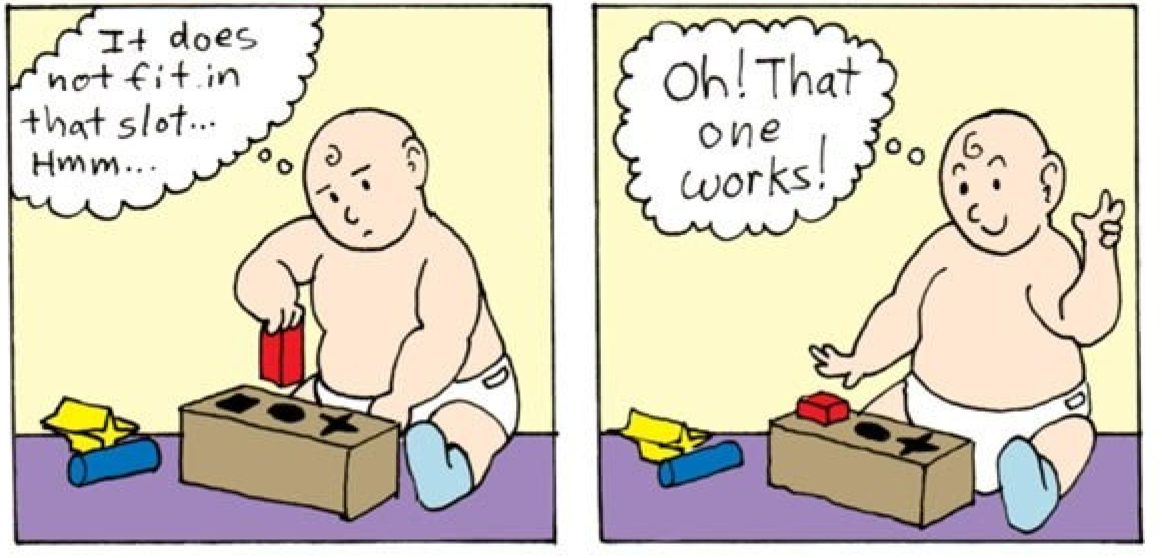
Introduction to RL

Robert Platt
Northeastern University

(some slides/material borrowed from Rich Sutton)

What is reinforcement learning?

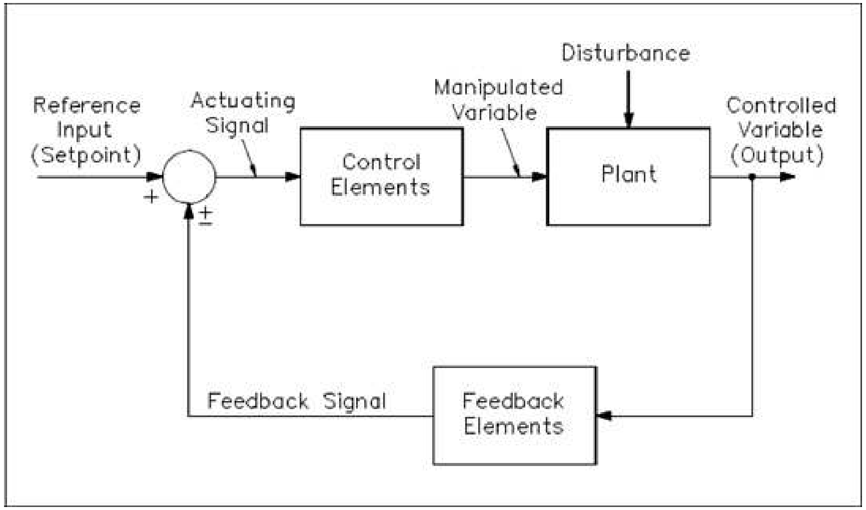
RL is learning through trial-and-error without a model of the world



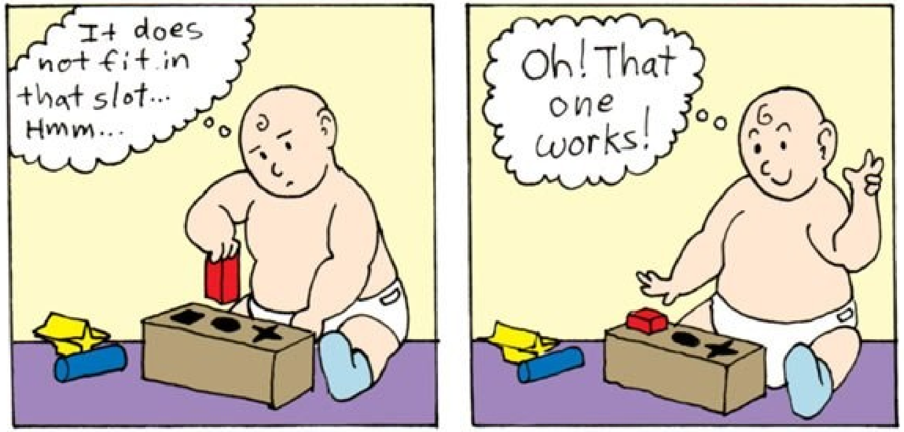
What is reinforcement learning?

