Carnegie Mellon

School of Computer Science

Deep Reinforcement Learning and Control

Introduction to Deep Reinforcement Learning and Control

Fall 2019, CMU 10-703

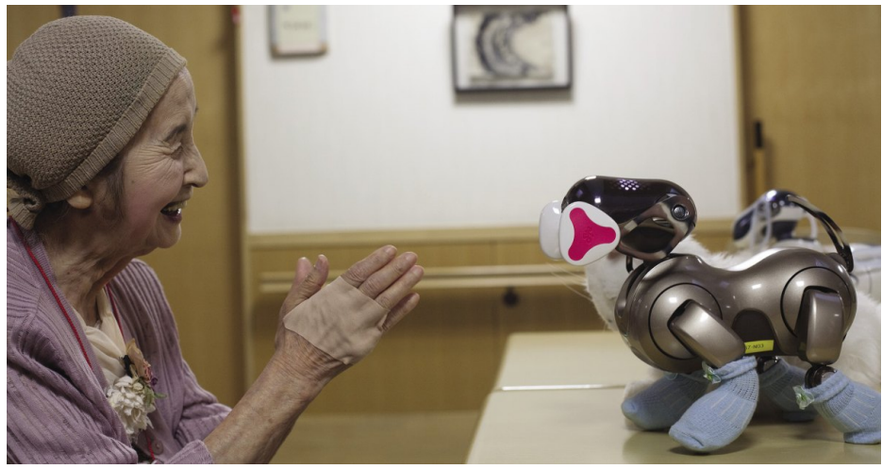
Katerina Fragkiadaki

Course Logistics

- Course website : all you need to know
- Grading: 7 Homework assignments (implementation and question/ answering), many optional and extra grade questions
- Resources: AWS for those that do not have access to GPUs
- HW code for guidance will be in TensorFlolw, you can use your favorite deep learning package
- People can audit the course, unless there are no seats left in class
- The readings on the schedule are **required**

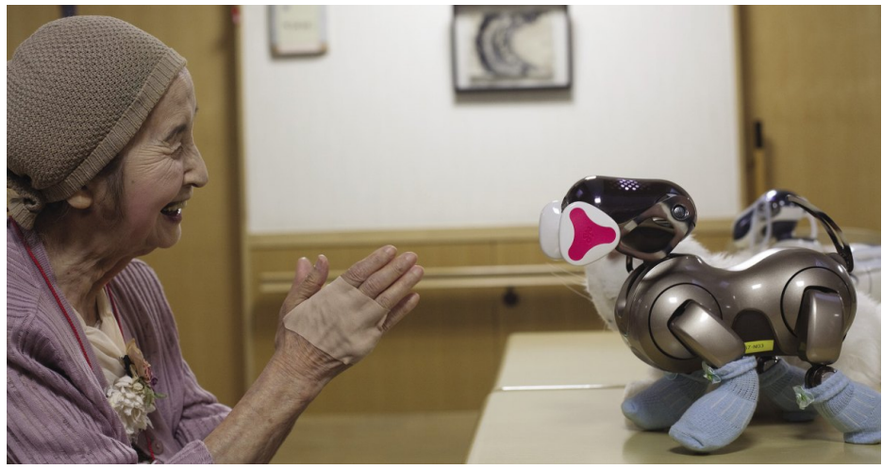
Goal of the Course: Learning behaviors

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

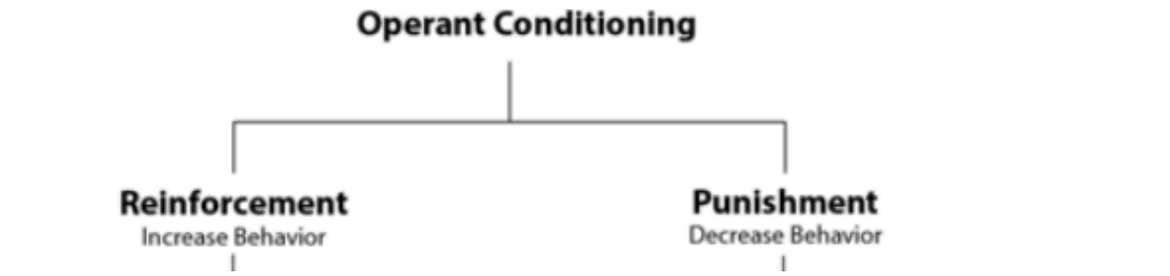
Sea squirts digest their own brain when they decide not to move anymore

Learning behaviours through reinforcement

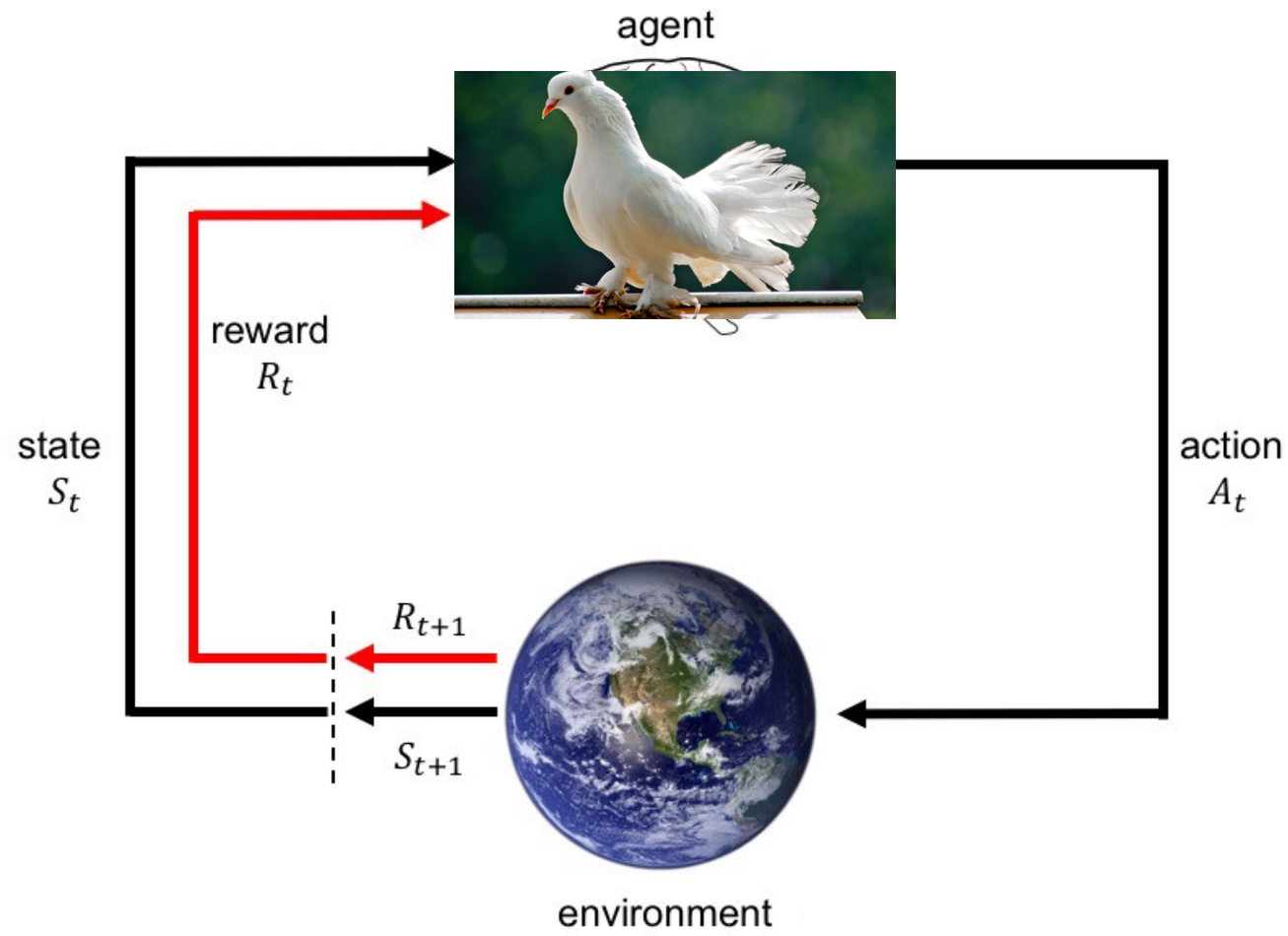
Behavior is primarily shaped by reinforcement rather than free-will.

- 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



Reinforcement learning



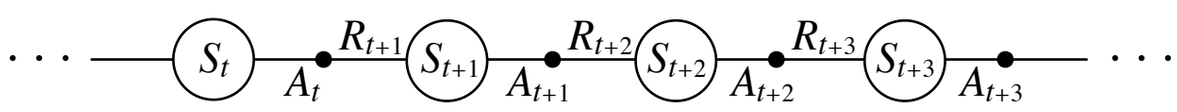
Agent and environment interact at discrete time steps: $t = 0, 1, 2, 3, \ldots$

