Reinforcement Learning

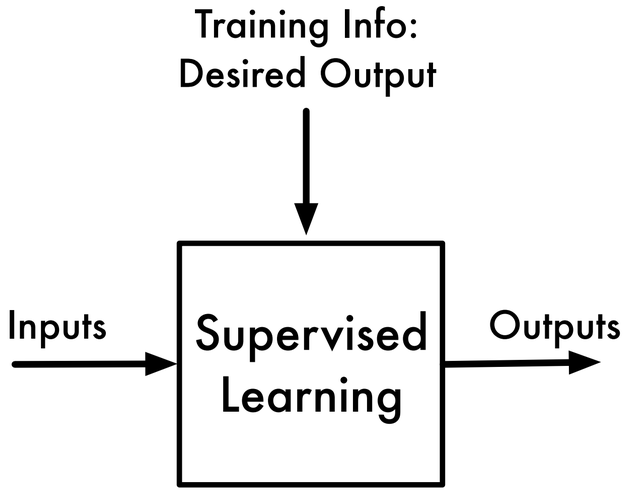
February 3rd, 2017

Rajiv shah

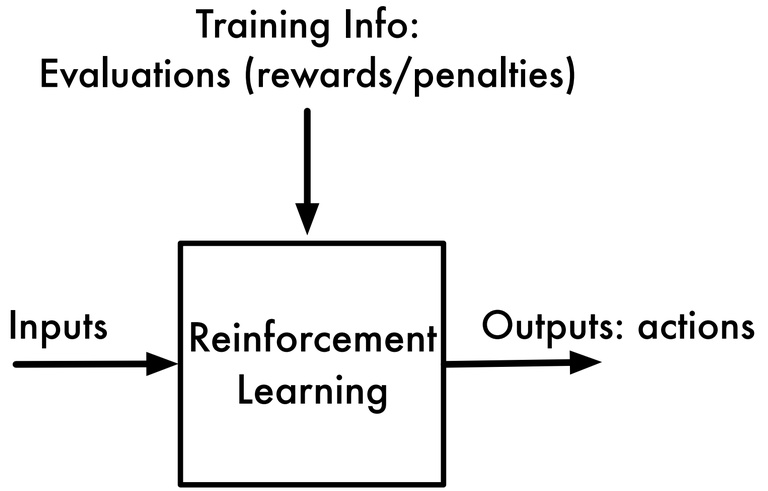
RajivShah.com
rshah@pobox.com
 github.com/rajshah4/