RL is learning through trial-and-error without a model of the world

This is different from standard control/planning systems:



vs



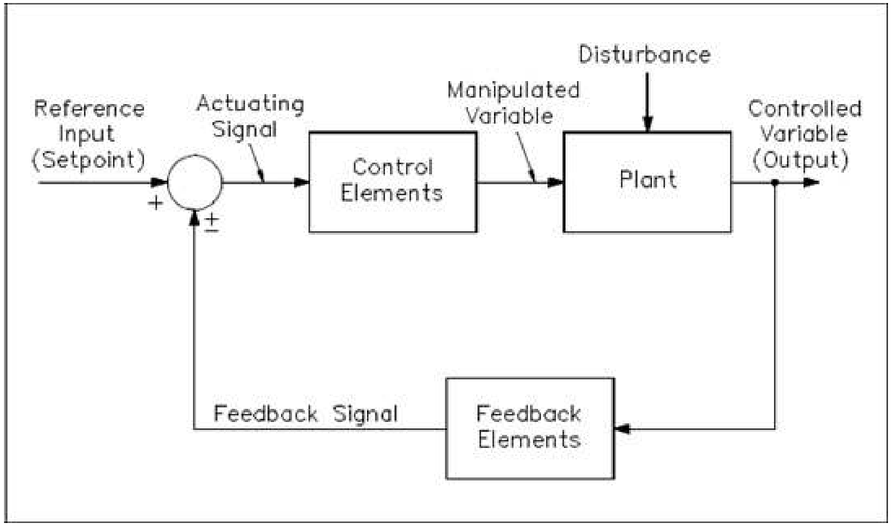
Standard control system

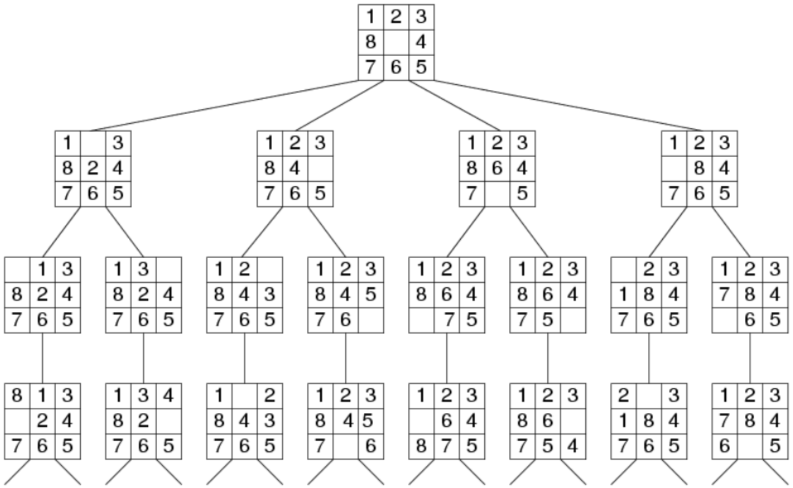
Reinforcement learning

What is reinforcement learning?

RL is learning through trial-and-error without a model of the world

This is different from standard control/planning systems:
- require a model of the world
  - i.e. you need to hand-code the “successor function”
- often require the world to be expressed in a certain way
  - e.g. symbolic planners assume symbolic representation
  - e.g. optimal control assume algebraic representation





What is reinforcement learning?

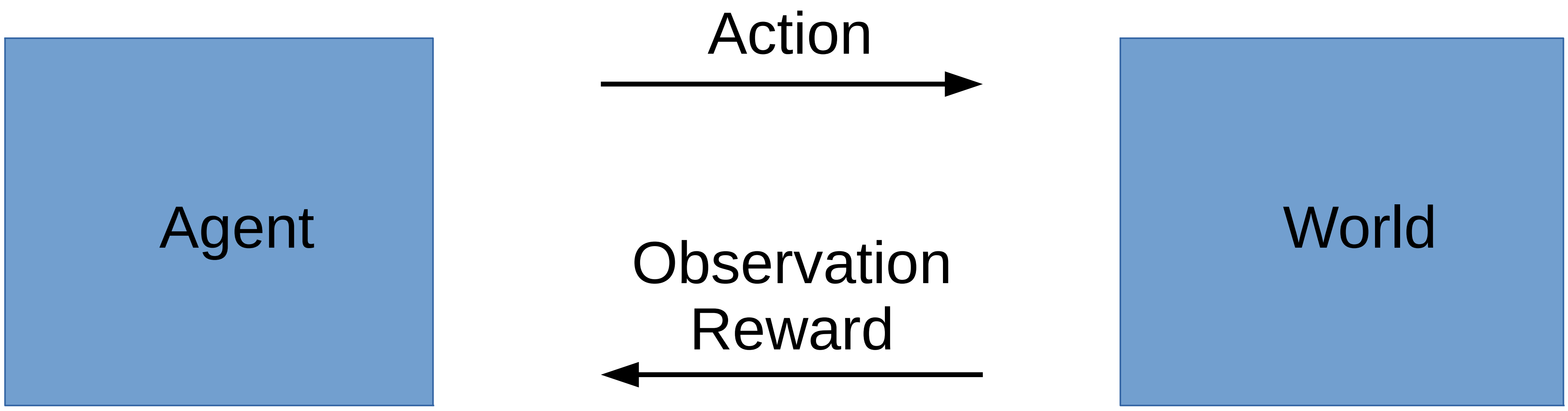
RL is learning through trial-and-error without a model of the world

This is different from standard control/planning systems:

- require a model of the world
  - i.e. you need to hand-code the “successor function”
- often require the world to be expressed in a certain way
  - e.g. symbolic planners assume symbolic representation
  - e.g. optimal control assume algebraic representation

RL doesn’t require any of this
RL intuitively resembles natural learning
RL is *harder* than planning b/c you don’t get the model
RL can be less efficient that control/planning b/c of its generality

The RL Setting



On a single time step, agent does the following:

1. observe some information
2. select an action to execute
3. take note of any reward