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

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

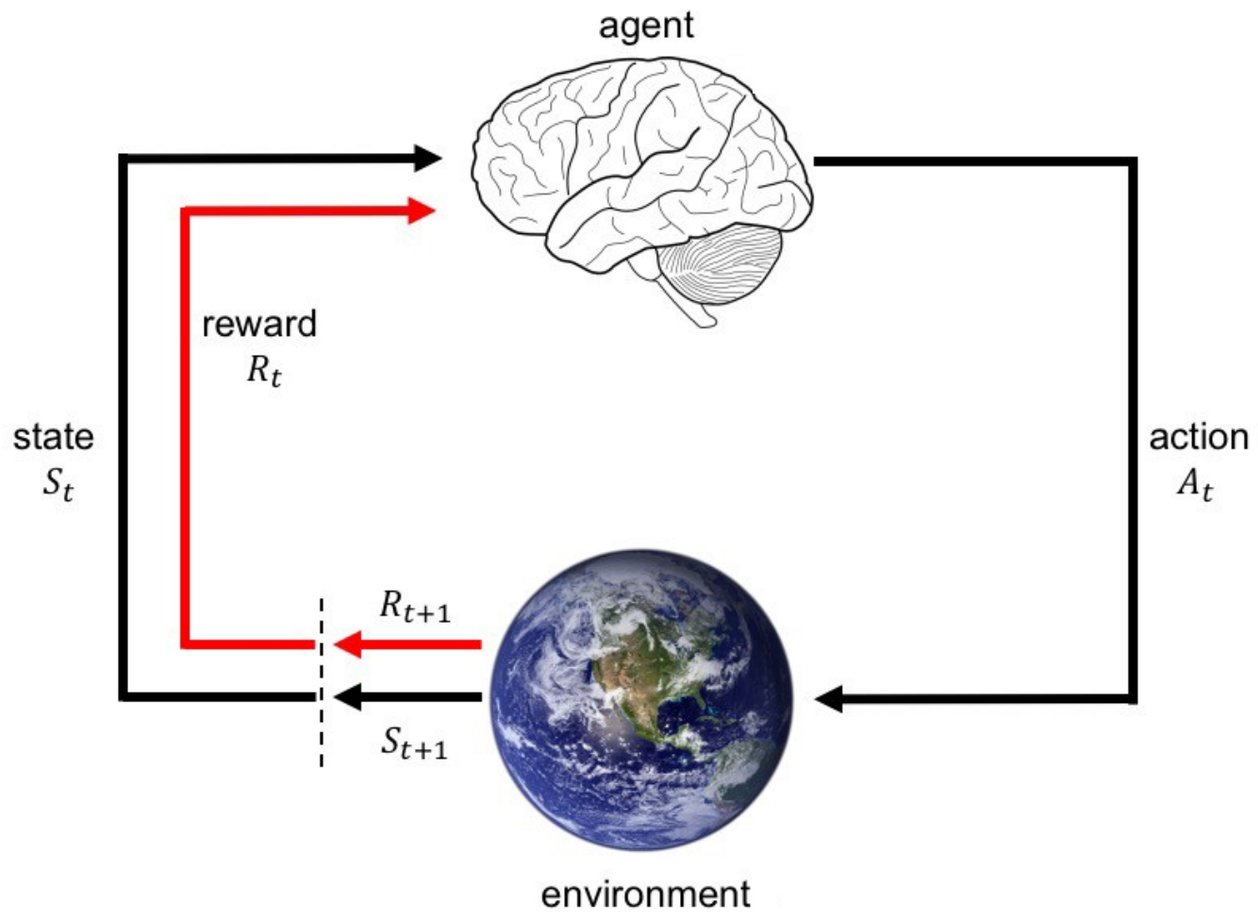
gets resulting reward: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

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



Reinforcement learning

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

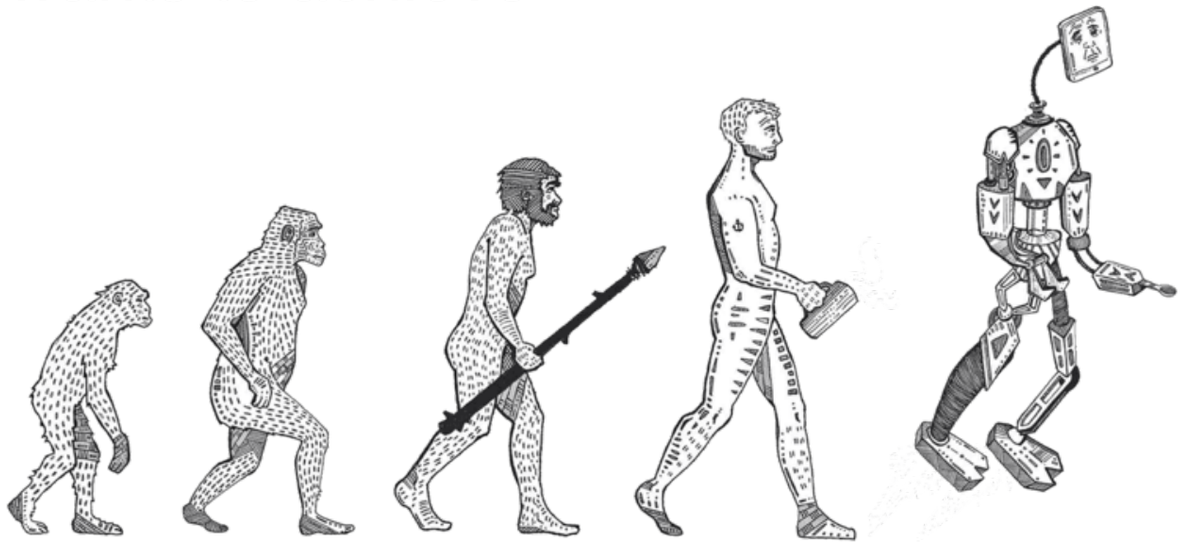


Note: Rewards can be intrinsic, i.e., generated by the agent and guided by its curiosity as opposed to an external task

Agent

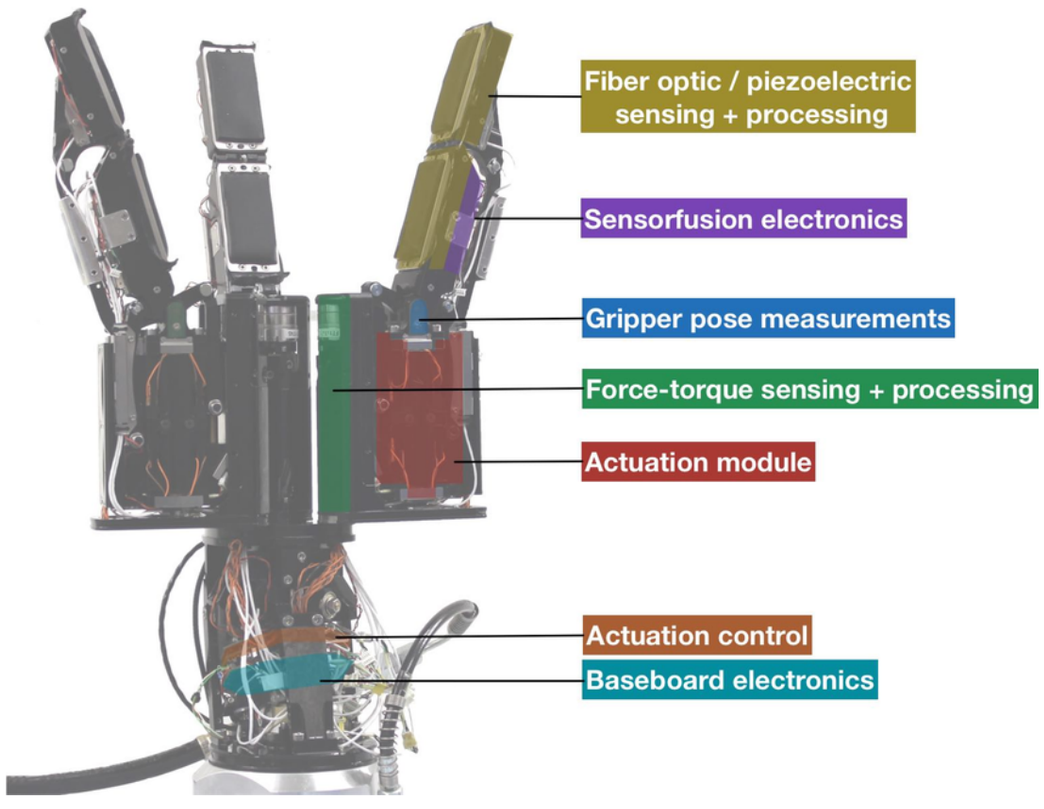
An entity that is equipped with

- 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.

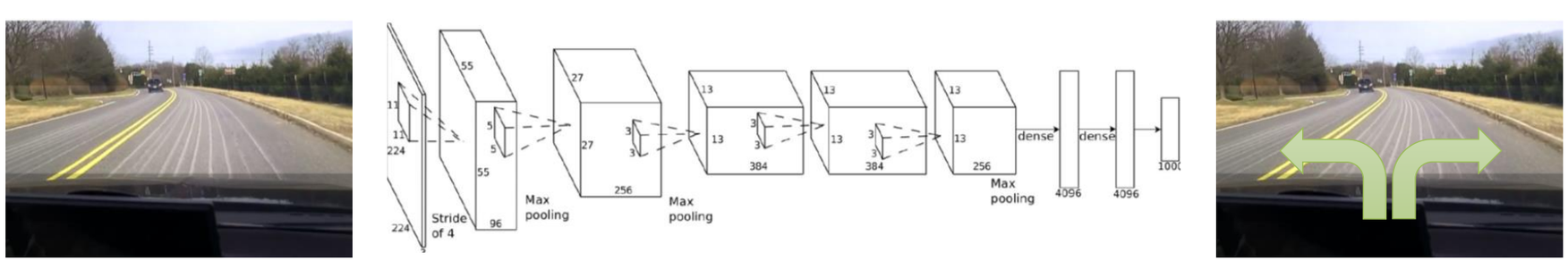
$$\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$$

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

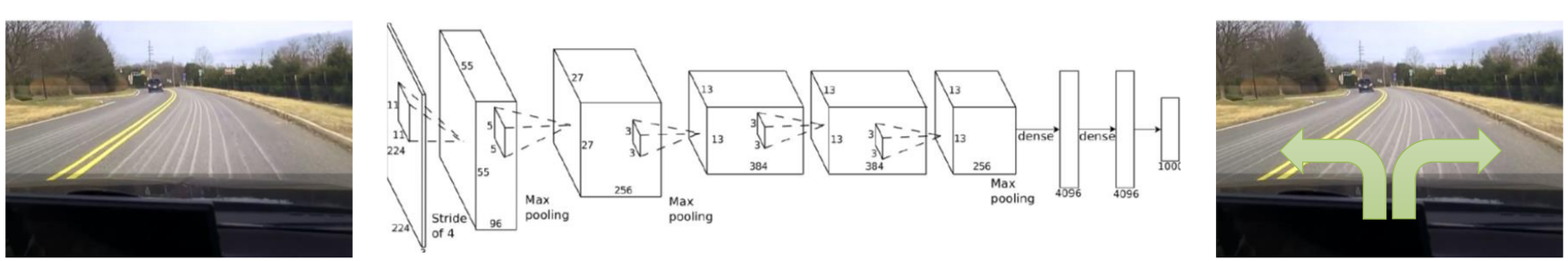


Policy π

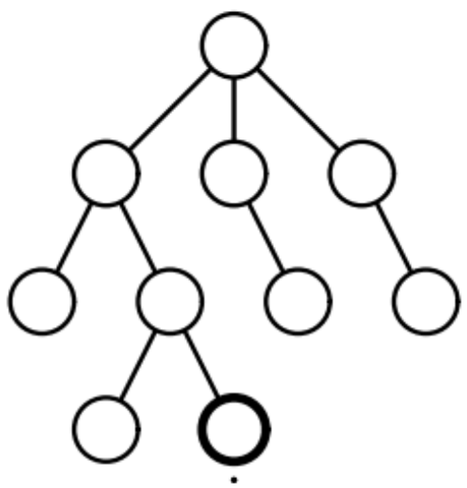
A mapping function from states to actions of the end effectors.

