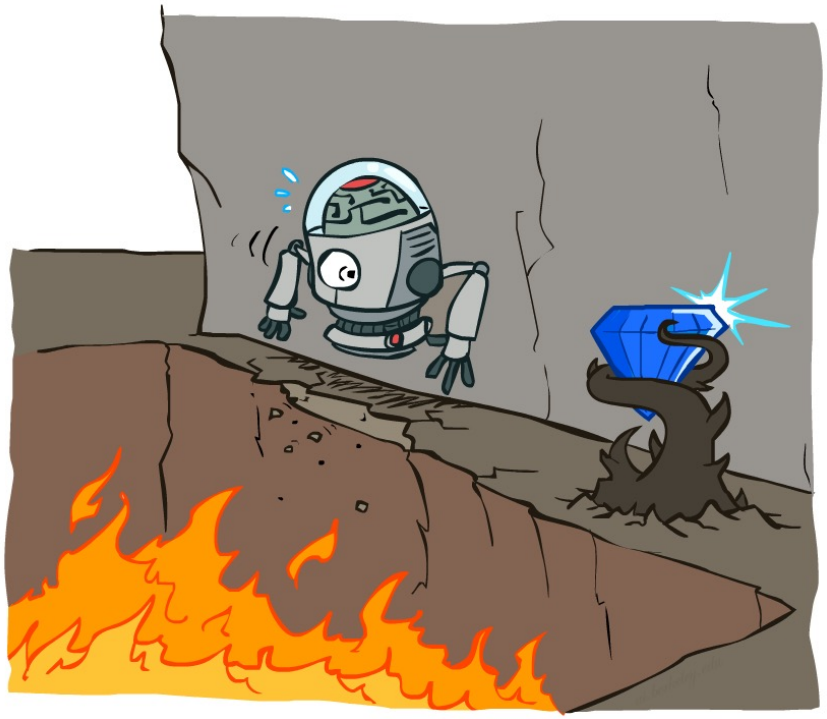
RIT

RL Foundations



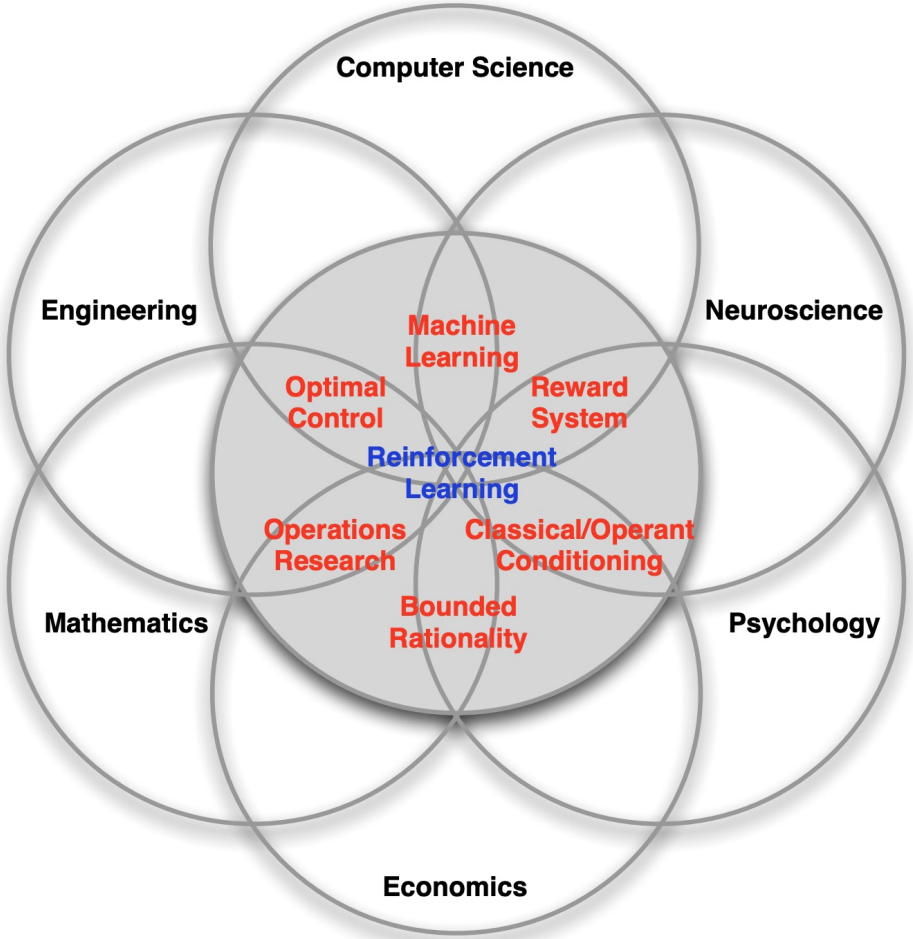
Dr. Jamison Heard

January 17, 2023

Slides adapted from UC Berkley's Intro to AI course

RIT

The Many Faces of RL



RIT

Its All About the Decisions

- What are some examples of AI decisions?
- **Decisions maximize expected utility (reward)**