Goal of agent: select actions that maximize cumulative reward in the long run

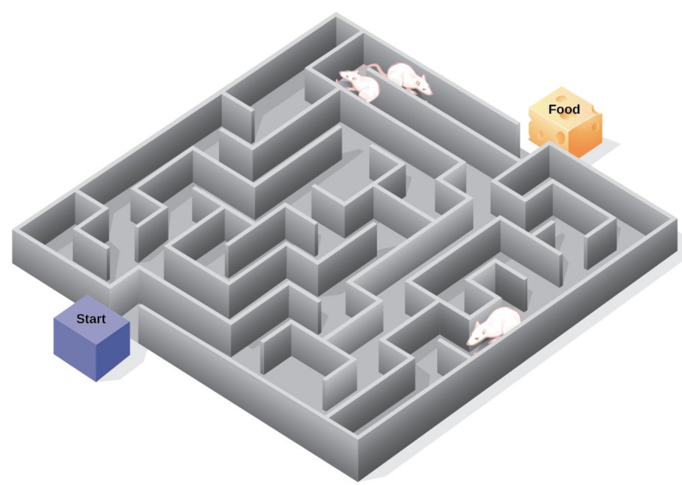
Example: rat in a maze

Agent

Move left/right/up/down

Observe position in maze
Reward = +1 if get cheese

World



Goal: maximize cheese eaten

Example: robot makes coffee

Agent

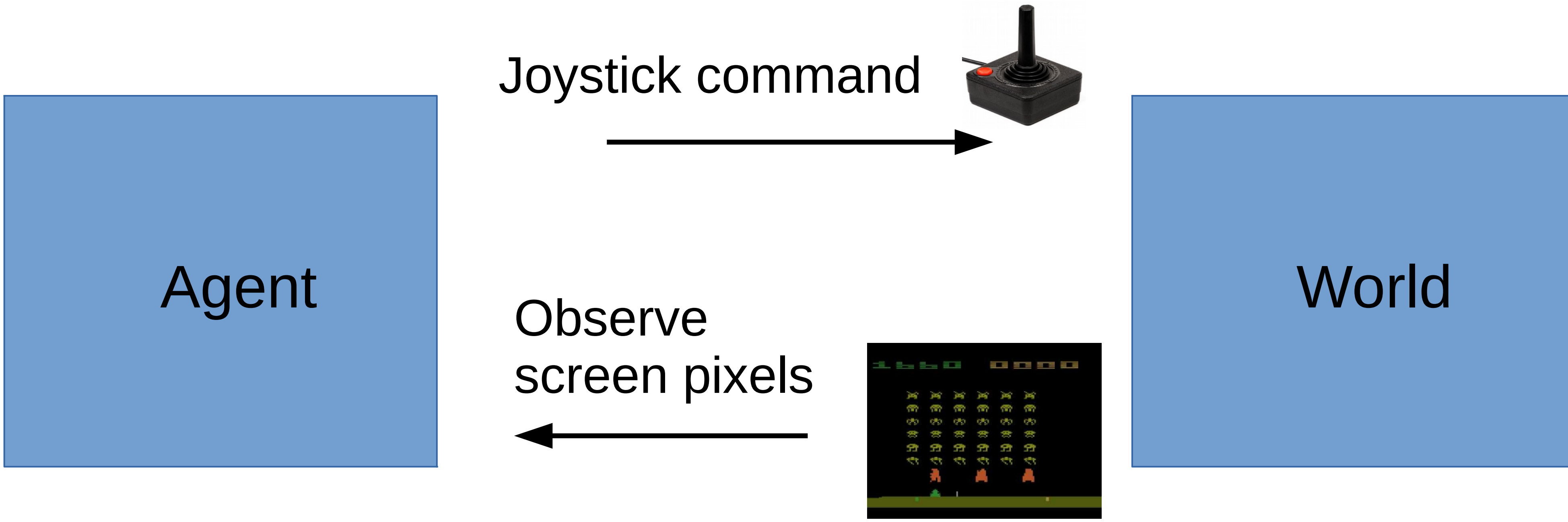
Move robot joints →

Observe camera image
Reward = +1 if coffee in cup ←

World

Goal: maximize coffee produced

Example: agent plays pong



Reward = game score

Goal: maximize game score

Think-Pair-Share Question

How would you express the problem of playing online texas hold-em as an RL problem?

Action = ?

Agent

World

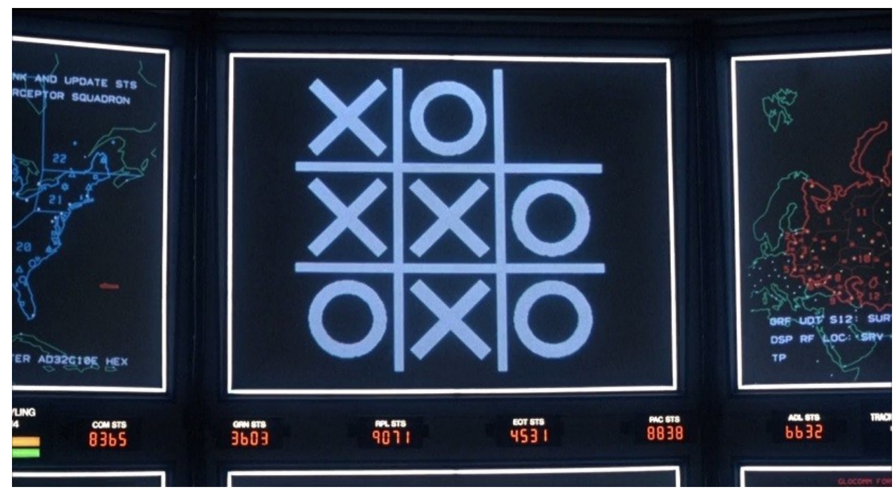
Observation = ?
Reward = ?

Goal: ?

RL example

Let’s say you want to program the computer to play tic-tac-toe

How might you do it?



RL example

Let's say you want to program the computer to play tic-tac-toe

How might you do it?