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

It can be a shallow or deep function mapping



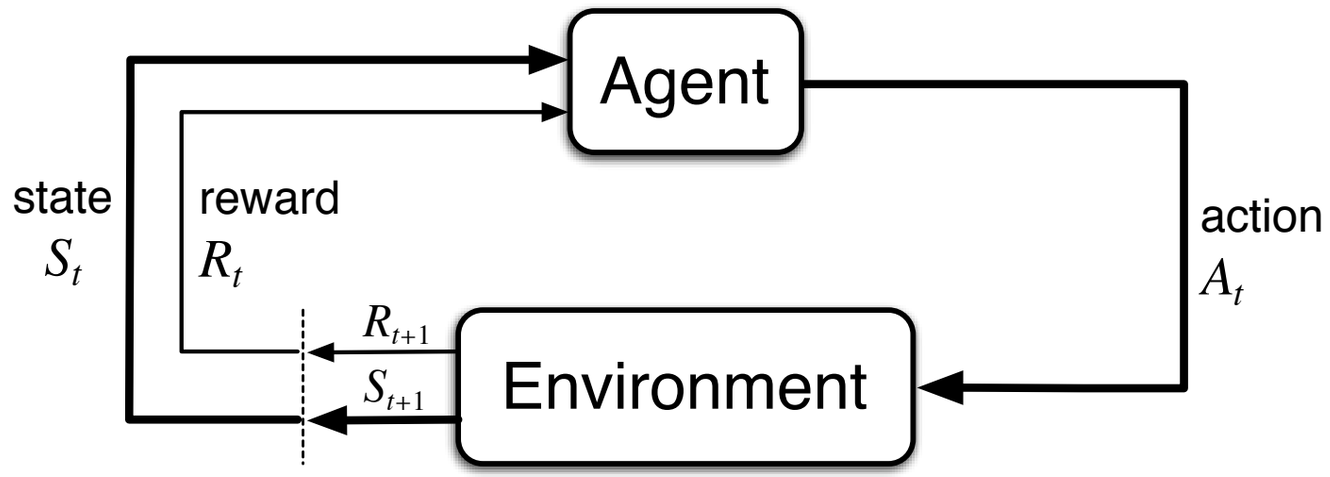
or it can be as complicated as involving a tree look-ahead search



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 $\mathbf{R}_t$

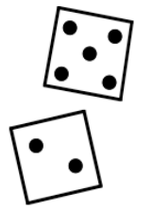
They are scalar values provided by the environment to the agent that indicate whether goals have been achieved, e.g., ***1 if goal is achieved, 0 otherwise, or -1 for overtime step the goal is not achieved***

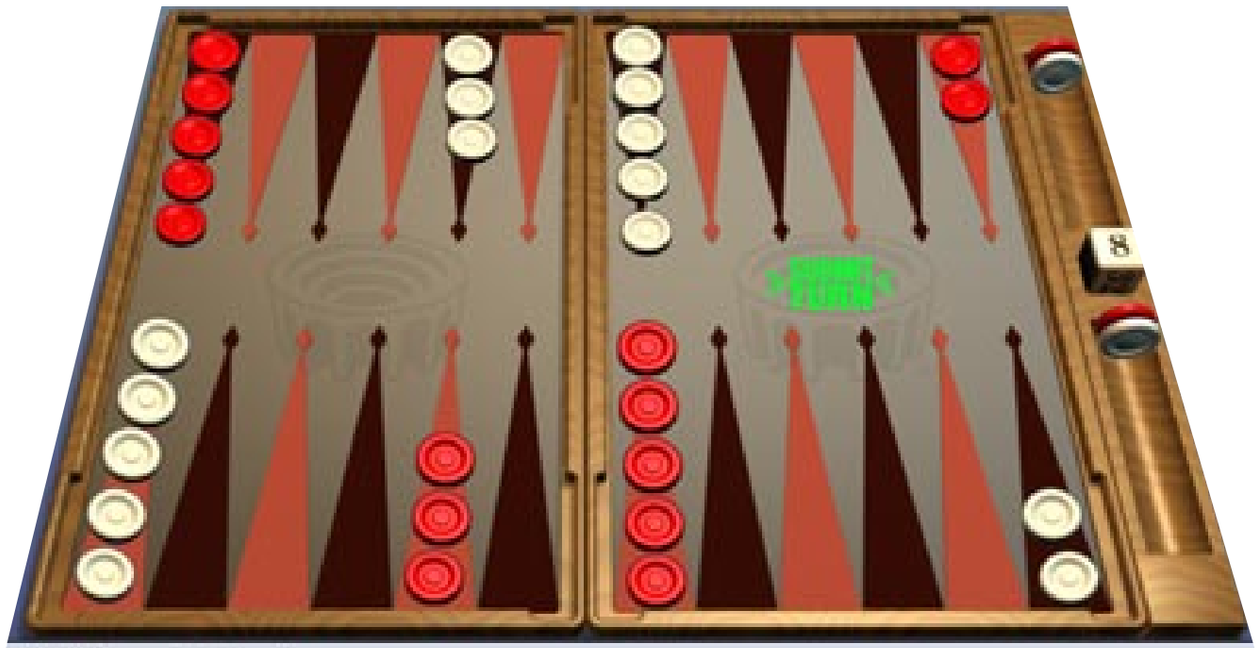
- Rewards specify what the agent needs to achieve, not how to achieve it.

- 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



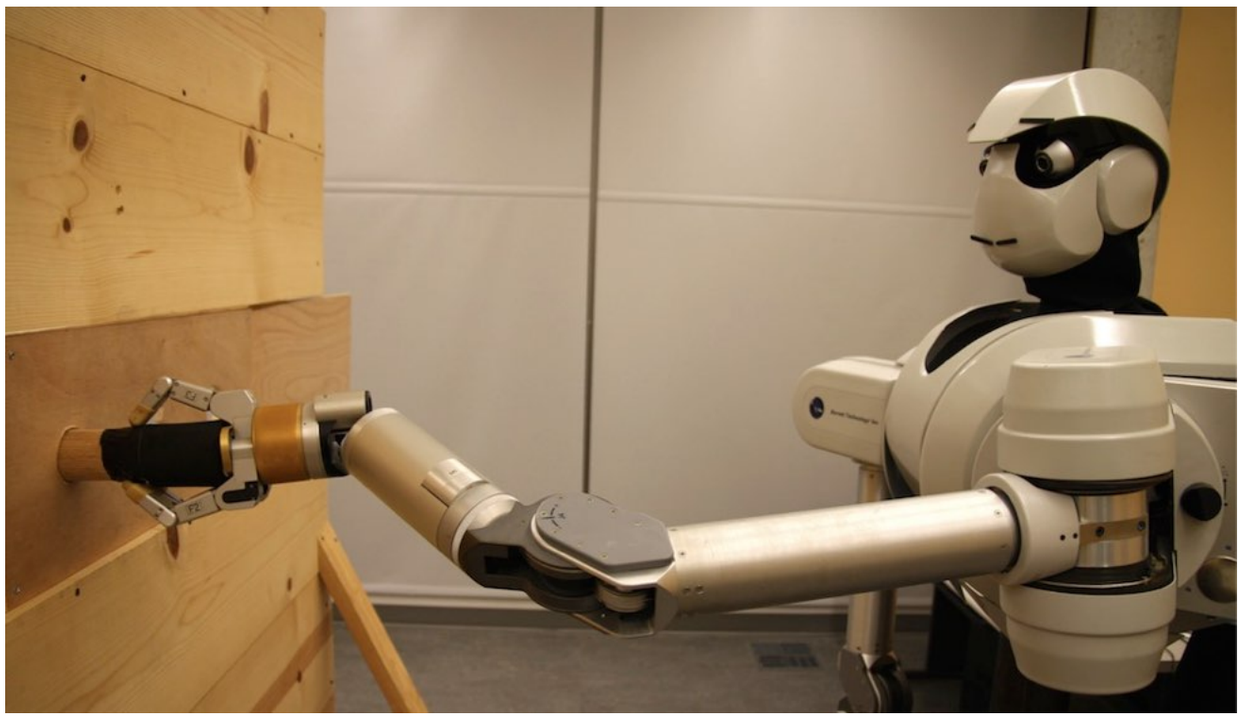


Learning to Drive

- 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

Peg in Hole Insertion Task

- States: Joint configurations (7DOF)
- Actions: Torques on joints
- Rewards: Penalize jerky motions, reaching target pose
- Q: why we do not simply use Euclidean distances between current and target poses/locations?



