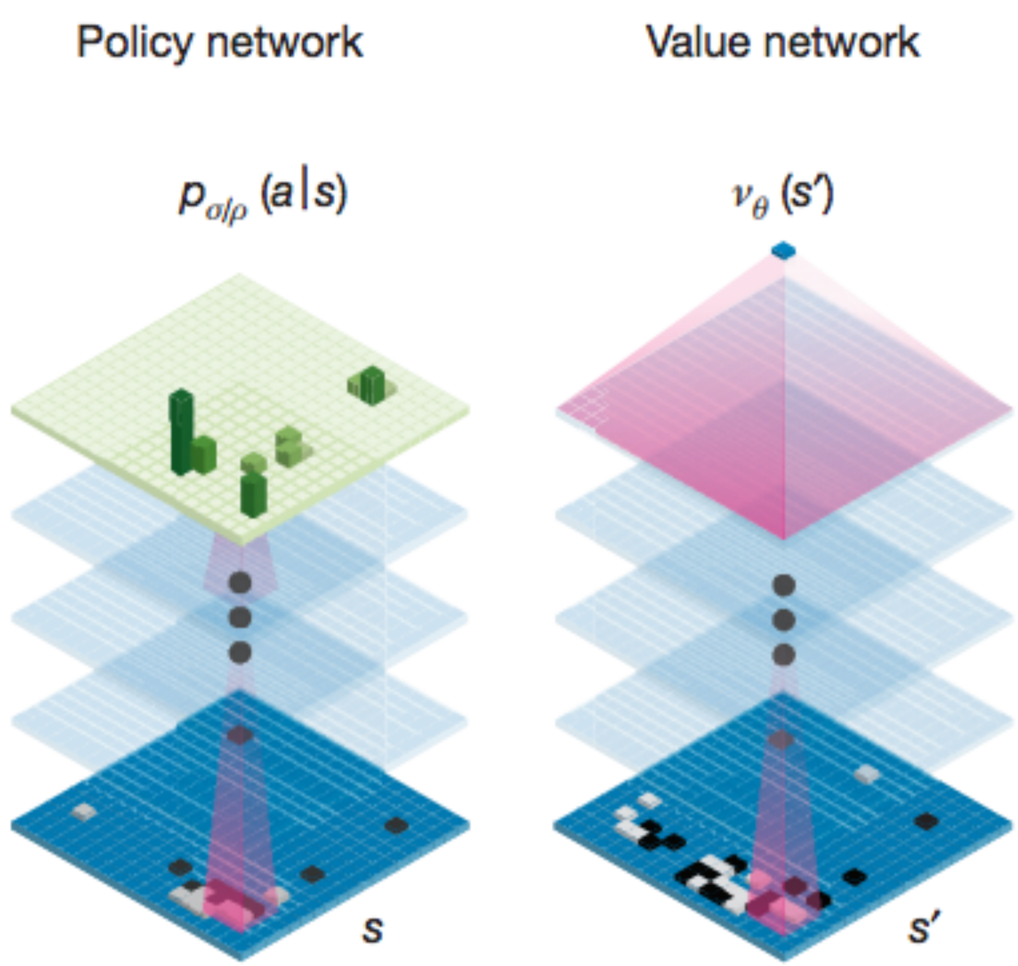
Idea: arXiv your successes

GO

AlphaGo

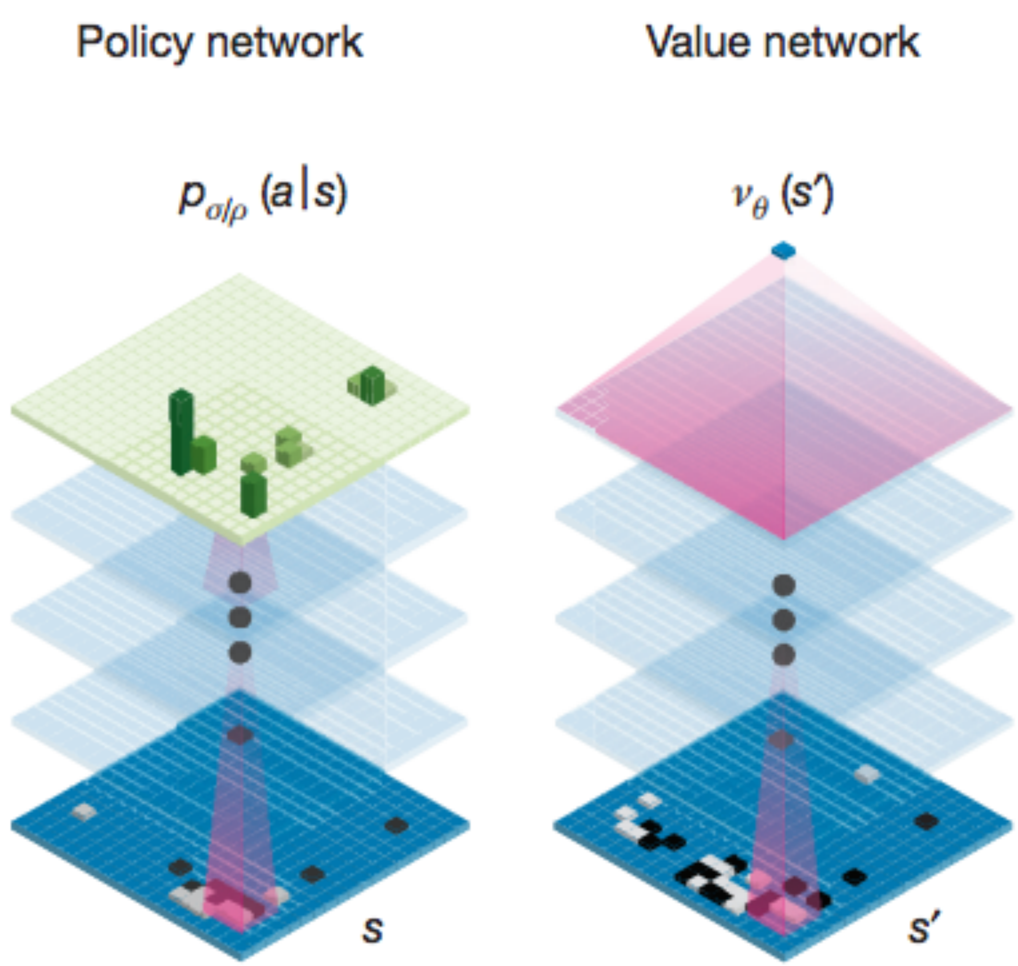
- Monte Carlo Tree Search with neural nets
- expert demonstrations
- self play