1. <u>search</u>:
   – mini-max tree search
   – plans for the optimal opponent, not actual opponent

2. <u>evolutionary computation</u>:
   – start w/ population of random policies; have them play each other
   – can view this as hillclimbing in policy space wrt a fitness function

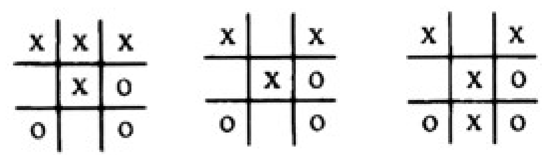
RL example

Let’s say you want to program the computer to play tic-tac-toe

How might you do it?

3. RL:

Value function:
– estimate value function *V(s)* over states *s*

– examples of states:  …

s_k s_{k+1} s_{k+2}

– *V(s)* denotes expected reward from state *s* (+1 win, -1 lose, 0 draw)

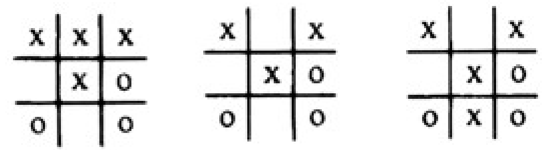
Game play:
– the agent selects actions that lead to states with high values, *V(s)*
– the agent gradually gets lots of experience of the results of executing various actions from different states

But how estimate value function?

RL example

<u>Value function:</u>
– estimate value function *V(s)* over states *s*

– examples of states:



s_k s_{k+1} s_{k+2} …

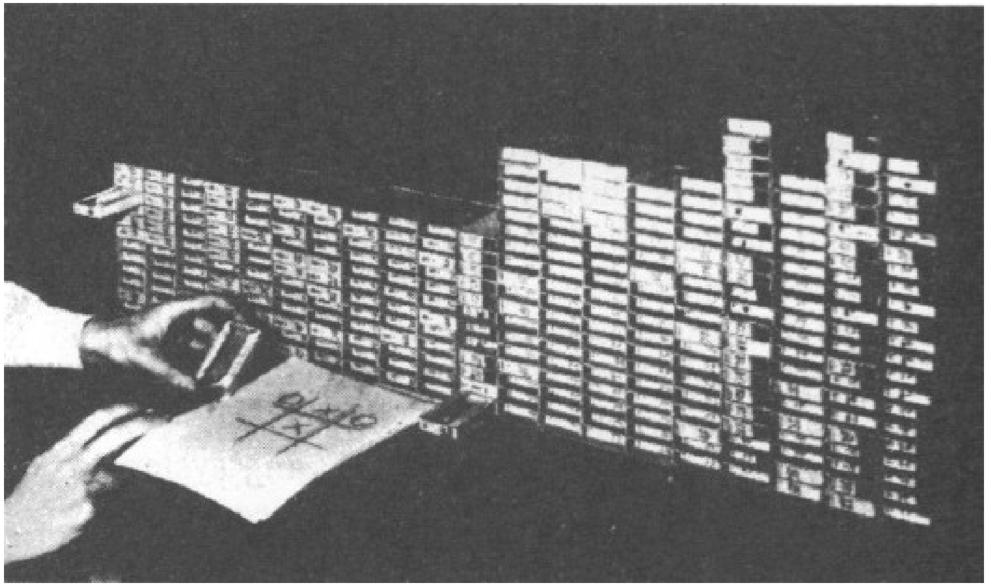
– *V(s)* denotes expected reward from state *s* (+1 win, -1 lose, 0 draw)

<u>Game play:</u>
– the agent selects actions that lead to states with high values, *V(s)*
– the agent gradually gets lots of experience of the results of executing various actions from different states

<u>But how estimate value function?</u>

$$V(s_t) \leftarrow (1-\alpha)V(s_t) + \alpha(R_t + V(s_{t+1}))$$

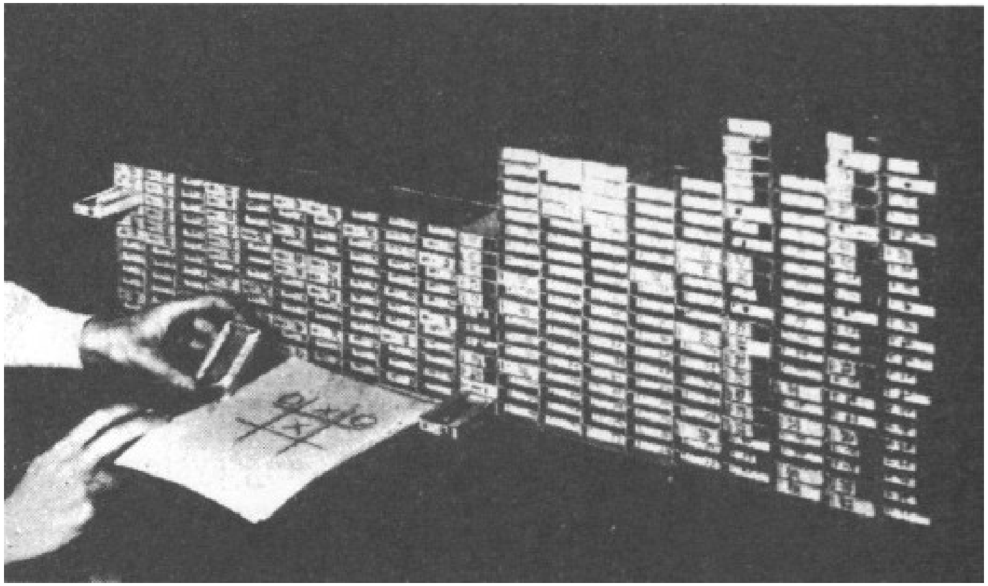
RL example: MENACE



Donald Michie teaching MENACE to play tic-tac-toe (1960)