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

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

- Transition function or next step function:

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

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

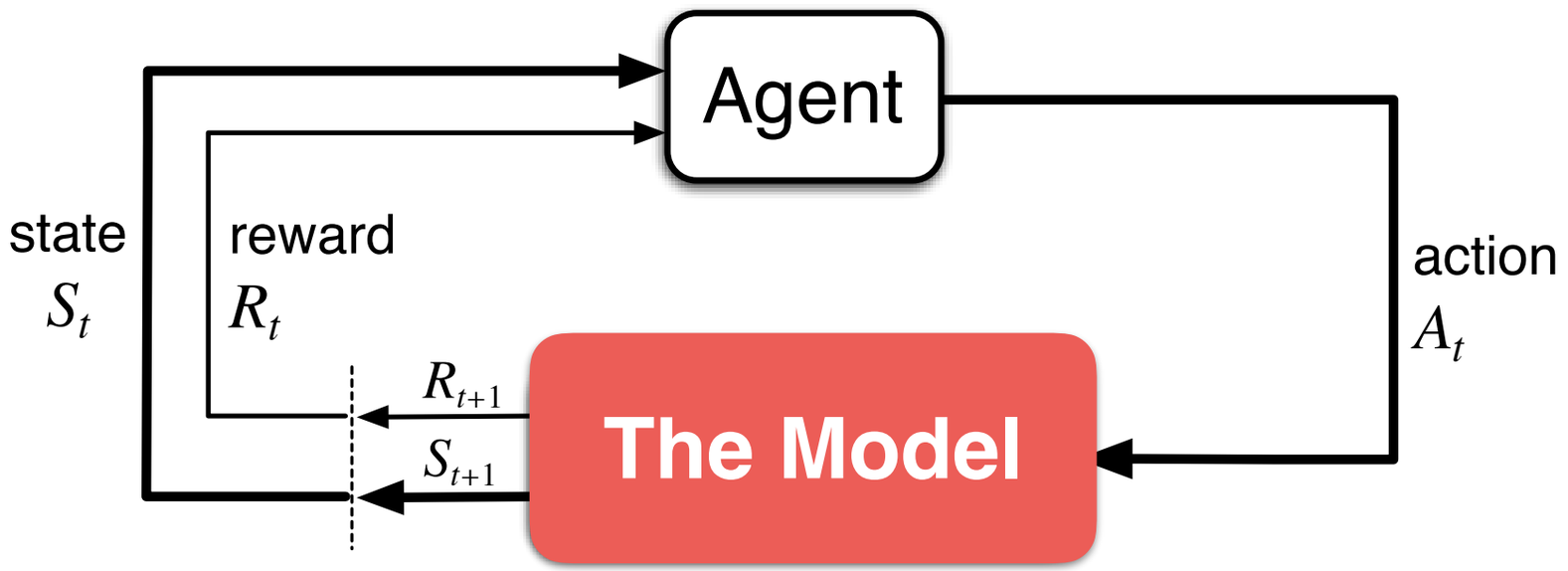
Rajesh P. N. Rao[1] and Dana H. Ballard[2]

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

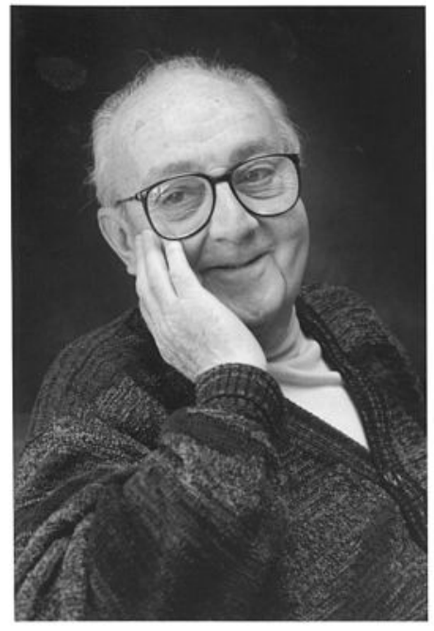
Plan: a sequence of actions



Model-free VS model-based RL

- An estimated (learned) model is never perfect.

  ``All models are wrong but some models are useful”



George Box

- Due to model errors, model-free methods often achieve better policies though need more interactions with the environment.

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

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

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

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

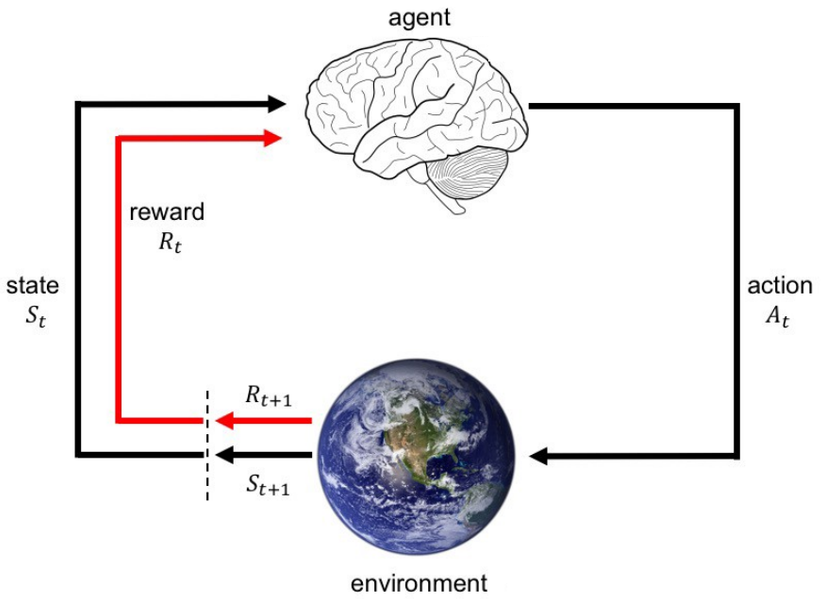
Value functions encode our wisdom

q_π(day before HW is due, start working on HW) = ?

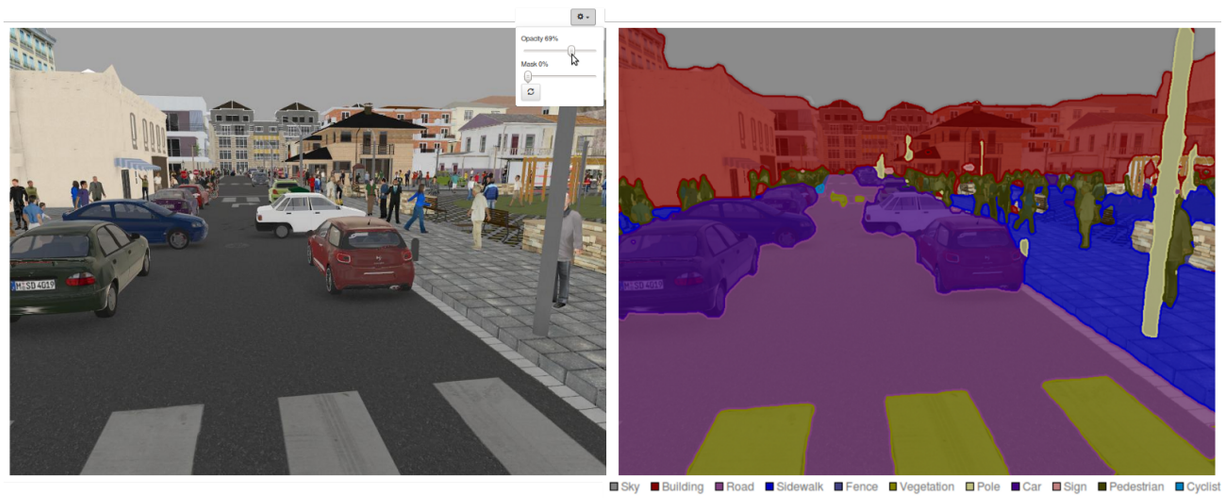
Reinforcement learning (and why we like it)

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



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 each state is for the goal she is trying to achieve.

Other forms of supervision for learning behaviours?

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

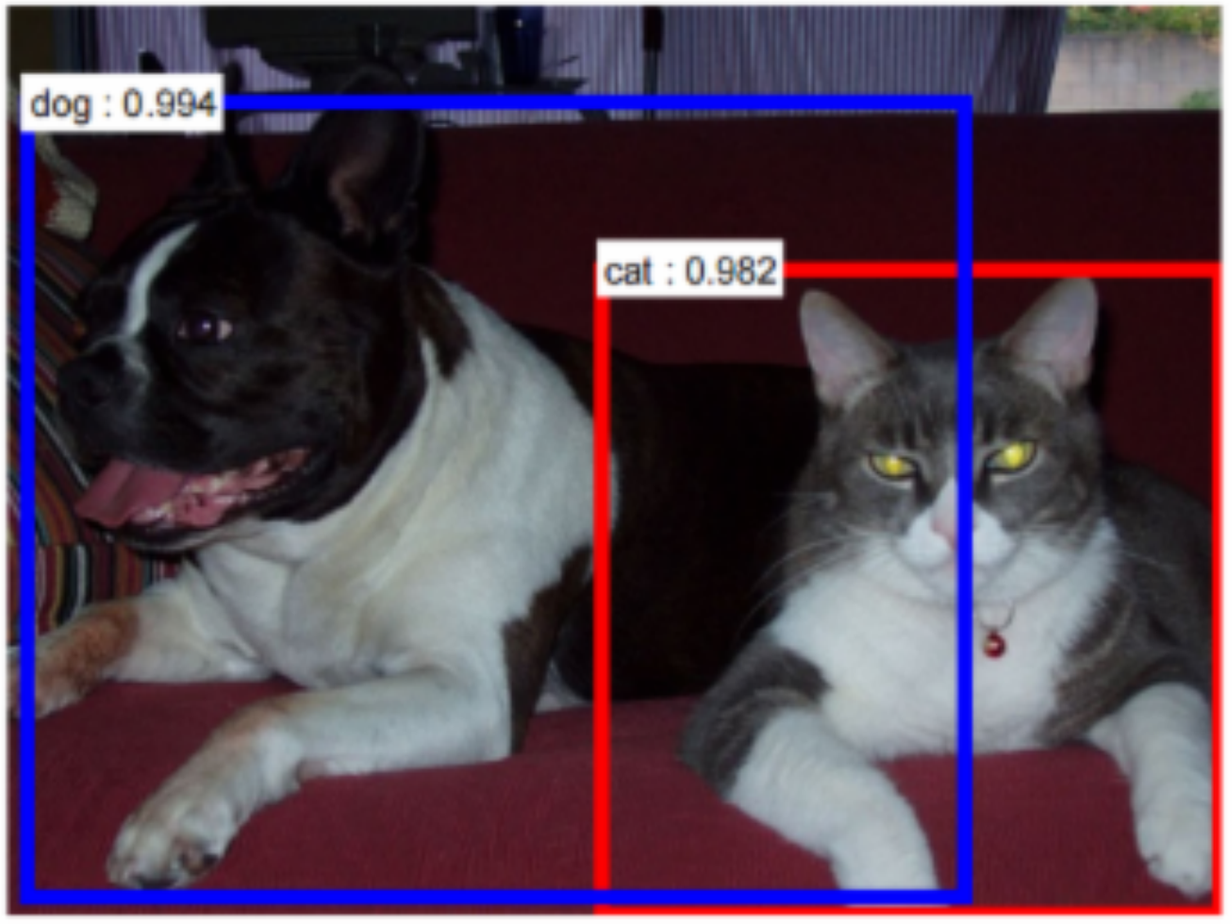
   It was way easier for athletes to perfection the jump, once someone showed the right general trajectory

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?