AlphaGo



Policy net trained to mimic expert moves, and then fine-tuned using self-play

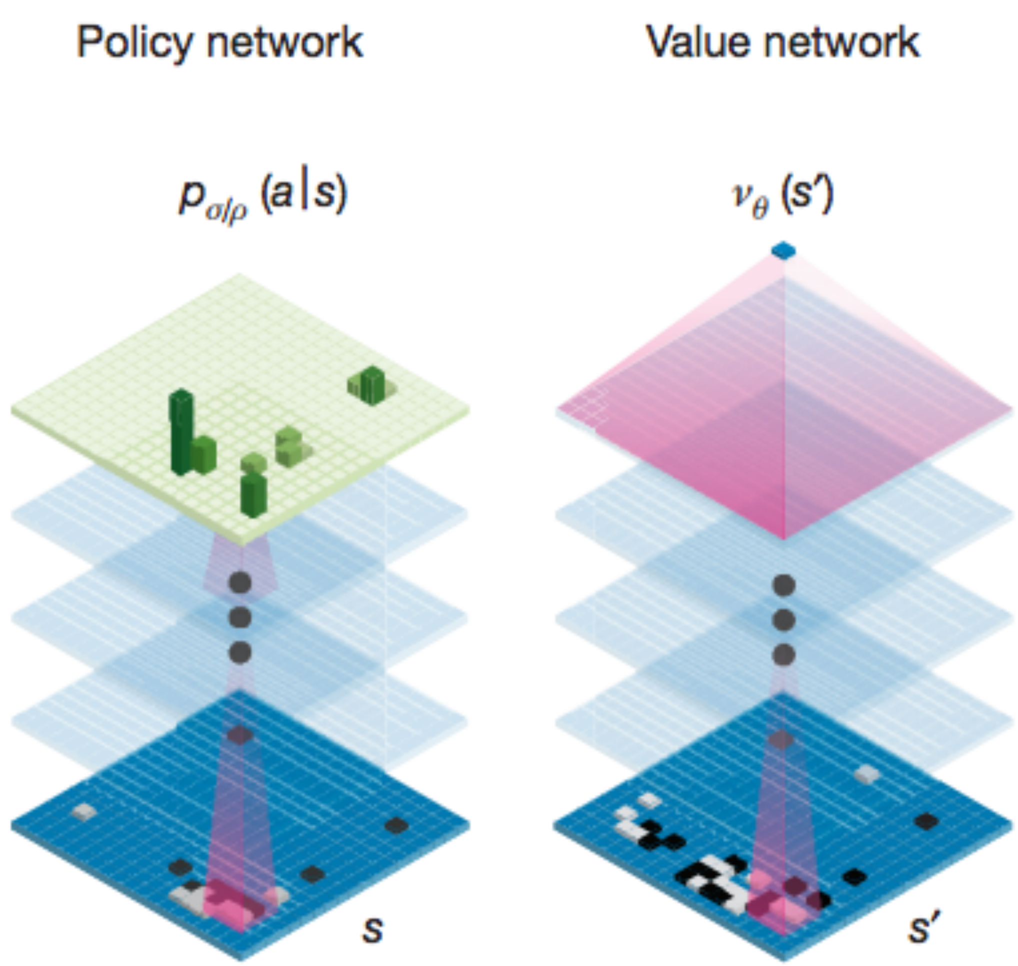
AlphaGo



Policy net trained to mimic expert moves, and then fine-tuned using self-play

Value network trained with regression to predict the outcome, using self play data of the best policy.