 rajcs4

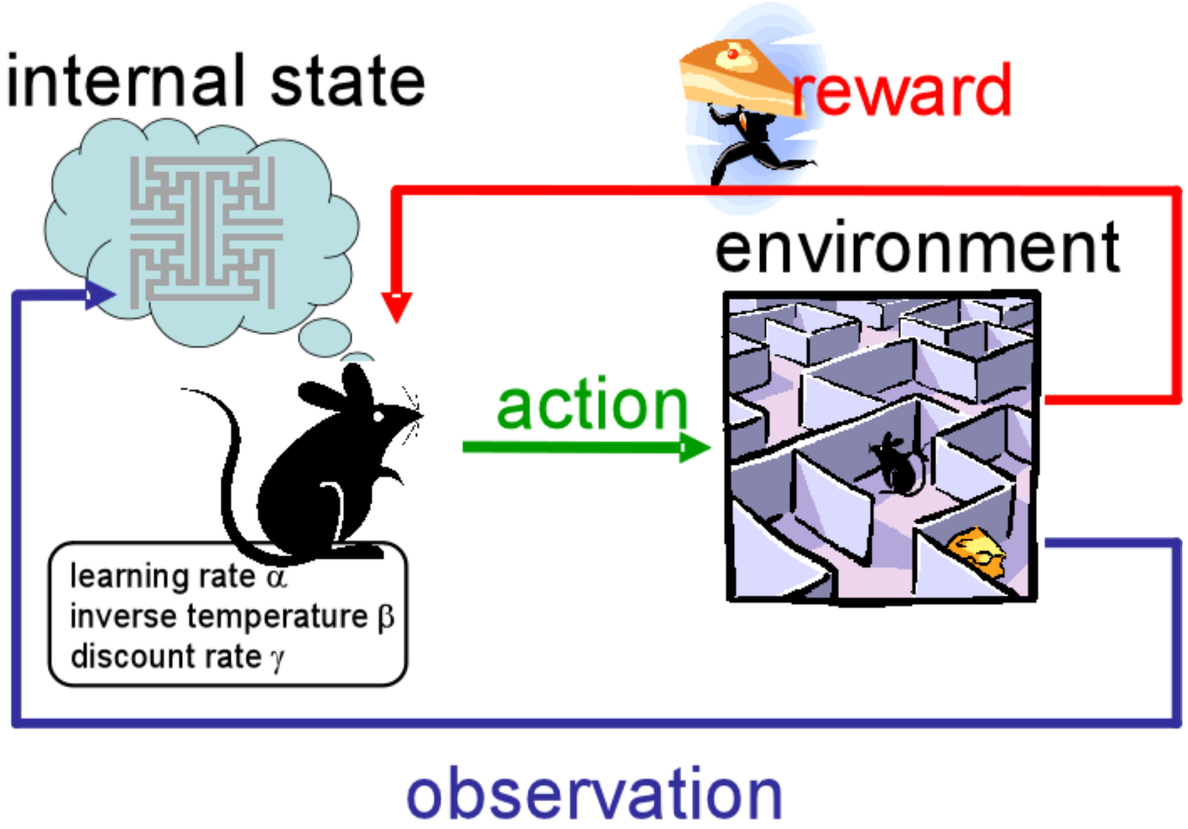
supervised learning



reinforcement learning



agent
environment
reward

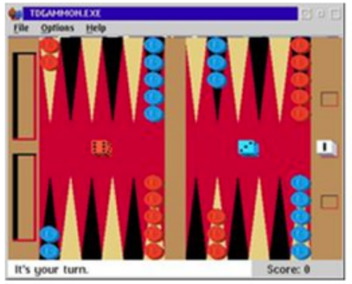


applications

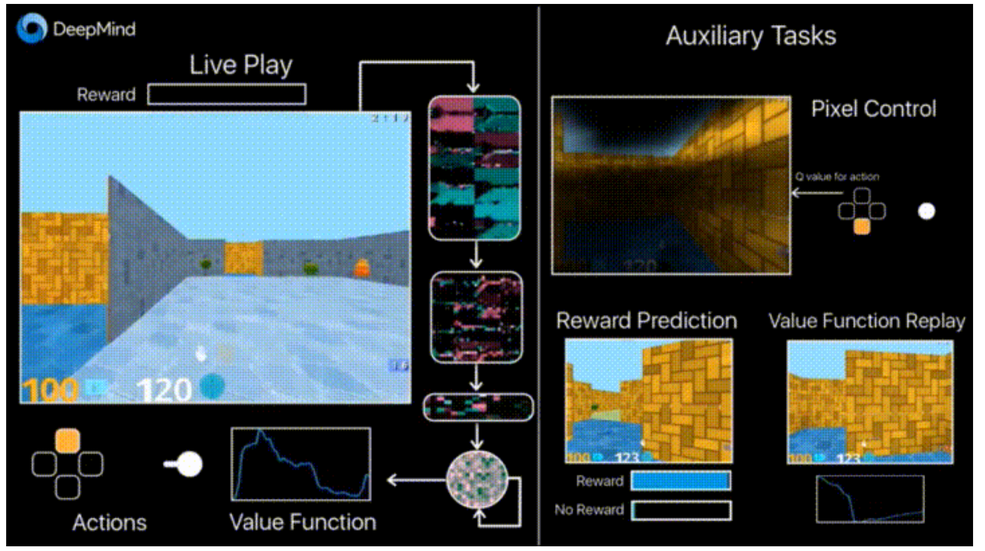

Success Failure
x2 speed
The robot collects pictures of
successful/failed bin picking trials

Example: TD-Gammon

- Tesauro (1995)
- RL to play Backgammon to become the world championship
- Immediate reward
  - +100 if win
  - -100 if lose
  - 0 for all other states



- Trained by playing 1.5 million games against itself
- Now approximately equal to best human player



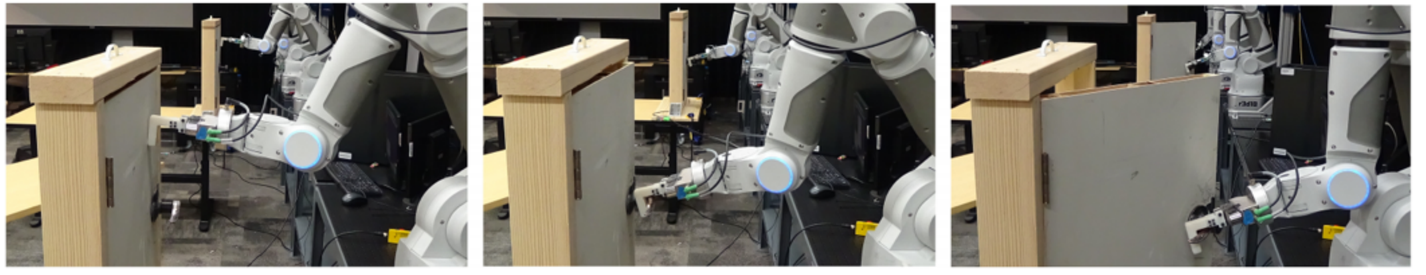


Fig. 6: Two robots learning to open doors using asynchronous NAF. The final policy learned with two workers could achieve a 100% success rate on the task across 20 consecutive trials.