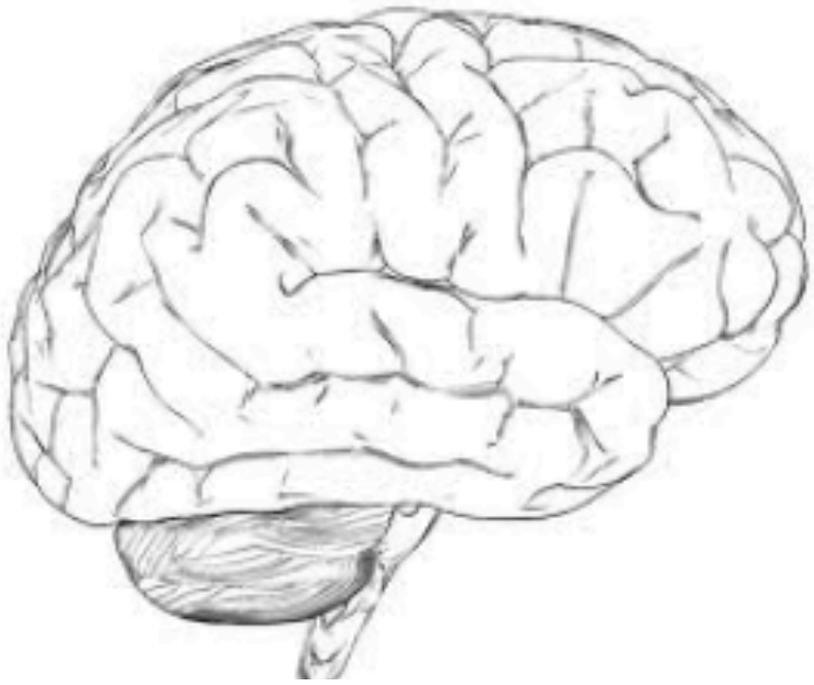
RL Versus ML

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

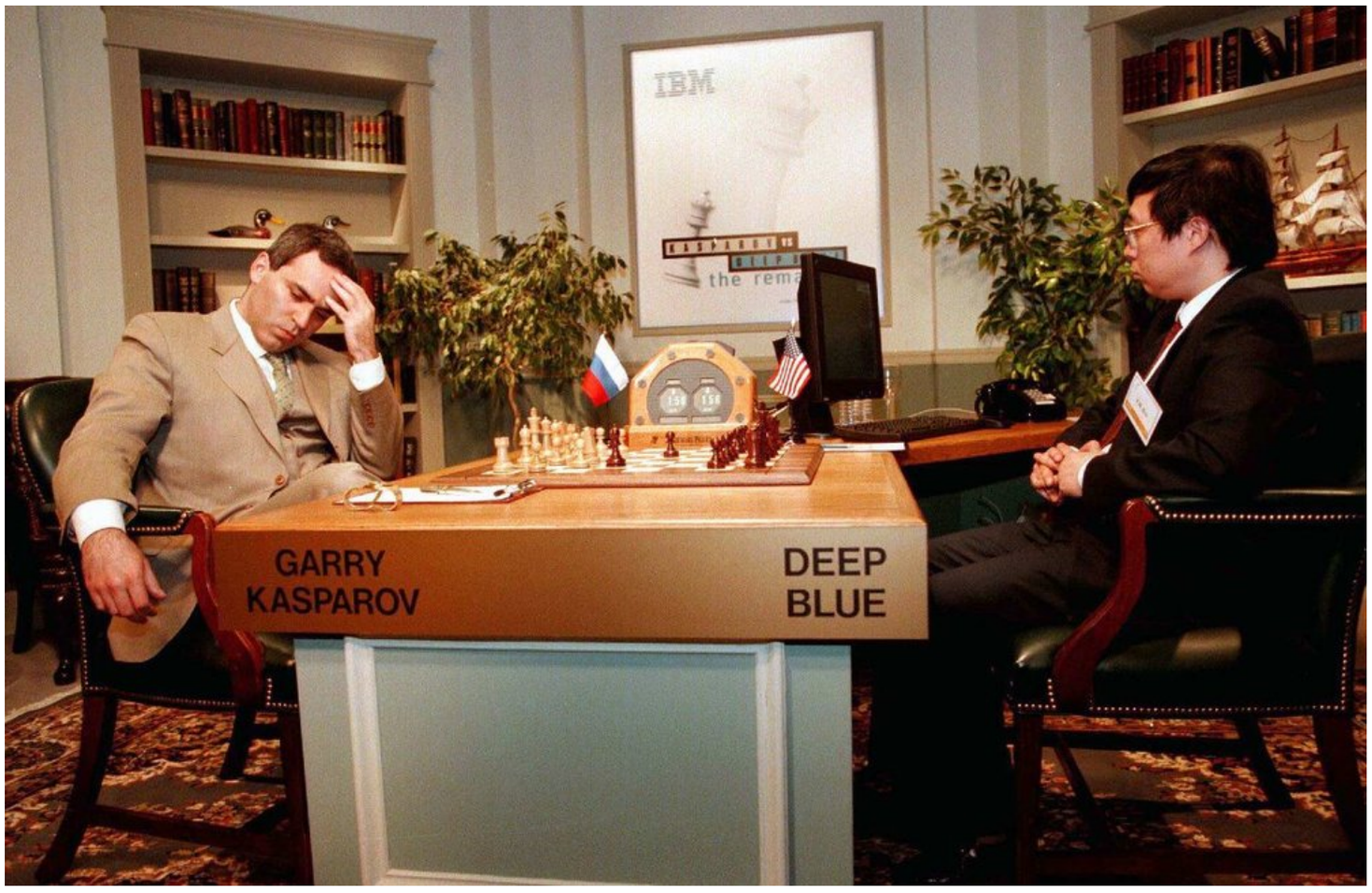
Learning to Act

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
4) We cannot use gradients as easily..

Successes so far (for learning policies by imitation and reinforcement)

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

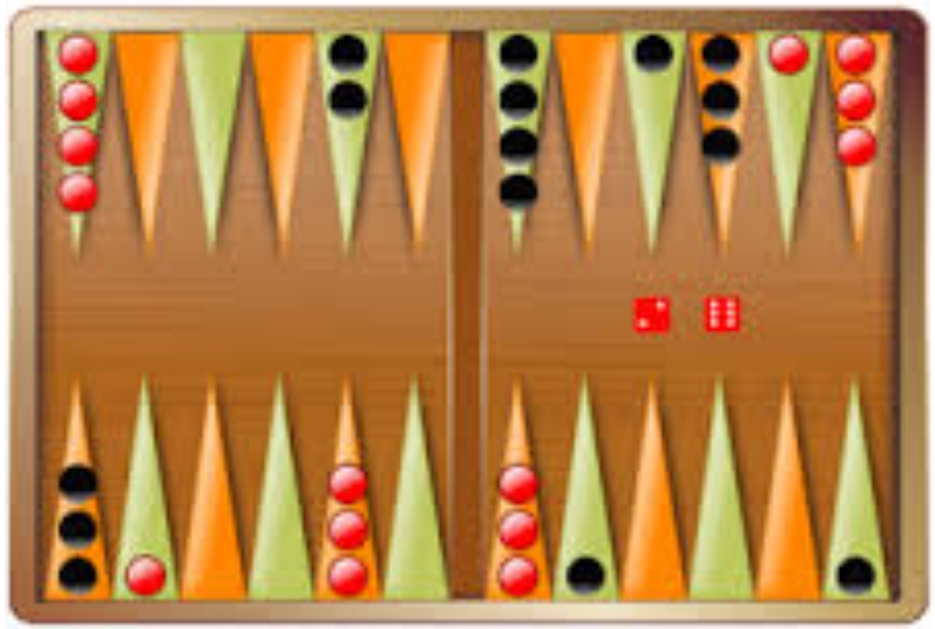
Backgammon

How is it different than chess?