RIT

RL Characteristics

What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a *reward* signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

Rewards

- A reward R_t is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximise cumulative reward

Reinforcement learning is based on the reward hypothesis

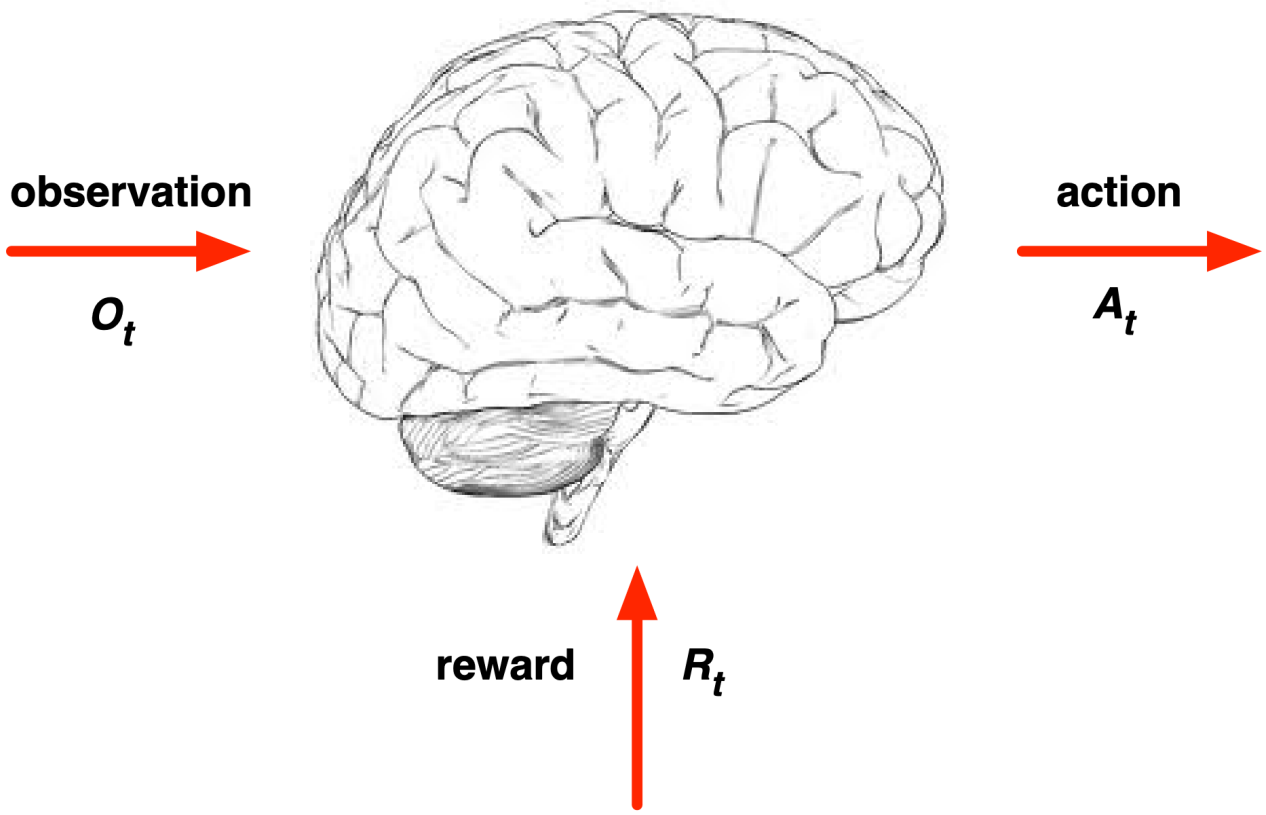
Definition (Reward Hypothesis)

All goals can be described by the maximisation of expected cumulative reward

Do you agree with this statement?