AlphaGo

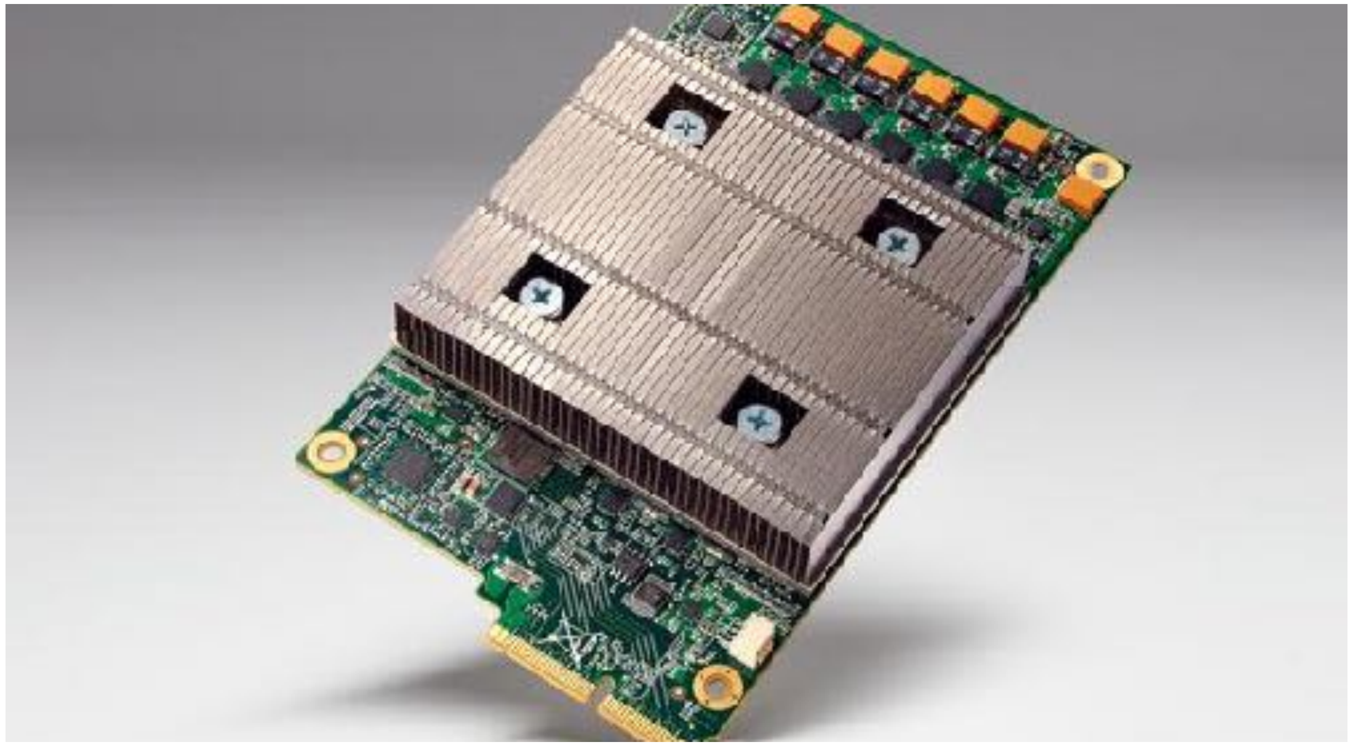


Policy net trained to mimic expert moves, and then fine-tuned using self-play

Value network trained with regression to predict the outcome, using self play data of the best policy.

At test time, policy and value nets guide a MCTS to select stronger moves by deep look ahead.