Backgammon

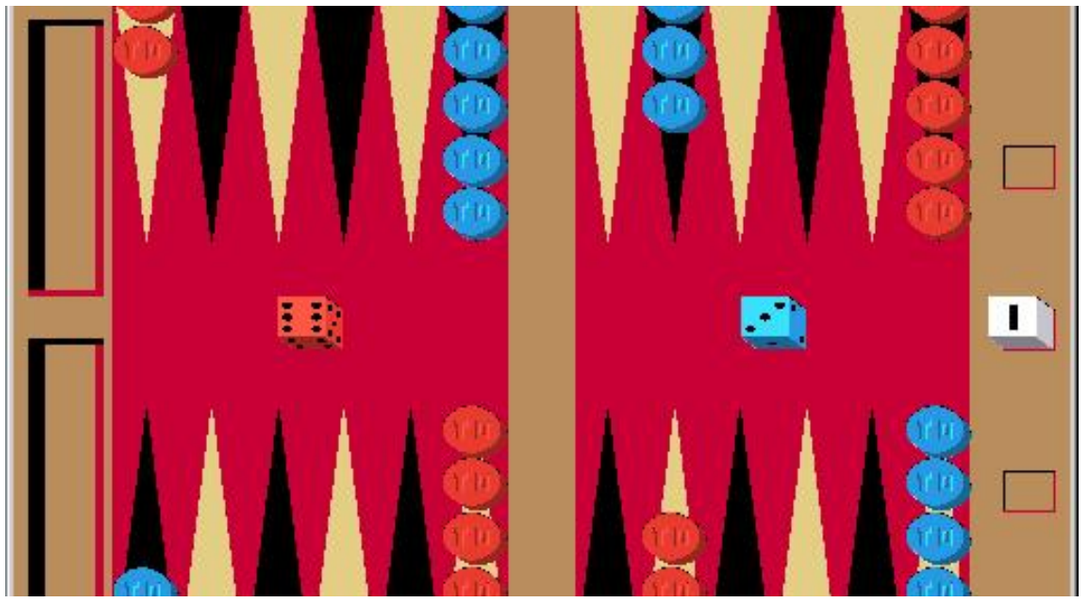
High branching factor due to dice roll prohibits brute force deep searches such as in chess

Neuro-Gammon



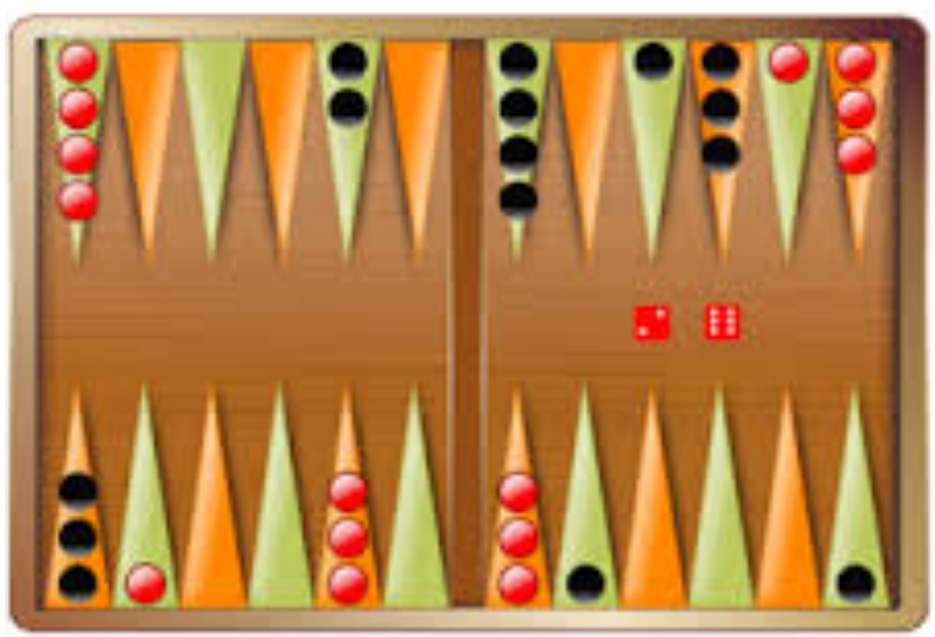
- Developed by Gerald Tesauro in 1989 in IBM’s research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

TD-Gammon



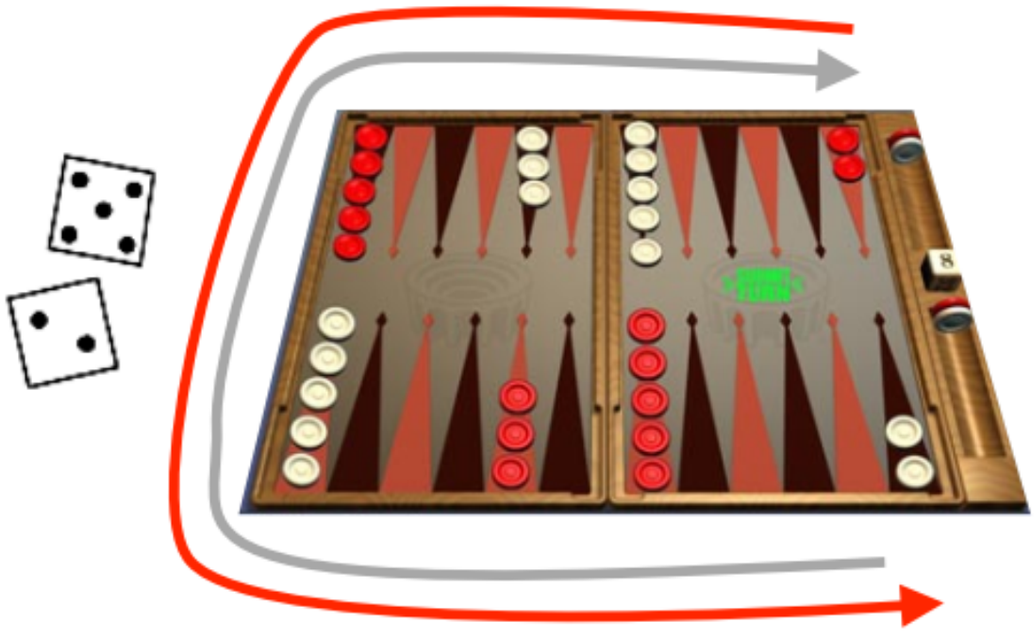
- Developed by Gerald Tesauro in 1992 in IBM’s research center
- A neural network that trains itself to be an evaluation function by playing against itself starting from random weights
- Achieved performance close to top human players of its time

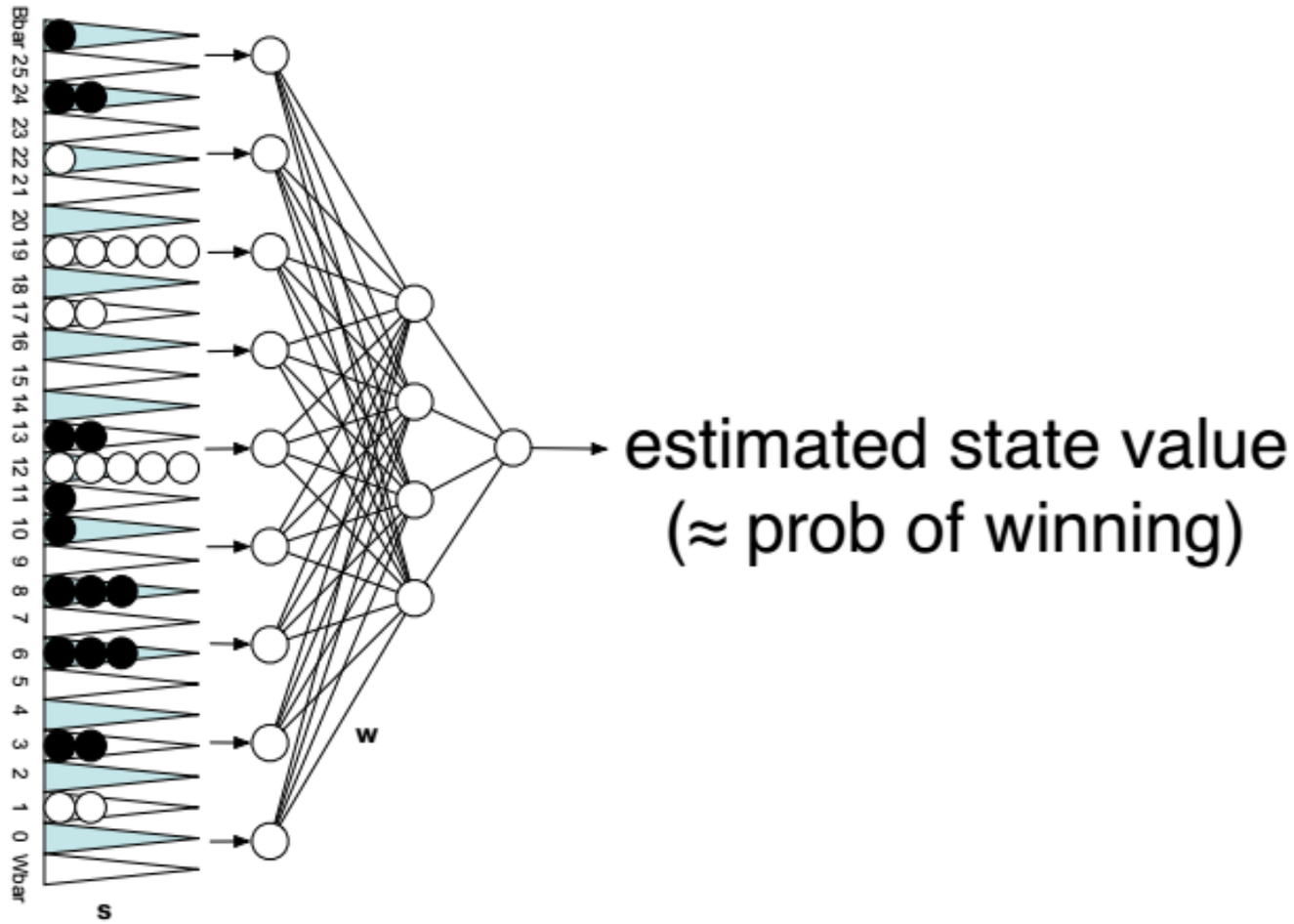
Neuro-Gammon



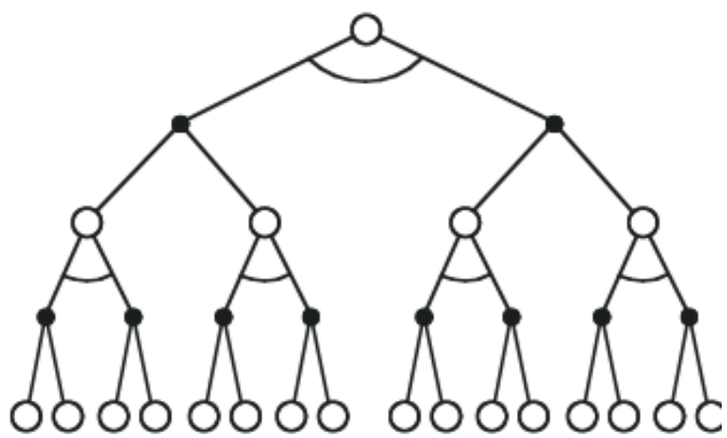
- Developed by Gerald Tesauro in 1989 in IBM’s research center
- Trained to mimic expert demonstrations using supervised learning
- Achieved intermediate-level human player