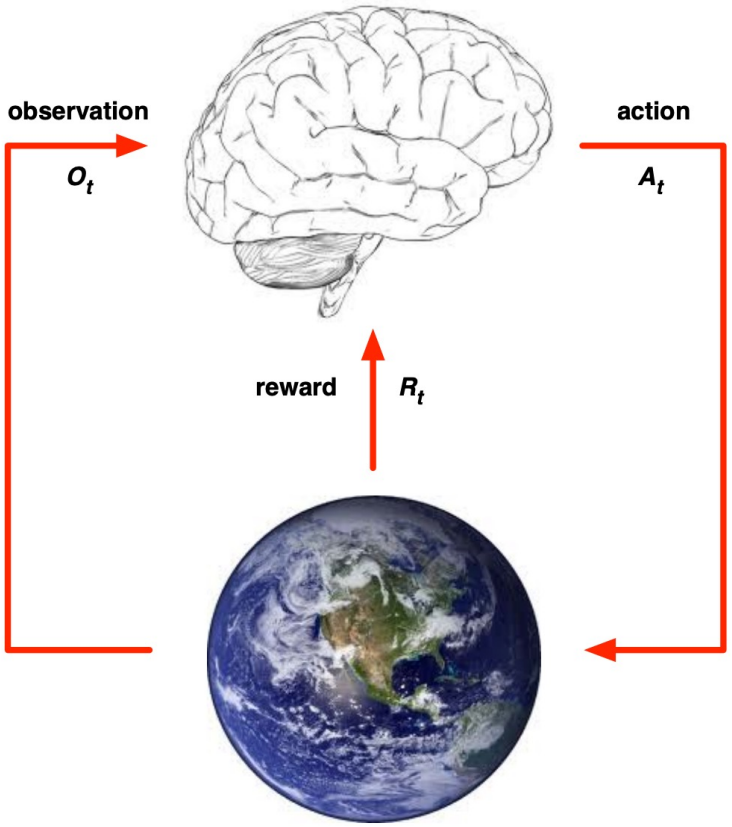
RIT

Sequential Decisions



RIT

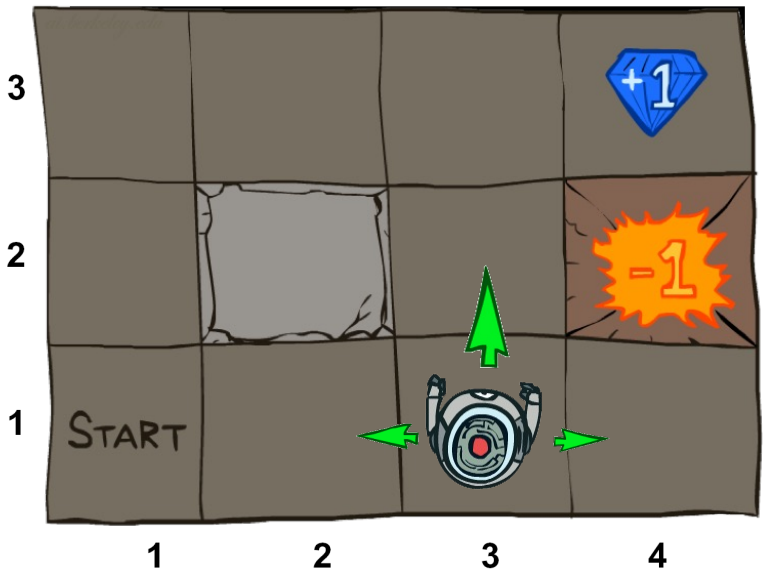
RL Environments



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

RIT Example: Grid World

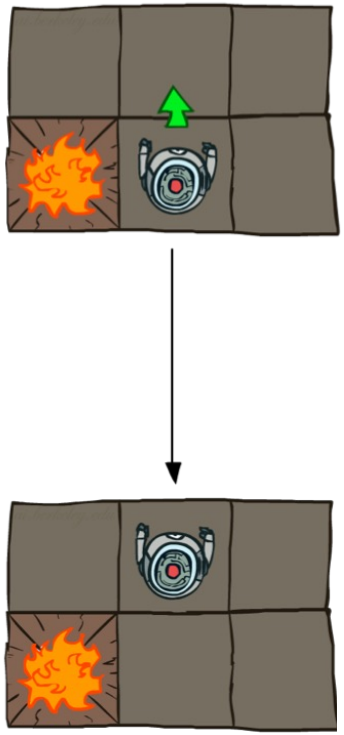
- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - 80% of the time, the action North takes the agent North (if there is no wall there)
  - 10% of the time, North takes the agent West; 10% East
  - If there is a wall in the direction the agent would have been taken, the agent stays put
- The agent receives rewards each time step
  - Small "living" reward each step (can be negative)
  - Big rewards come at the end (good or bad)
- Goal: maximize sum of rewards