<u>Can a “machine” comprised only of matchbooks learn to play tic-tac-toe?</u>

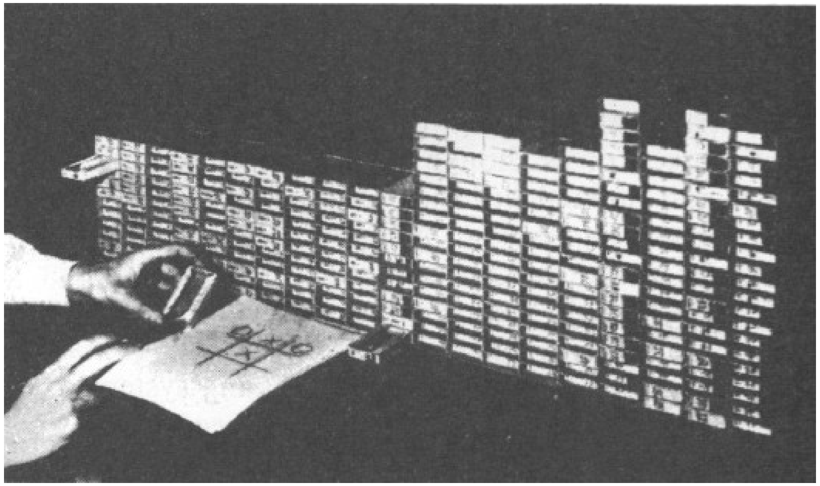
RL example: MENACE



Donald Michie teaching MENACE to play tic-tac-toe (1960)

<u>Can a “machine” comprised only of matchbooks learn to play tic-tac-toe?</u>

RL example: MENACE



How it works:

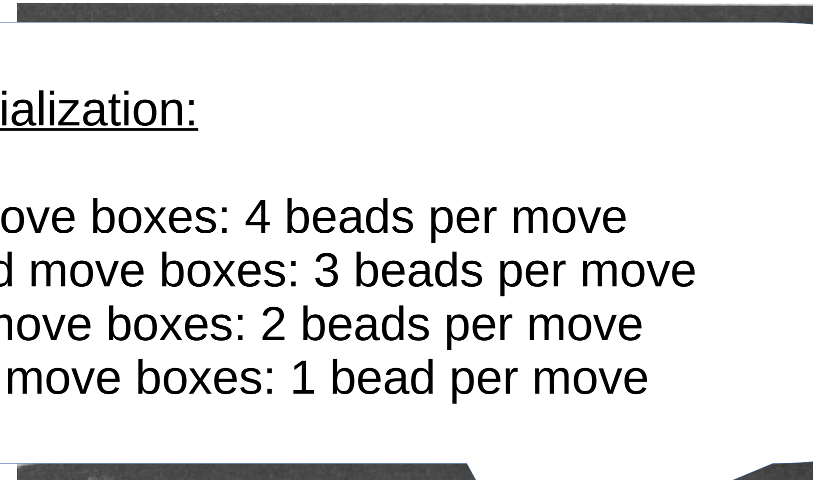
Gameplay:

– each tic-tac-toe board position corresponds to a matchbox
– at the beginning of play, each matchbox is filled will beads of different colors
– there are nine bead colors: one for each board position
– when it is MENACE's turn, open drawer corresponding to board configuration and select a bead randomly. Make the corresponding move. Leave bead on table and leave matchbox open.

Reward:

– play an entire game to its conclusion until it ends: win/lose/draw
– if MENACE loses the game, remove beads from table and throw them away
– if MENACE draws, replace each bead back into the box it came from. Add an extra bead of the same color to each box.
– if MENACE wins, replace each bead back into the box it came from. Add **THREE** extra beads of the same color to each box.

RL example: MENACE



Bead initialization:

– First move boxes: 4 beads per move
– Second move boxes: 3 beads per move
– Third move boxes: 2 beads per move
– Fourth move boxes: 1 bead per move

How it works:

Gameplay:
– each tic-tac-toe board position corresponds to a matchbox
– at the beginning of play, each matchbox is filled will beads of different colors
– there are nine bead colors: one for each board position
– when it is MENACE’s turn, open drawer corresponding to board configuration and select a bead randomly. Make the corresponding move. Leave bead on table and leave matchbox open.

Reward:
– play an entire game to its conclusion until it ends: win/lose/draw
– if MENACE loses the game, remove beads from table and throw them away
– if MENACE draws, replace each bead back into the box it came from. Add an extra bead of the same color to each box.
– if MENACE wins, replace each bead back into the box it came from. Add **THREE** extra beads of the same color to each box.

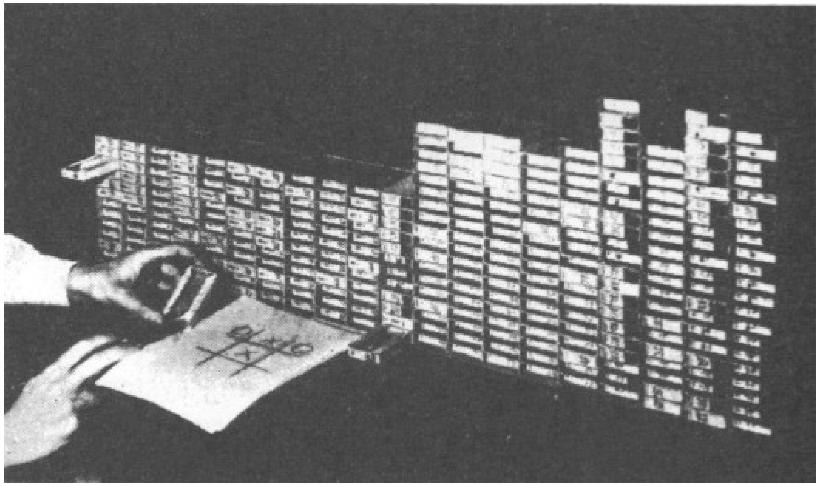
Think-Pair-Share Question

<u>Questions</u>:

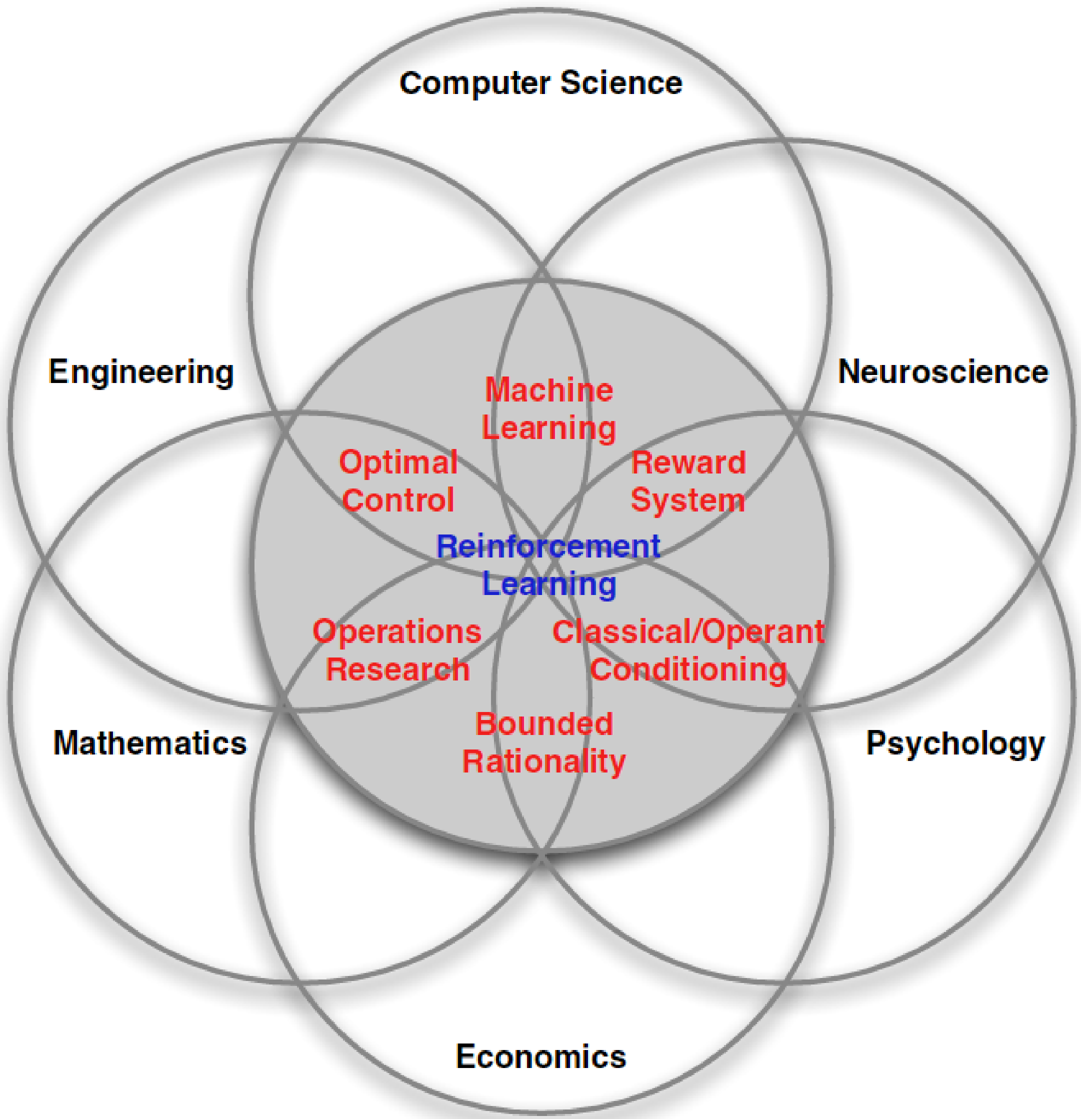
– why did Michie use that particular bead initialization?

– why add an extra bead when you get to a draw?

– how might this learning algorithm fail? How would you fix it?
  What tradeoff do you face?



Where does RL live?