Evaluation function





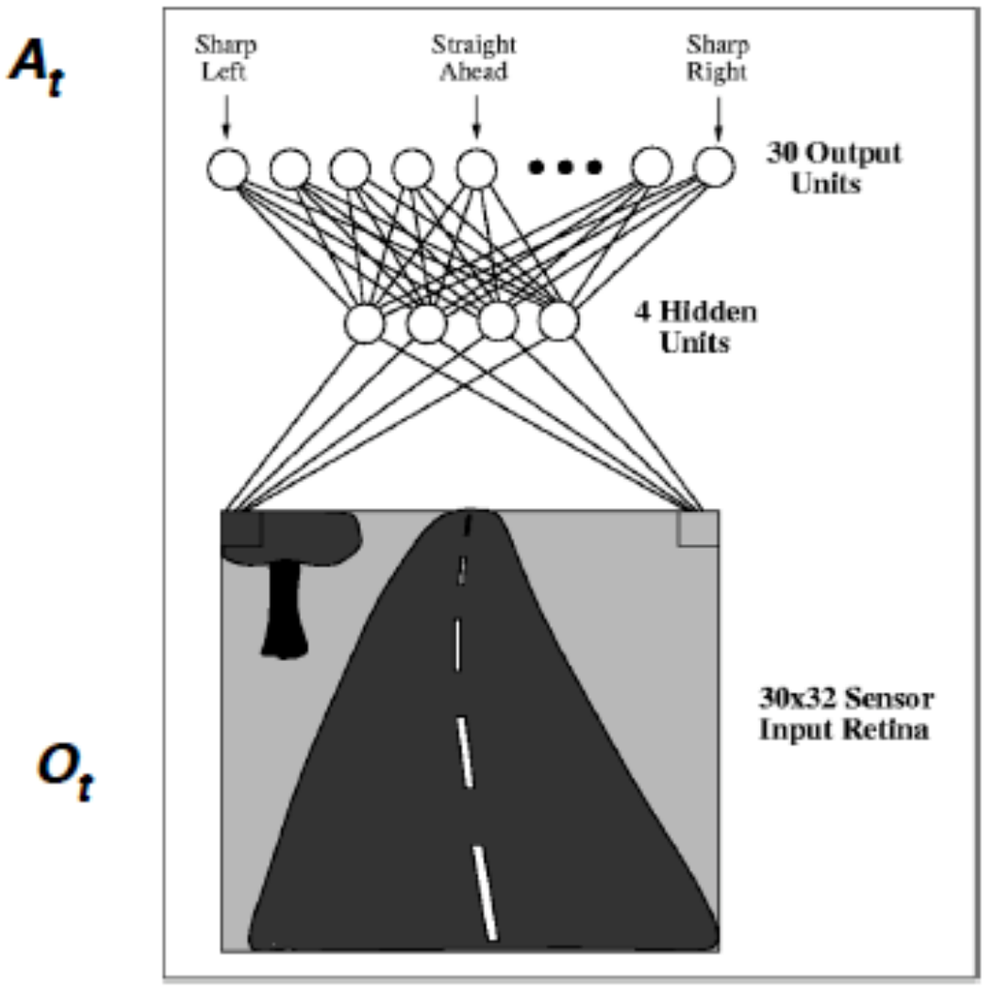
Action selection
by a shallow search



Self-Driving Cars

Self-Driving Cars

Policy network π:
mapping of
observations to actions



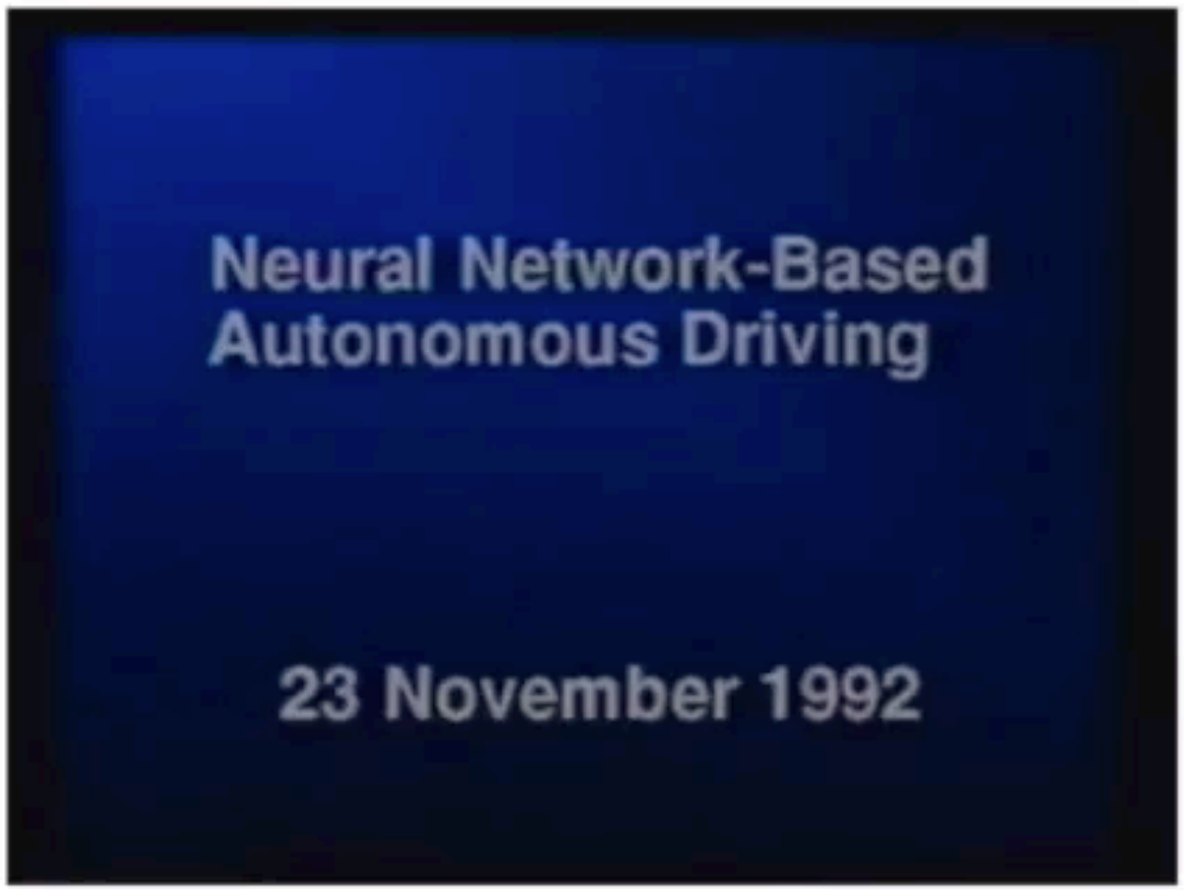
1989

ALVINN, an autonomous land vehicle in a neural network

Dean A. Pomerleau
Carnegie Mellon University

Self-Driving Cars

- ALVINN video



[Courtesy of Dean Pomerleau]

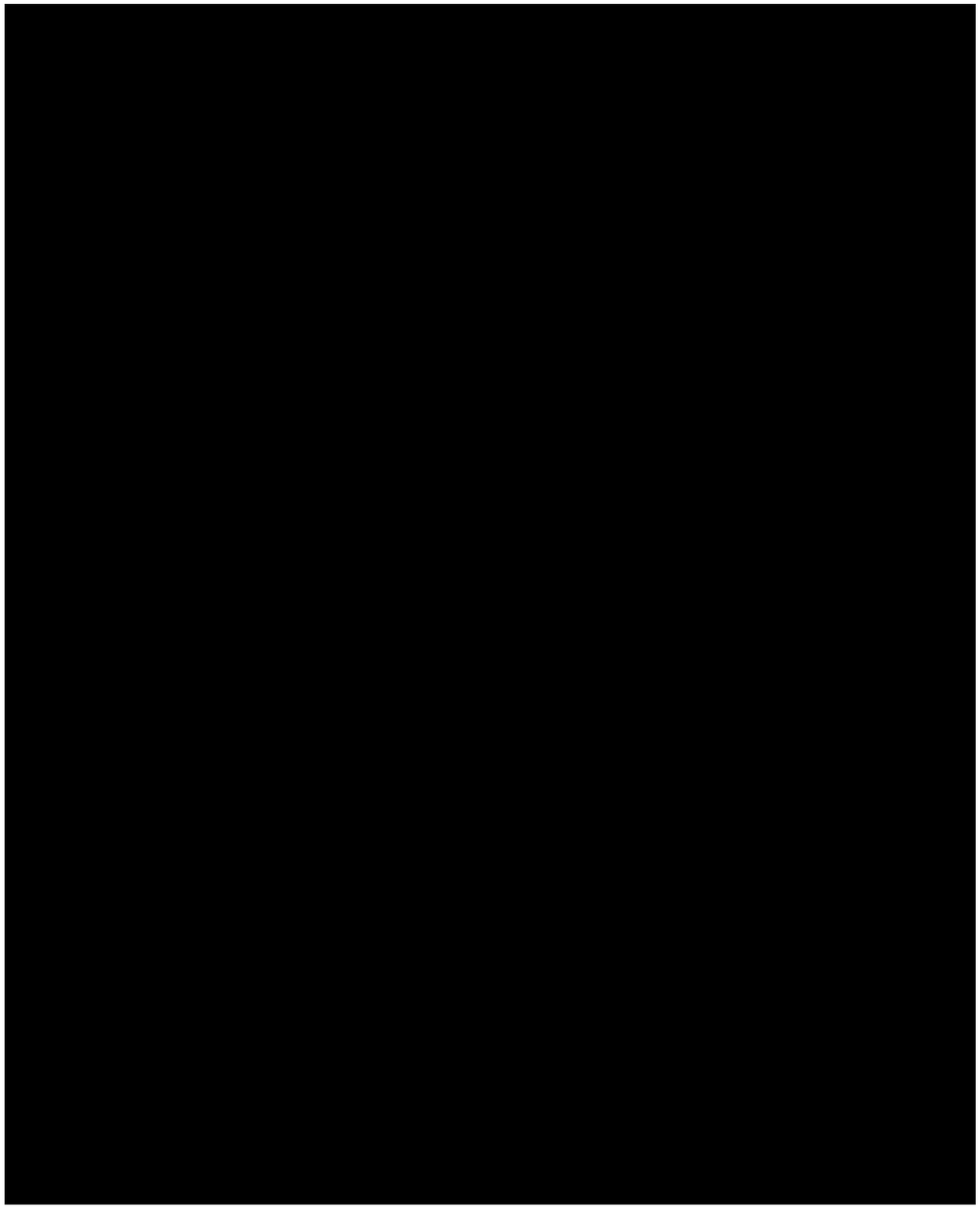
- behavior cloning- learning from the human driver
- data augmentation to deal with compounding errors