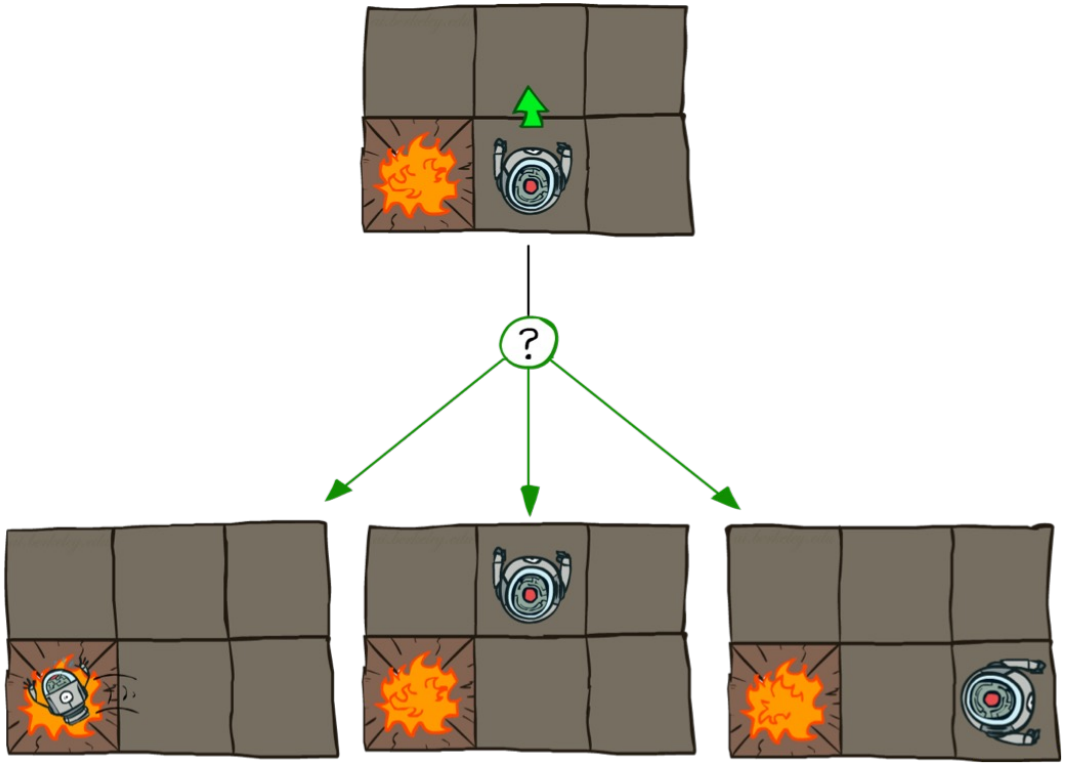
RIT

Grid World Actions

Deterministic Grid World

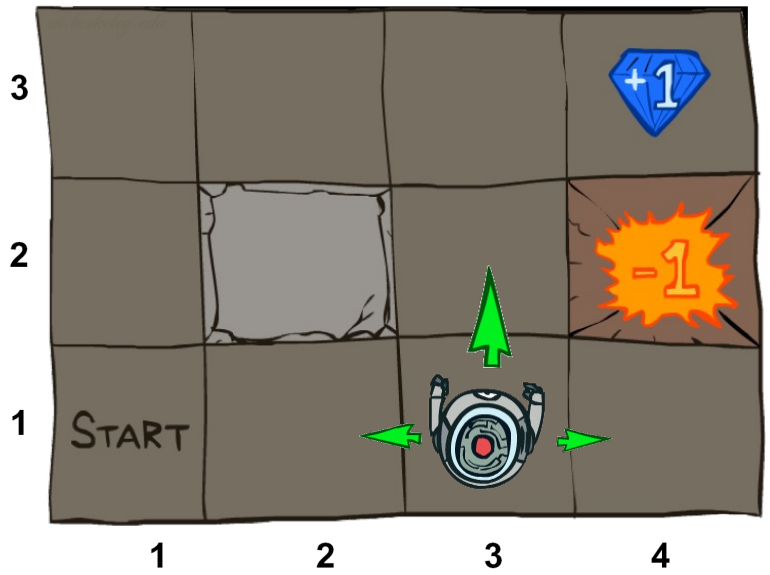


Stochastic Grid World



RIT Markov Decision Processes

- An MDP is defined by:
  - A set of states s ∈ S
  - A set of actions a ∈ A
  - A transition function T(s, a, s')
    - Probability that a from s leads to s', i.e., P(s'| s, a)
    - Also called the model or the dynamics
  - A reward function R(s, a, s')
    - Sometimes just R(s) or R(s')
  - A start state
  - Maybe a terminal state