apply RL

tic tac toe

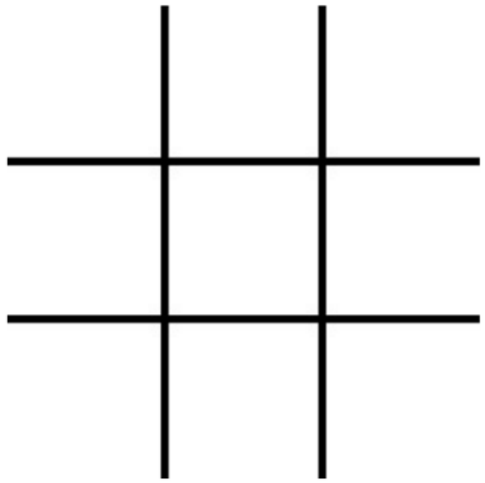


figure 1: Plain grid

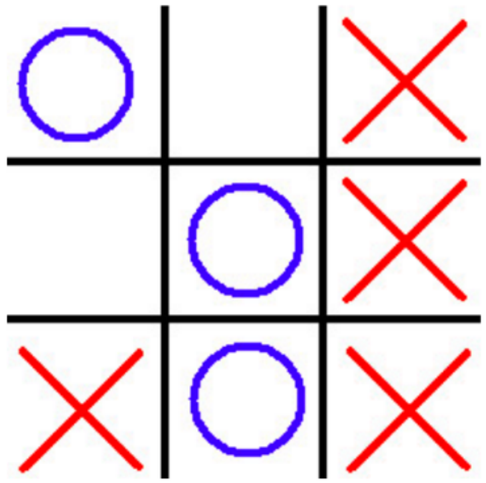


figure 2: X’s won

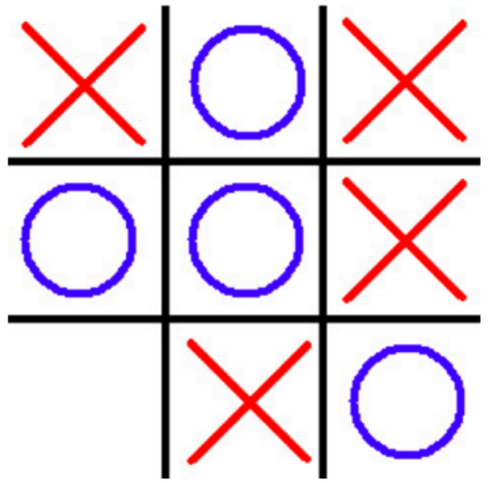
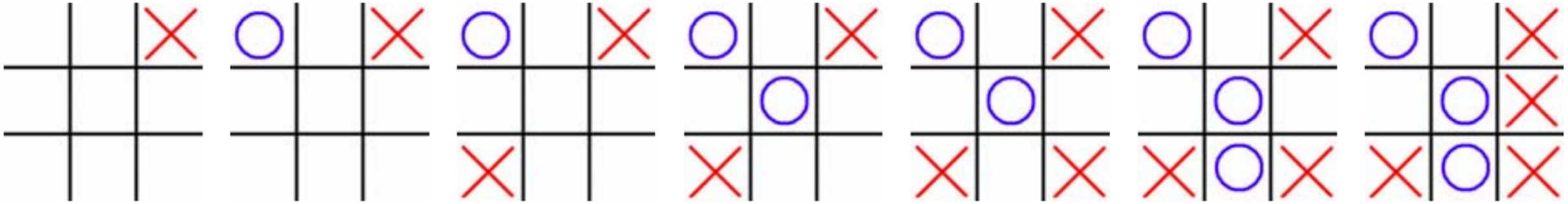