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

GO

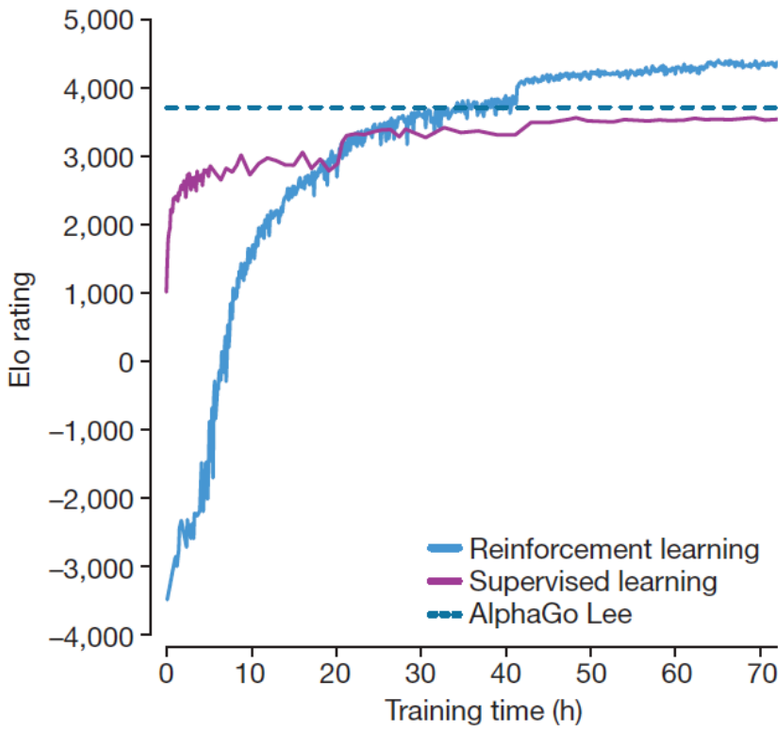
AlphaGo

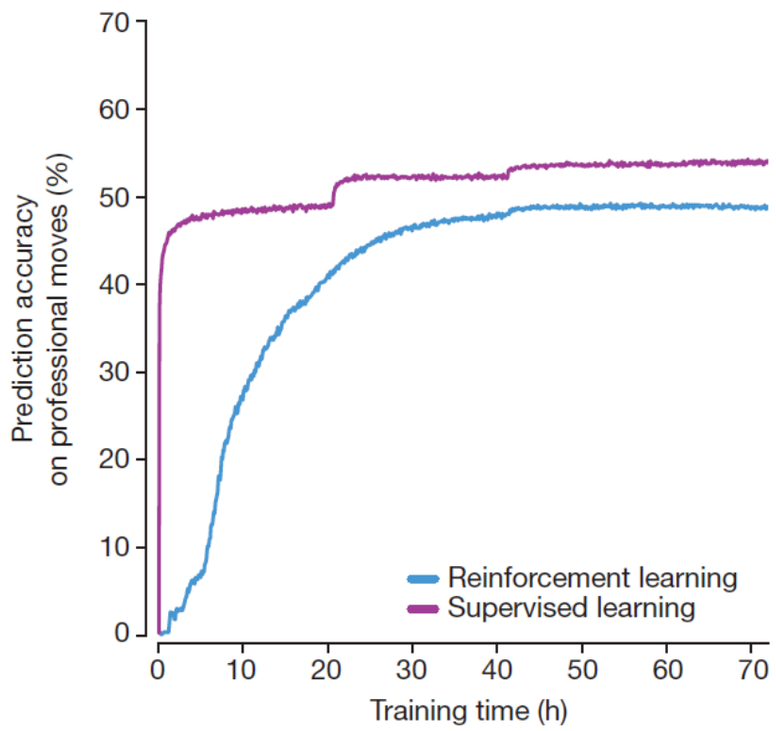
- Monte Carlo Tree Search with neural nets
- expert demonstrations
- self play

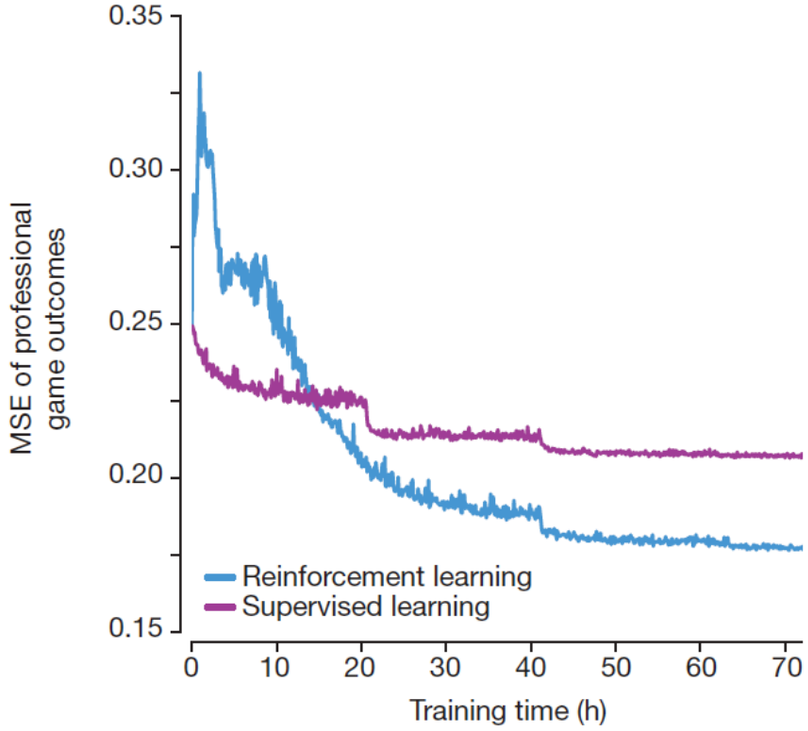
AlphaGo

Tensor Processing Unit from Google

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

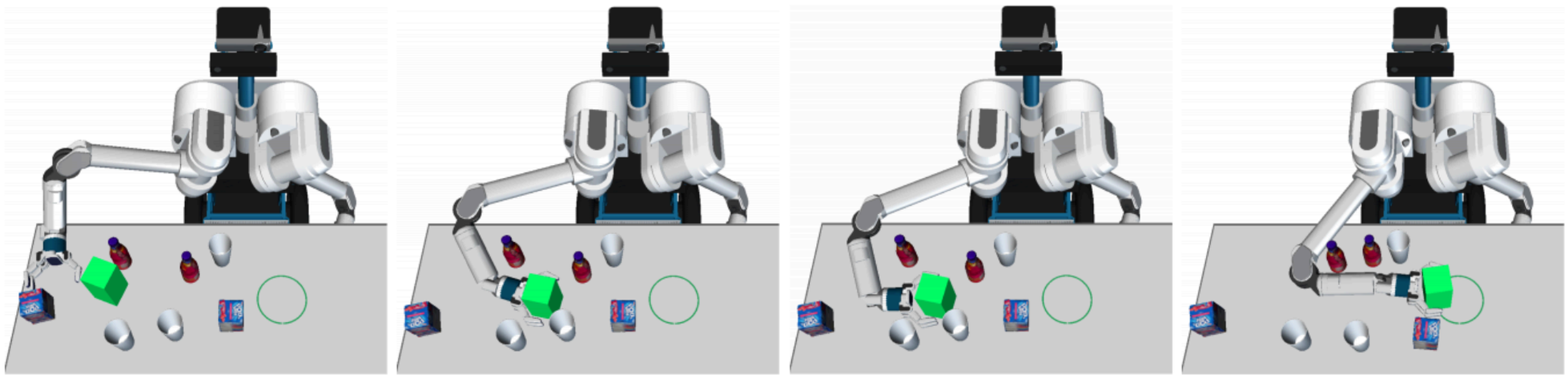
How the world of Alpha Go is different than the real world?

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 representations - Two extremes

- **Assuming we know everything about the objects (object locations, 3D shapes, physical properties).** Use planners to search for the action sequence to achieve a desired goal.

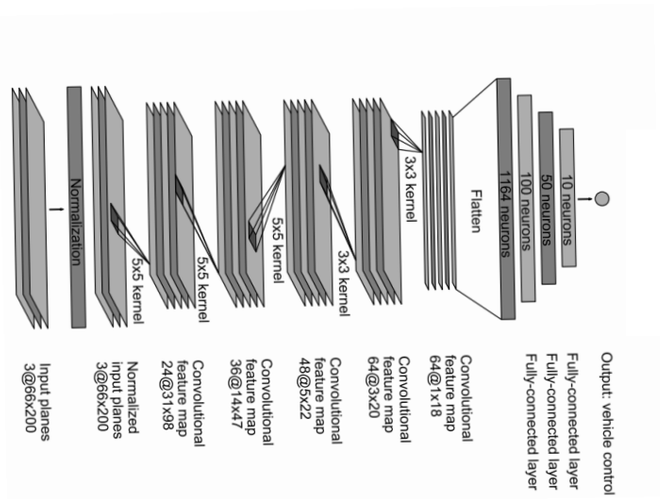


Rearrangement Planning via Heuristic Search, Jennifer E. King, Siddhartha S. Srinivasa

State representations - Two extremes

- **Assuming we know everything about the objects (object locations, 3D shapes, physical properties).** Use planners to search for the action sequence to achieve a desired goal.

- **Assuming we know nothing about the objects.** 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

https://cmudeeprl.github.io/703website/lectures/

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?

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.

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
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."

No cognition. Just sensing and action

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