AlphaGo

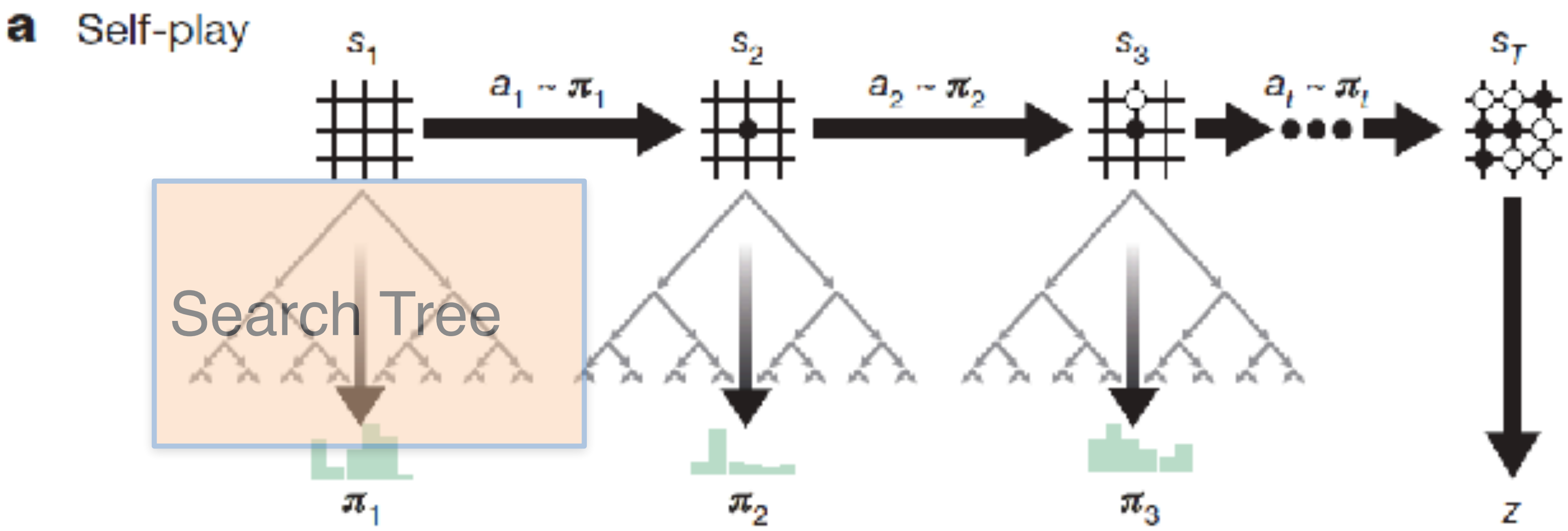


Tensor Processing Unit from Google

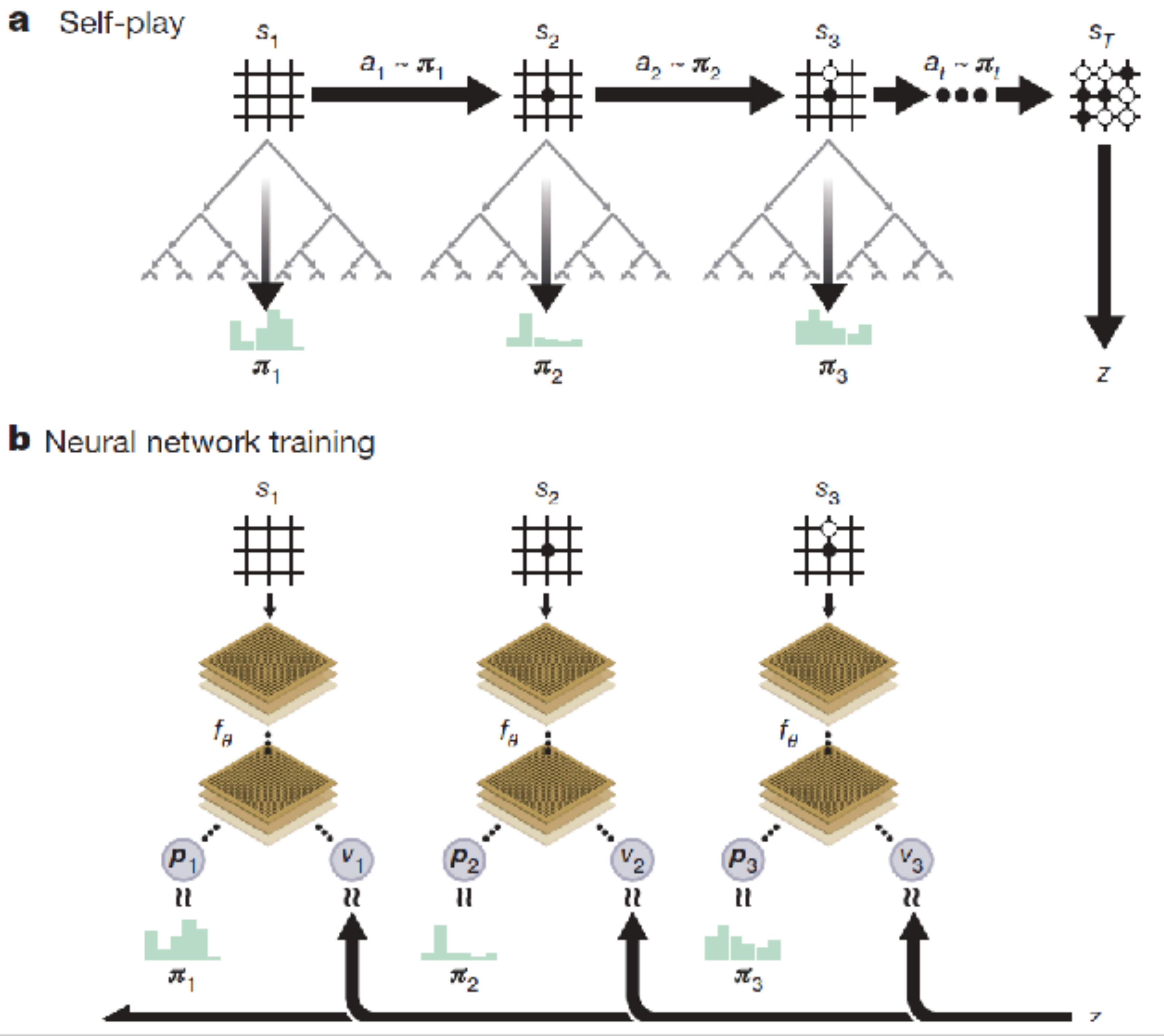
AlphaGoZero

- No human supervision!
- MCTS to select great moves during training and testing!