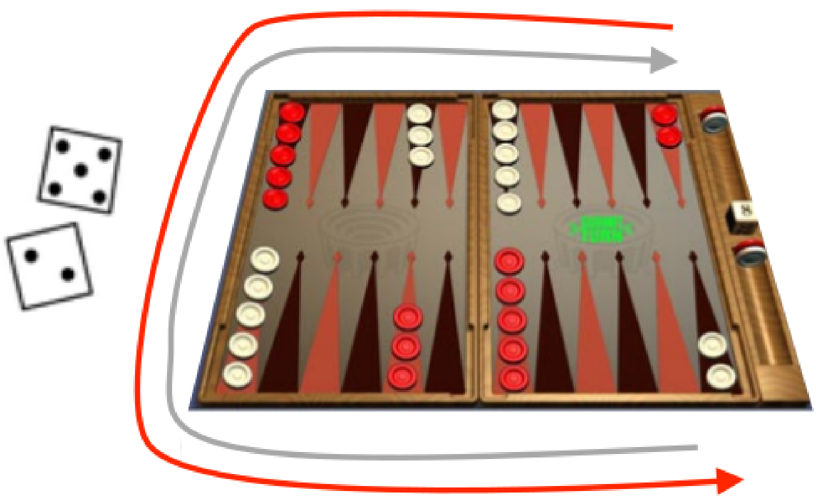
Key challenges in RL

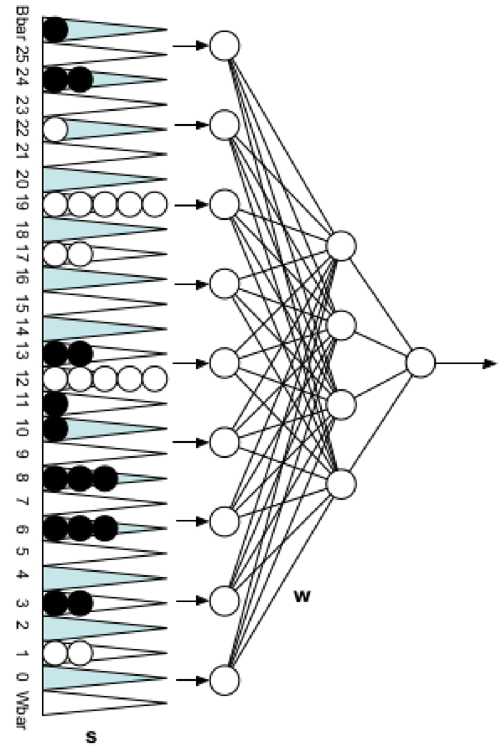
– no model of the environment
– agent only gets a scalar reward signal
– delayed feedback
– need to balance exploration of the world exploitation of learned knowledge
– real world problems can be non-stationary

Major historical RL successes

- Learned the world's best player of Backgammon (Tesauro 1995)
- Learned acrobatic helicopter autopilots (Ng, Abbeel, Coates et al 2006+)
- Widely used in the placement and selection of advertisements and pages on the web (e.g., A-B tests)
- Used to make strategic decisions in *Jeopardy!* (IBM's Watson 2011)
- Achieved human-level performance on Atari games from pixel-level visual input, in conjunction with deep learning (Google Deepmind 2015)
- In all these cases, performance was better than could be obtained by any other method, and was obtained without human instruction

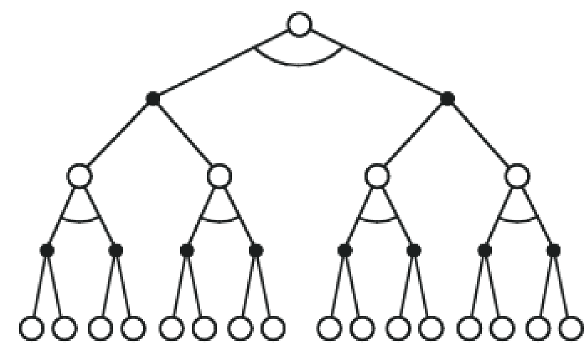
Example: TD-Gammon





estimated state value
(≈ prob of winning)