figure 3: Draw

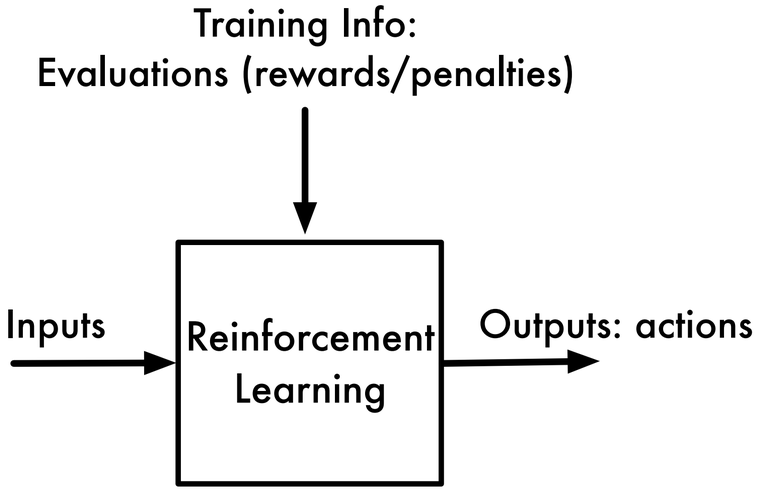
states/ probabilities



P(1a)=0.5625 P(1b)=0.125 P(2a)=0.625 P(2b)=0.25 P(3a)=0.75 P(3b)=0 P(4a)=1

figure 4: Updating afterstates

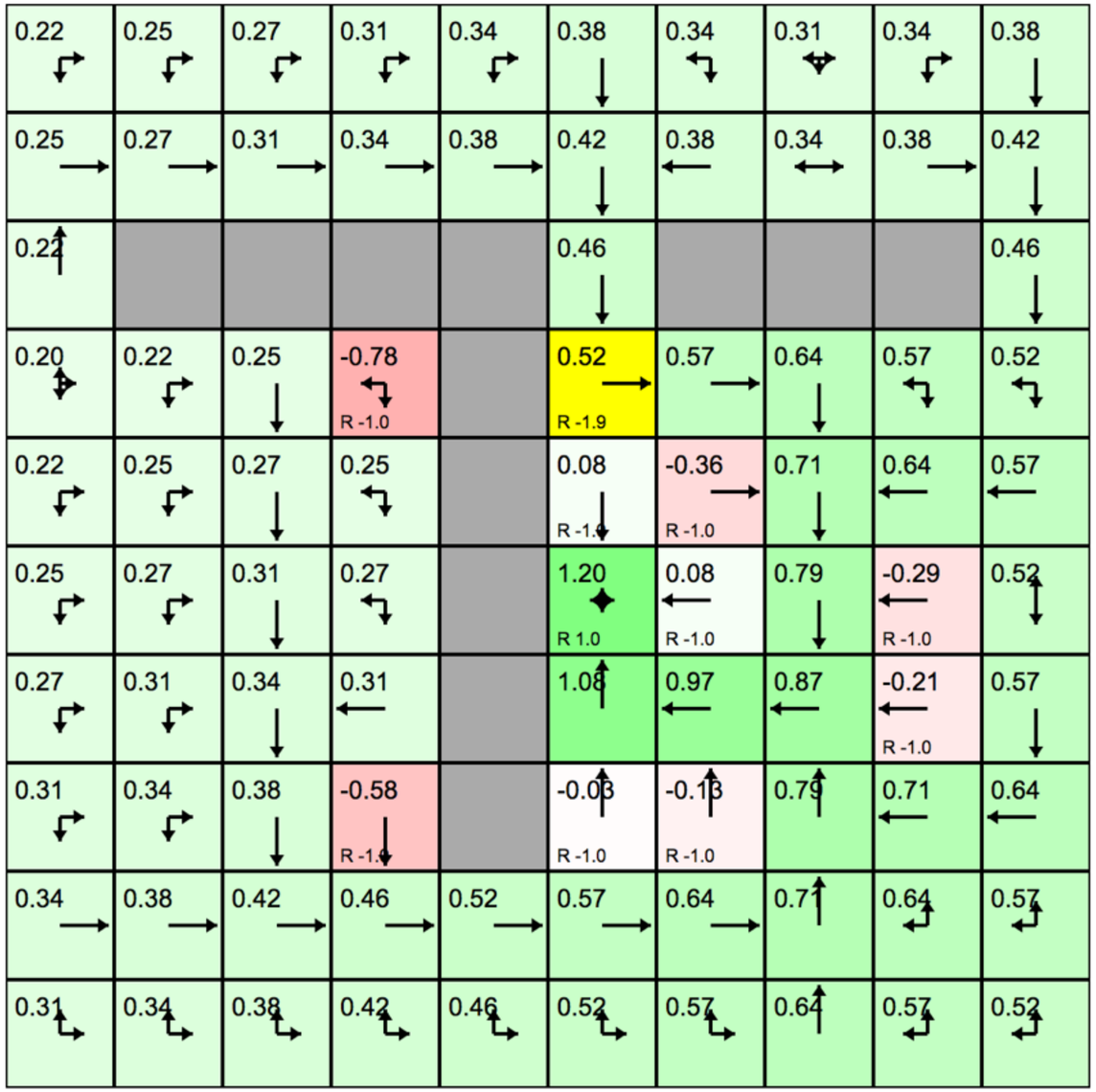
lookup table



5890 possible states

GridWorld: Dynamic Programming Demo

Policy Evaluation (one sweep) | Policy Update | Toggle Value Iteration | Reset



rewards

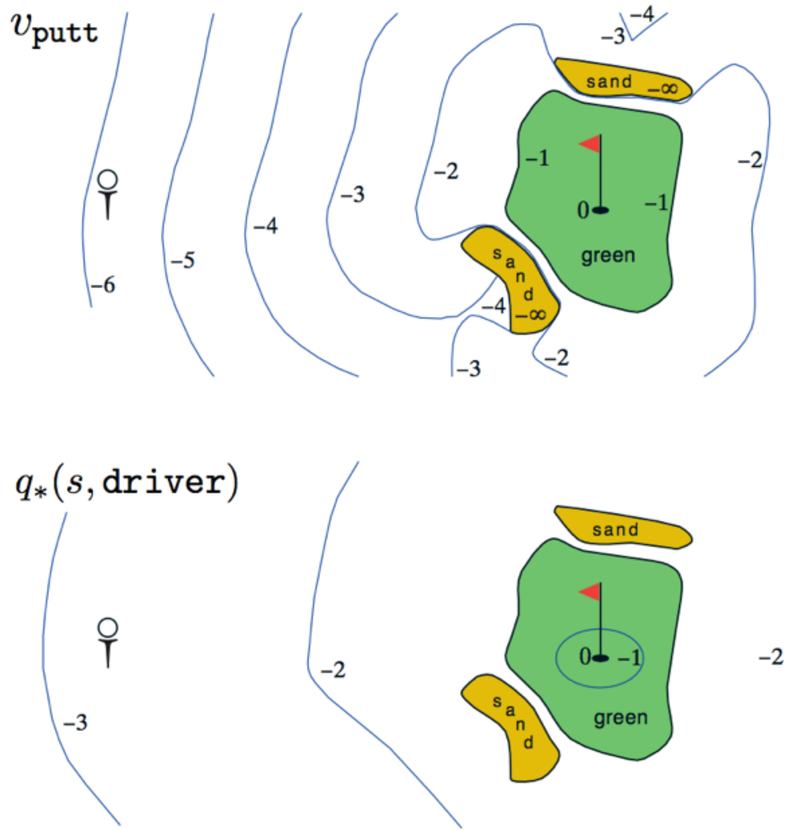