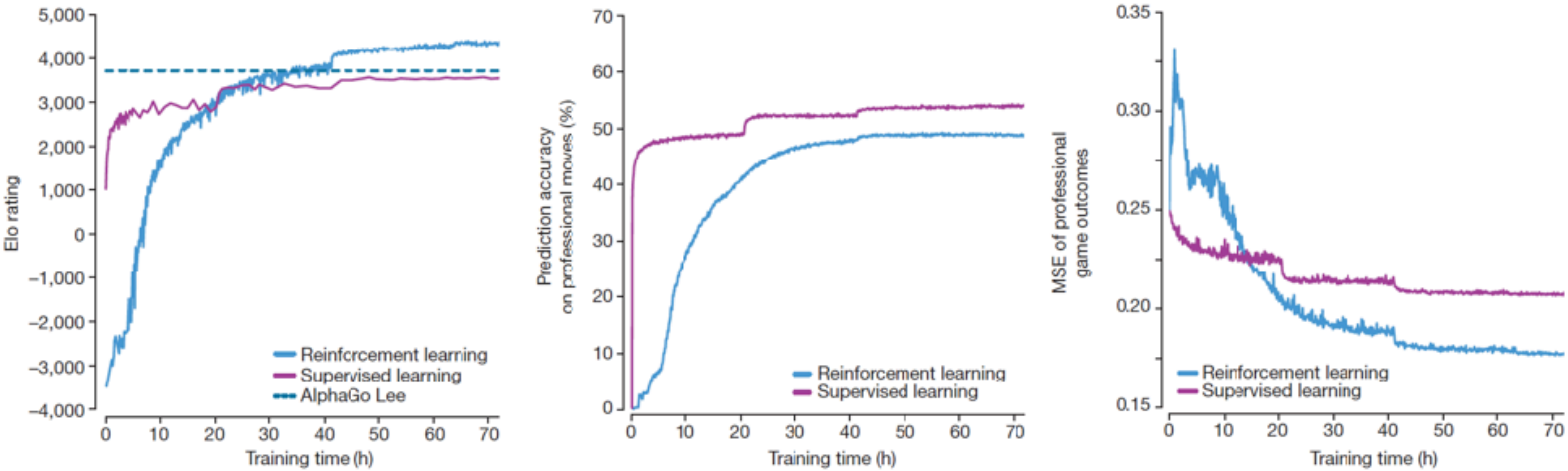
AlphaGoZero



AlphaGoZero



AlphaGoZero



Go Versus the real world

How the world of Alpha Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse
3. **Discrete Vs Continuous actions**
4. **One goal Vs many goals**
5. **Rewards automatic VS rewards need themselves to be detected**

Alpha Go Versus the real world

How the world of Alpha Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse

Go Versus the real world

How the world of Alpha Go is different than the real world?

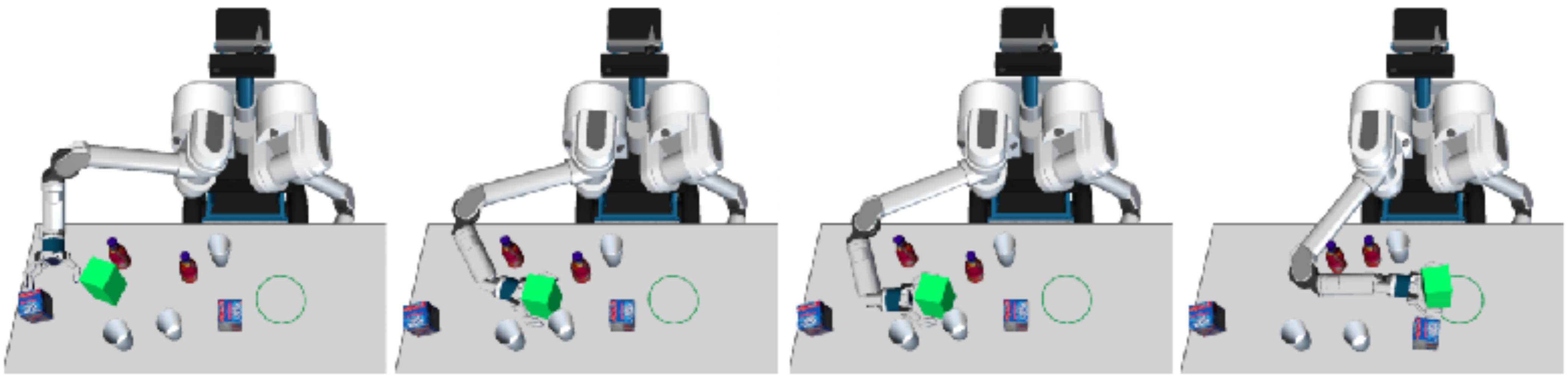
1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse

State estimation: To be able to act you need first to be able to see, detect the objects that you interact with, detect whether you achieved your goal

State estimation

Most works are between two extremes:

- Assuming the world model known (object locations, shapes, physical properties obtain via AR tags or manual tuning), they use planners to search for the action sequence to achieve a desired goal.

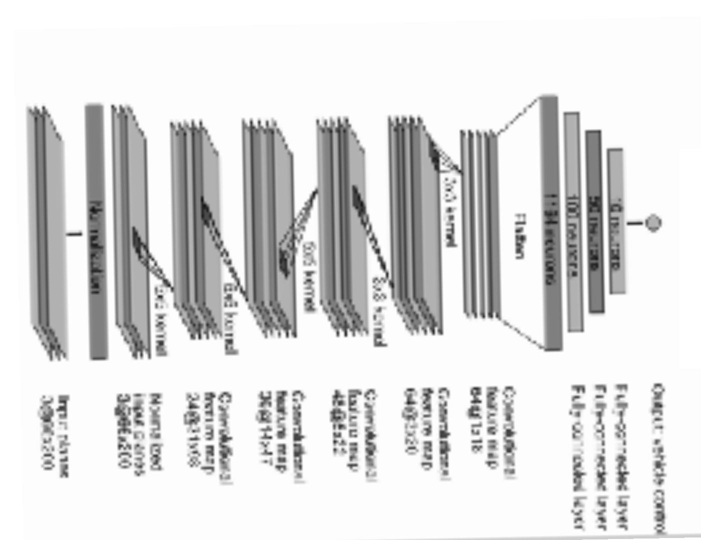


Rearrangement Planning via Heuristic Search, Jennifer E. King, Siddhartha S. Srinivasa

State estimation

Most works are between two extremes:

- Assuming the world model known (object locations, shapes, physical properties obtain via AR tags or manual tuning), they use planners to search for the action sequence to achieve a desired goal.
- Do not attempt to detect any objects and learn to map RGB images directly to actions



End-to-End Learning for Self-Driving Cars, NVIDIA

Go Versus the real world

How the world of Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse
3. **Discrete Vs Continuous actions**
4. **One goal Vs many goals**
5. **Rewards automatic VS rewards need themselves to be detected**

Go Versus the real world

How the world of Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse
3. **Discrete Vs Continuous actions** (curriculum learning, progressively add degrees of freedom)
4. **One goal Vs many goals**
5. **Rewards automatic VS rewards need themselves to be detected**

Go Versus the real world

How the world of Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse
3. **Discrete Vs Continuous actions** (curriculum learning, progressively add degrees of freedom)
4. **One goal Vs many goals** (generalized policies parametrized by the goal, Hindsight Experience Replay)
5. **Rewards automatic VS rewards need themselves to be detected**

Alpha Go Versus the real world

How the world of Go is different than the real world?