Action selection
by a shallow search



Start with a random Network

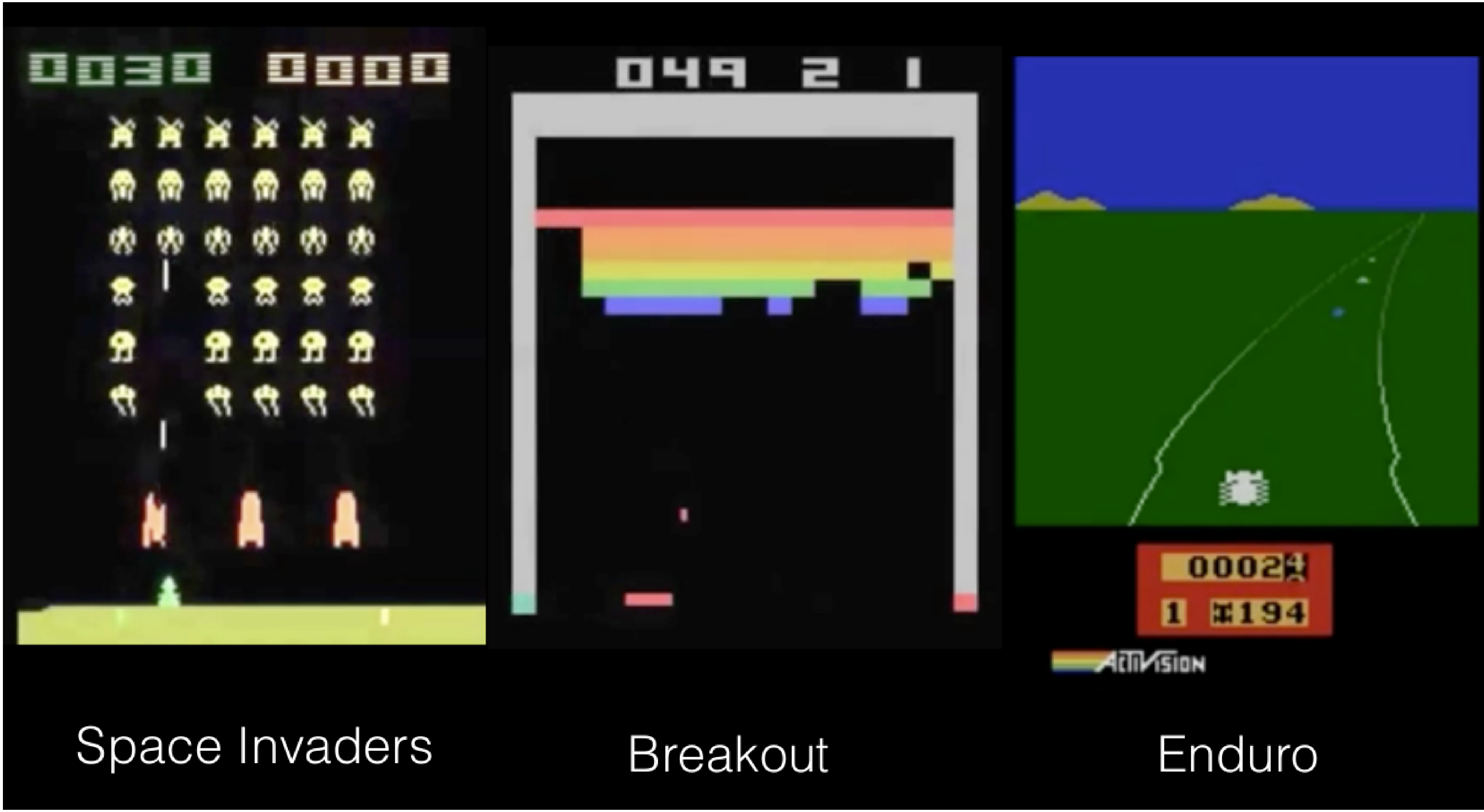
Play millions of games against itself

Learn a value function from this simulated experience

Six weeks later it's the best player of backgammon in the world

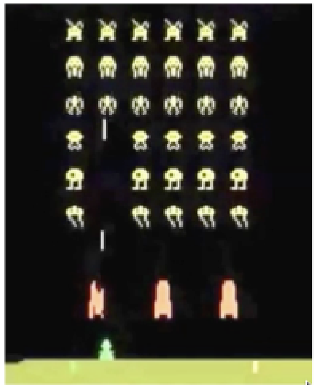
Originally used expert handcrafted features, later repeated with raw board positions

RL + Deep Learing on Atari Games



RL + Deep Learing on Atari Games

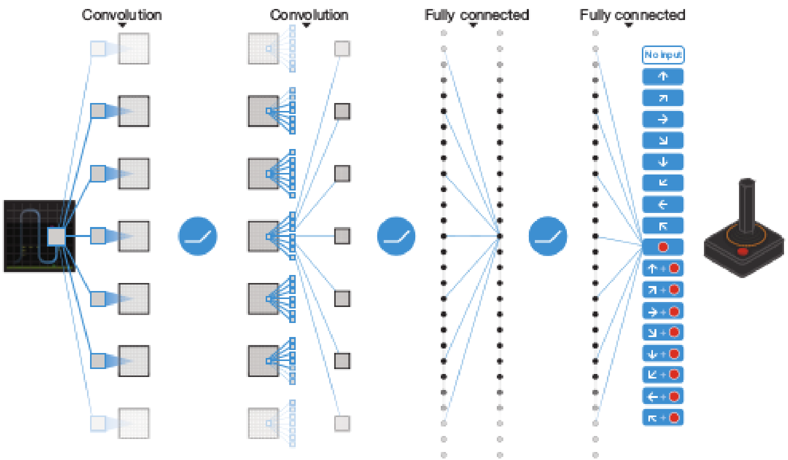
Google Deepmind 2015, Bowling et al. 2012





- Learned to play 49 games for the Atari 2600 game console, without labels or human input, from self-play and the score alone

mapping raw screen pixels



to predictions of final score for each of 18 joystick actions

- Learned to play better than all previous algorithms and at human level for more than half the games

Same learning algorithm applied to all 49 games! w/o human tuning

Major historical RL successes

- Learned the world's best player of Backgammon (Tesauro 1995)
- Learned acrobatic helicopter autopilots (Ng, Abbeel, Coates et al 2006+)
- Widely used in the placement and selection of advertisements and pages on the web (e.g., A-B tests)
- Used to make strategic decisions in *Jeopardy!* (IBM's Watson 2011)
- Achieved human-level performance on Atari games from pixel-level visual input, in conjunction with deep learning (Google Deepmind 2015)
- In all these cases, performance was better than could be obtained by any other method, and was obtained without human instruction

The singularity

The singularity

At some point, humankind will probably create a machine that is pretty smart – smarter than us in many ways.

- this event is generally known as the “singularity” although it means slightly different things to different people.

Advances in AI abilities are coming faster (last 5 yrs)

- IBM’s Watson beats the best human players of *Jeopardy!* (2011)
- Deep neural networks greatly improve the state of the art in speech recognition and computer vision (2012–)
- Google’s self-driving car becomes a plausible reality (≈2013)
- Deepmind’s DQN learns to play Atari games at the human level, from pixels, with no game-specific knowledge (≈2014, *Nature*)
- University of Alberta’s Cepheus solves Poker (2015, *Science*)
- Google Deepmind’s AlphaGo defeats the world Go champion, vastly improving over all previous programs (2016)

The singularity

At some point, humankind will probably create a machine that is pretty smart – smarter than us in many ways.

- this even is generally known as the “singularity” although it means slightly different things to different people.

It’s hard to know what would happen after that event.

One thought: our new inventions might better be modeled as new “species” rather than new “machines”.

Think-Pair-Share Question

When will we understand the principles of intelligence well enough to create, using technology, artificial minds that rival our own in skill and generality? Which of the following best represents *your* current views?

A. Never
B. Not during your lifetime
C. During your lifetime, but not before 2045
D. Before 2045
E. Before 2035
F. It's already happened and we're all living in a simulation of reality

What do you think happens after that?

This course

Content:

– Most of the material comes from Sutton and Barto’s book, Reinforcement Learning: an Introduction, second ed.

– We will also cover selected topics in deep RL not covered in that book.

Objectives:

– understand theoretical underpinnings of RL

– gain practical knowledge of how to solve problems using RL

This course

Workload:

– written/programming assignments approximately weekly (60% of grade)

– end of semester project (40% of grade)

Prerequisites:

– you need to be able to write Python code and install tensorflow.

– need to be "mathematically mature", i.e. be able to understand concepts explained mathematically.

  – background in probability and linear algebra

What do you want to learn?