Figure 3.6: A golf example: the state-value function for putting (above) and the optimal action-value function for using the driver (below).

bellman

Q-value Bellman Equation

The basic idea:

Follow policy

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \cdots$$

$$= r_{t+1} + \gamma \left(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} \cdots \right)$$

$$= r_{t+1} + \gamma R_{t+1}$$

Action a

So:

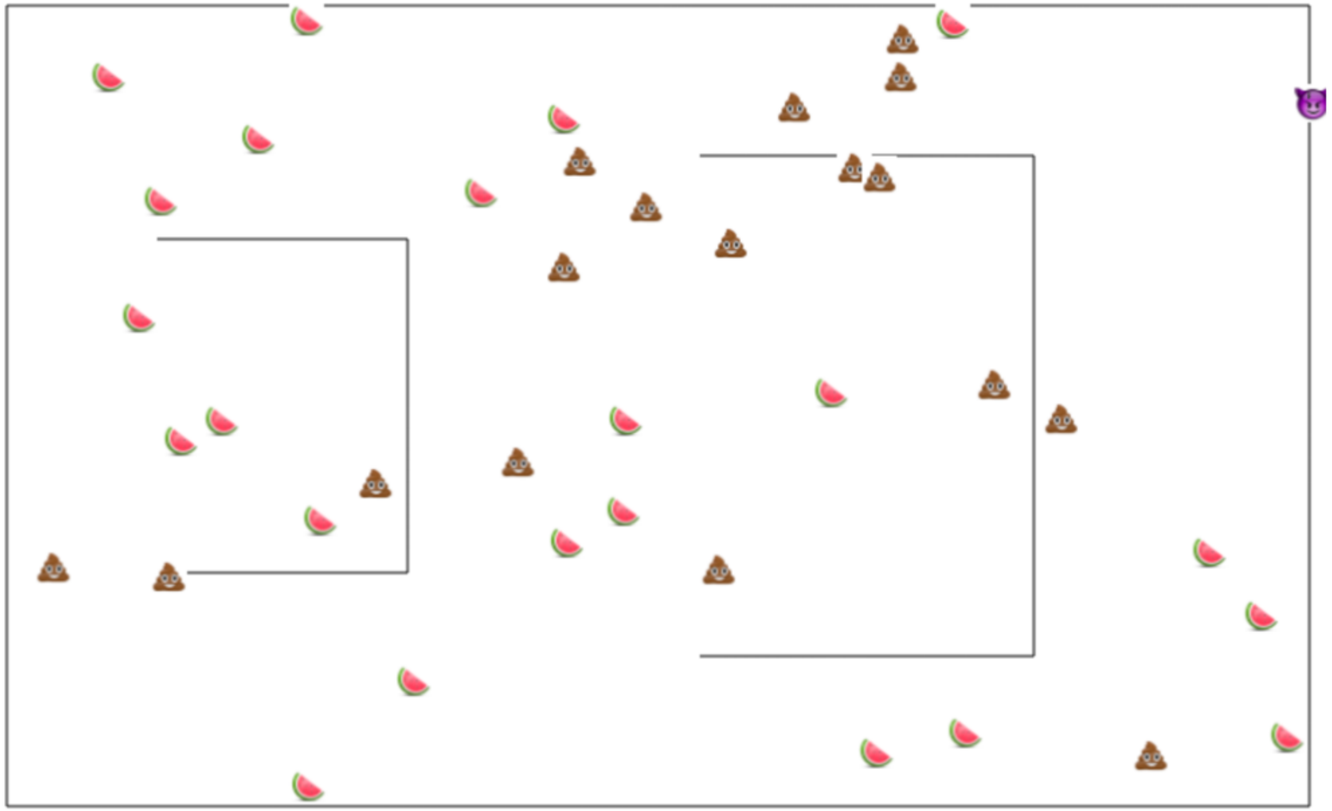
$$Q^{\pi}(s,a) = E_{\pi}\left\{ R_t \middle| s_t = s, a_t = a \right\}$$