1. **Known environment** (known entities and dynamics) **Vs Unknown environment** (unknown entities and dynamics).
2. Need for behaviors to **transfer** across environmental variations since the real world is very diverse
3. **Discrete Vs Continuous actions** (curriculum learning, progressively add degrees of freedom)
4. **One goal Vs many goals** (generalized policies parametrized by the goal, Hindsight Experience Replay)
5. **Rewards automatic VS rewards need themselves to be detected** (learning perceptual rewards, use Computer Vision to detect success)

What we will cover in this course

Date	*Topic (slides)*	*Assignments*	*Readings*
01/15	Course Introduction		[1, SB Ch1, Ch16]
01/17	Imitation via Behavior Cloning	HW1 is out	[20,22,23,36]
01/18	RECITATION: tensor flow, keras		
01/22	Introduction to policy search		[SB, Ch 3, 4,]
01/24	Monte Carlo learning and temporal difference learning in tabular MDPs		[SB, Ch 5, 6,7]
01/25	RECITATION: OPENAI gym, AWS		
01/29	Bayesian bandits, Monte Carlo tree search	HW2 is out, HW1 is due	[SB Ch 2,Ch 8]
01/31	Monte Carlo learning and temporal difference learning with value function approximation		[SB Ch 9]
02/05	Deep Q learning		[4,5,6]
02/07	Policy gradients, actor-critic methods		[SB Ch 13,7]
02/12	Policy gradients, actor-critic methods, DDPG	HW3 is out, HW2 is due	[SB Ch 13,7,14]
02/14	Monte Carlo tree search with neural networks, AlphaGo, AlphaGoZero		[2,3]
02/19	Model learning and model-based RL		[]
02/21	Model learning and model-based RL	HW3 is due	[]
02/26	Natural policy gradients		[12,13]
02/28	Exploration, intrinsic motivation		[9,10,11]
03/05	Exploration, intrinsic motivation	HW4 is out	
03/07	BUFFER		
03/12	SPRING BREAK		
03/14	SPRING BREAK		
03/19	Maximum-entropy RL		[19]
03/21	Multigoal RL	HW4 is due, HW5 is out	[17,18]
03/26	RL with auxiliary objectives, transfer learning, hierarchical RL		
03/28	Advanced evolutionary methods		[37]
04/02	Learning to imitate by watching videos		
04/04	Generative adversarial imitation learning	HW5 is due	[16]
04/09	BUFFER		
04/11	NO CLASSES		
04/16	Special topic: experimental design		
04/18	Special topic: visual perception in RL		
04/23	Special topic: learning to navigate		
04/25	Special topic: curriculum learning		
04/30	Project presentation and discussion		
05/02	Very recent advances and open problems		

AI’s paradox

Go Versus the real world

Beating the world champion is easier than moving the Go stones.

AI’s paradox

Hans Moravec

"it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"

AI’s paradox

Marvin Minsky

"we're more aware of simple processes that don't work well than of complex ones that work flawlessly"

Evolutionary explanation

Hans Moravec

We should expect the difficulty of reverse-engineering any human skill to be roughly proportional to the amount of time that skill has been evolving in animals.
The oldest human skills are largely unconscious and so appear to us to be effortless.
Therefore, we should expect skills that appear effortless to be difficult to reverse-engineer, but skills that require effort may not necessarily be difficult to engineer at all.

What is AI?

intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration, proving mathematical theorems and solving complicated word algebra problems.

Rodney Brooks

What is AI?

Rodney Brooks

intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration, proving mathematical theorems and solving complicated word algebra problems. "The things that children of four or five years could do effortlessly, such as visually distinguishing between a coffee cup and a chair, or walking around on two legs, or finding their way from their bedroom to the living room were not thought of as activities requiring intelligence.

What is AI?

Rodney Brooks

intelligence was "best characterized as the things that highly educated male scientists found challenging", such as chess, symbolic integration,
proving mathematical theorems and solving complic
"The thi

No cognition. Just sensing and action

years could do effortlessly, such as visually distinguishing between a coffee cup and a chair, or walking around on two legs, or finding their way from their bedroom to the living room were not thought of as activities requiring intelligence.

Learning from Babies

- *Be multi-modal*
- *Be incremental*
- *Be physical*
- *Explore*
- *Be social*
- *Learn a language*

The Development of Embodied Cognition: Six Lessons from Babies
Linda Smith, Michael Gasser