$$= E_{\pi}\left\{ r_{t+1} + \gamma V^{\pi}(s_{t+1}) \middle| s_t = s, a_t = a \right\}$$

Q learning

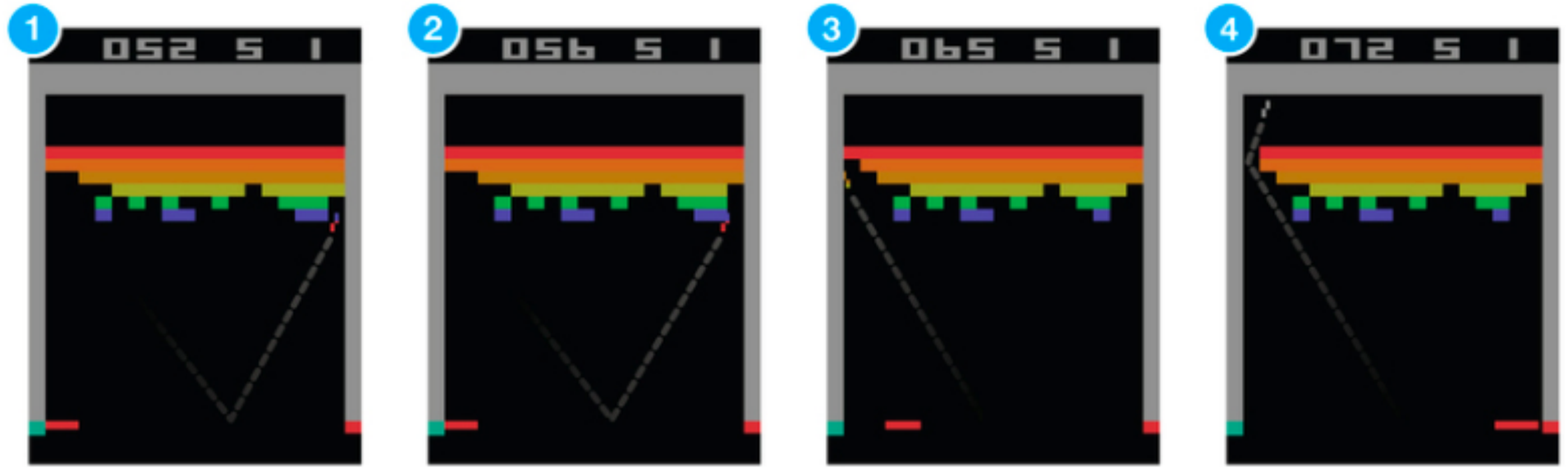
initialize $Q[num\_states, num\_actions]$ arbitrarily
observe initial state s
repeat
    select and carry out an action a
    observe reward r and new state s'
    $Q[s,a] = Q[s,a] + \alpha(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$
    $s = s'$
until terminated

Eat-Melon Demo



experience replay size: 5520
exploration epsilon: 0.9872030456852792
age: 5522
average Q-learning loss: 0.10564404568070364
smooth-ish reward: 0.5320415558446885

Exploration versus Exploitation



greyscale
last 4 images
84 x 84
$256^{84x84x4} \approx 10^{67970}$ possible game states

Deep Q learning

Layer	Input	Filter size	Stride	Num filters	Activation	Output
conv1	84x84x4	8x8	4	32	ReLU	20x20x32
conv2	20x20x32	4x4	2	64	ReLU	9x9x64
conv3	9x9x64	3x3	1	64	ReLU	7x7x64
fc4	7x7x64			512	ReLU	512
fc5	512			18	Linear	18

UNIVERSE

- Flash Games
- World of Bits
- Internet
- PC Games

GYM

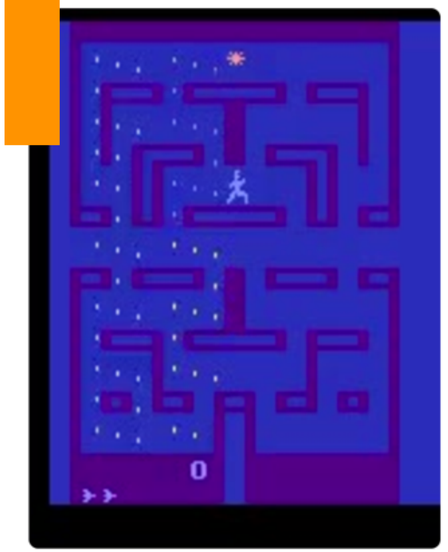
- Classic control
- Algorithmic
- Atari
- Board games
- Box2D
- MuJoCo
- Parameter tuning
- Toy text
- Safety
- Minecraft
- PyGame Learning Environment
- Soccer
- Doom

Atari

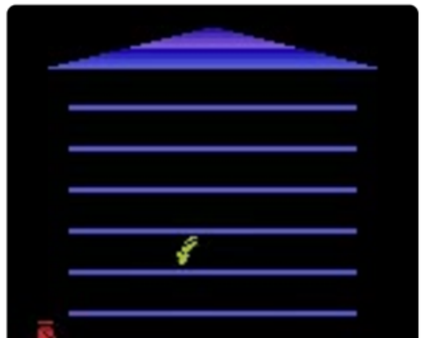
Reach high scores in Atari 2600 games.

AirRaid-ram-v0

AirRaid-v0



Alien-v0

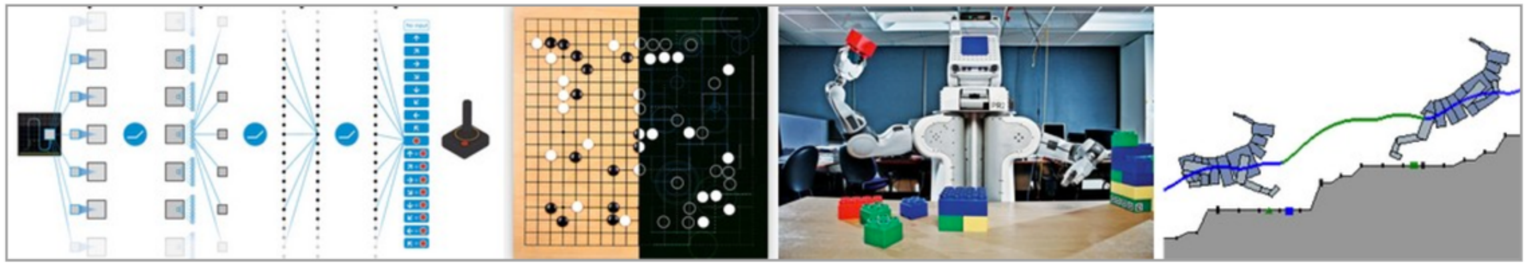


lines

Deep Reinforcement Learning: Pong from Pixels

May 31, 2016

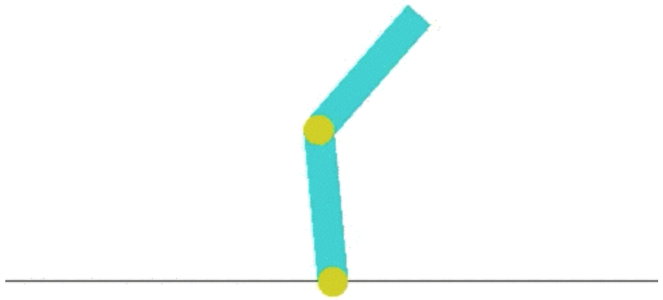
This is a long overdue blog post on Reinforcement Learning (RL). RL is hot! You may have noticed that computers can now automatically learn to play ATARI games (from raw game pixels!), they are beating world champions at Go, simulated quadrupeds are learning to run and leap, and robots are learning how to perform complex manipulation tasks that defy explicit programming. It turns out that all of these advances fall under the umbrella of RL research. I also became interested in RL myself over the last ~year: I worked through Richard Sutton's book, read through David Silver's course, watched John Schulmann's lectures, wrote an RL library in Javascript, over the summer interned at DeepMind working in the DeepRL group, and most recently pitched in a little with the design/development of OpenAI Gym, a new RL benchmarking toolkit. So I've certainly been on this funwagon for at least a year but until now I haven't gotten around to writing up a short post on why RL is a big deal, what it's about, how it all developed and where it might be going.



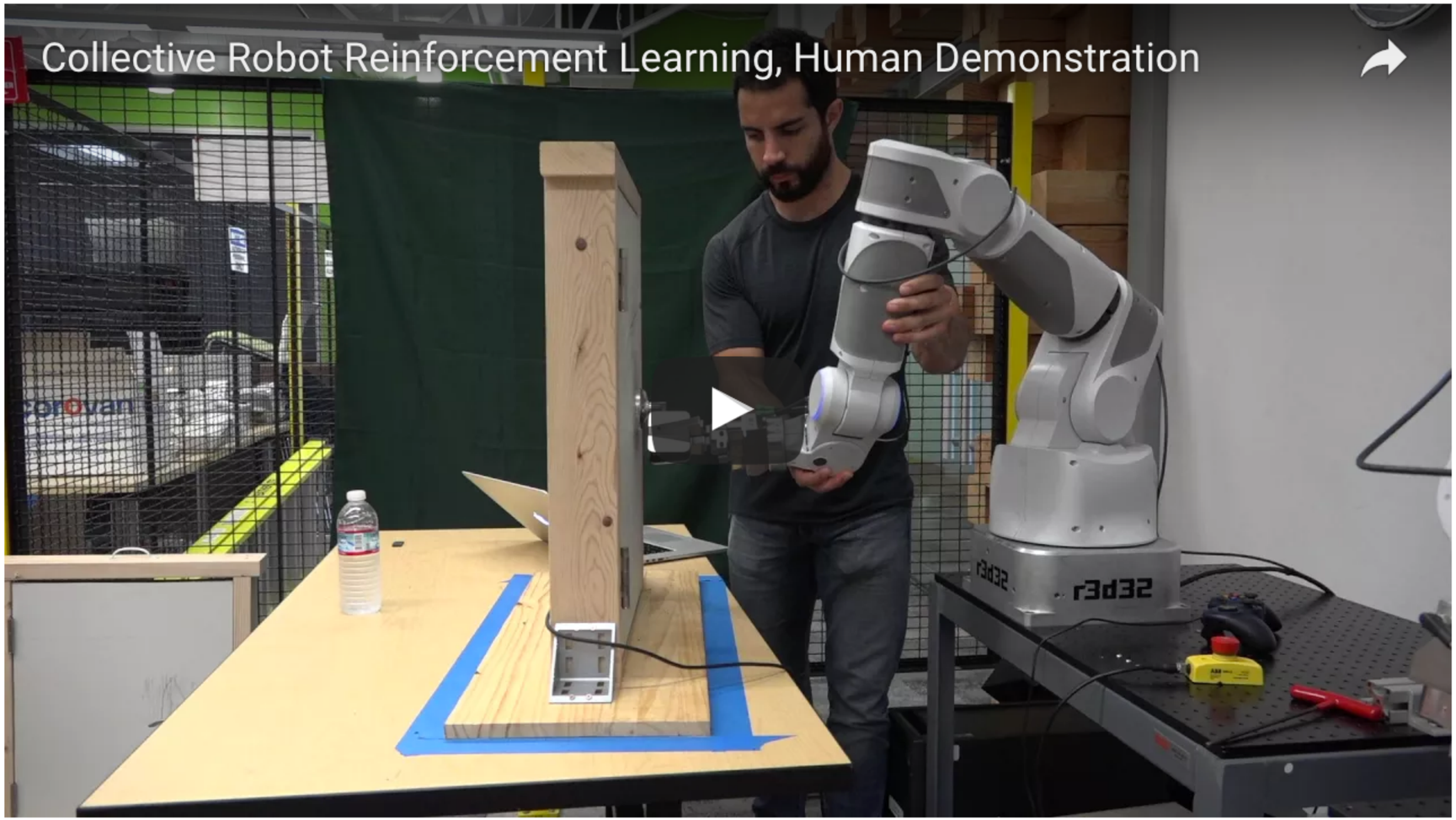
Examples of RL in the wild. **From left to right**: Deep Q Learning network playing ATARI, AlphaGo, Berkeley robot stacking Legos, physically-simulated quadruped leaping over terrain.

130 lines of python

Acrobat



imitation learning



applications of RL
understanding —> deep Q
how we further the use of RL

Reinforcement Learning

February 3rd, 2017

Rajiv shah

RajivShah.com
rshahATpobox.com
 github.com/rajshah4/

 rajcs4