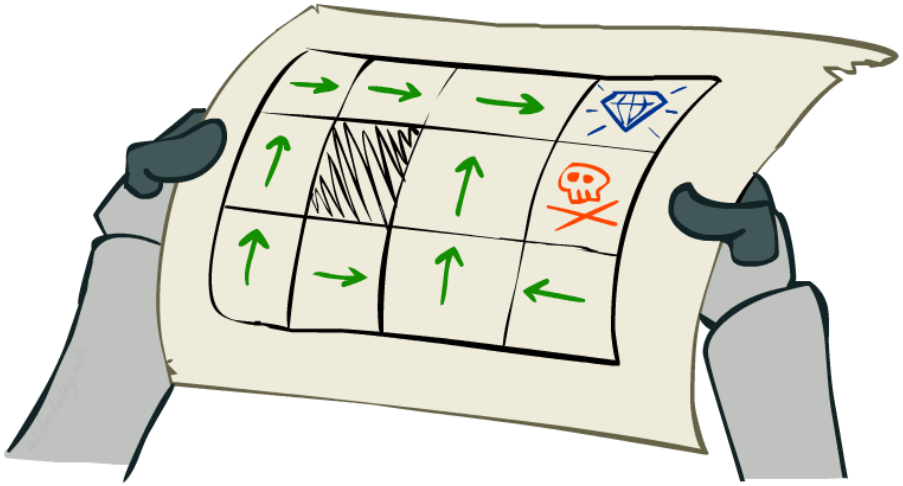
- MDPs are non-deterministic search problems

RIT

Policies

- In deterministic single-agent search problems, we wanted an optimal plan, or sequence of actions, from start to a goal

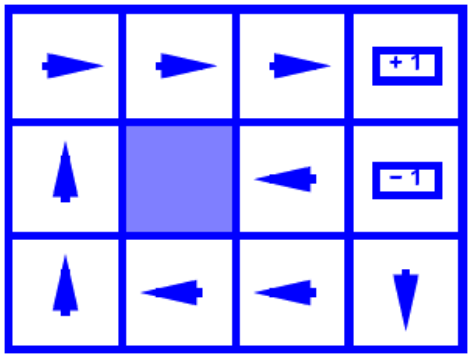
- For MDPs, we want an optimal policy $\pi^*: S \rightarrow A$
  - A policy π gives an action for each state
  - An optimal policy is one that maximizes expected utility if followed
  - An explicit policy defines a reflex agent



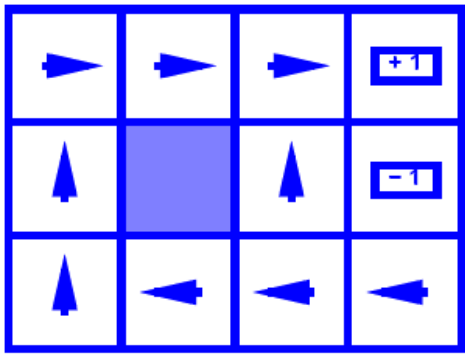
Optimal policy when R(s, a, s') = -0.03
for all non-terminals s

RIT

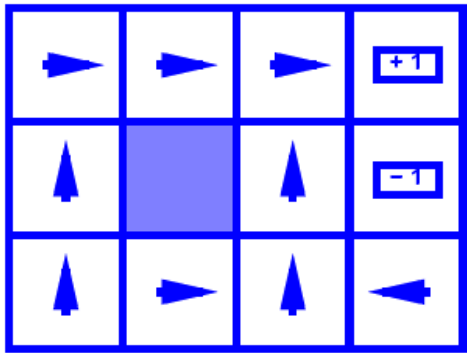
Optimal Policies



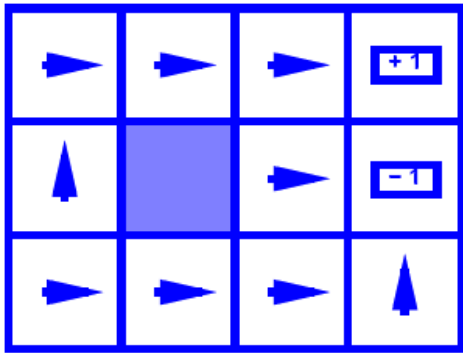
R(s) = -0.01



R(s) = -0.03



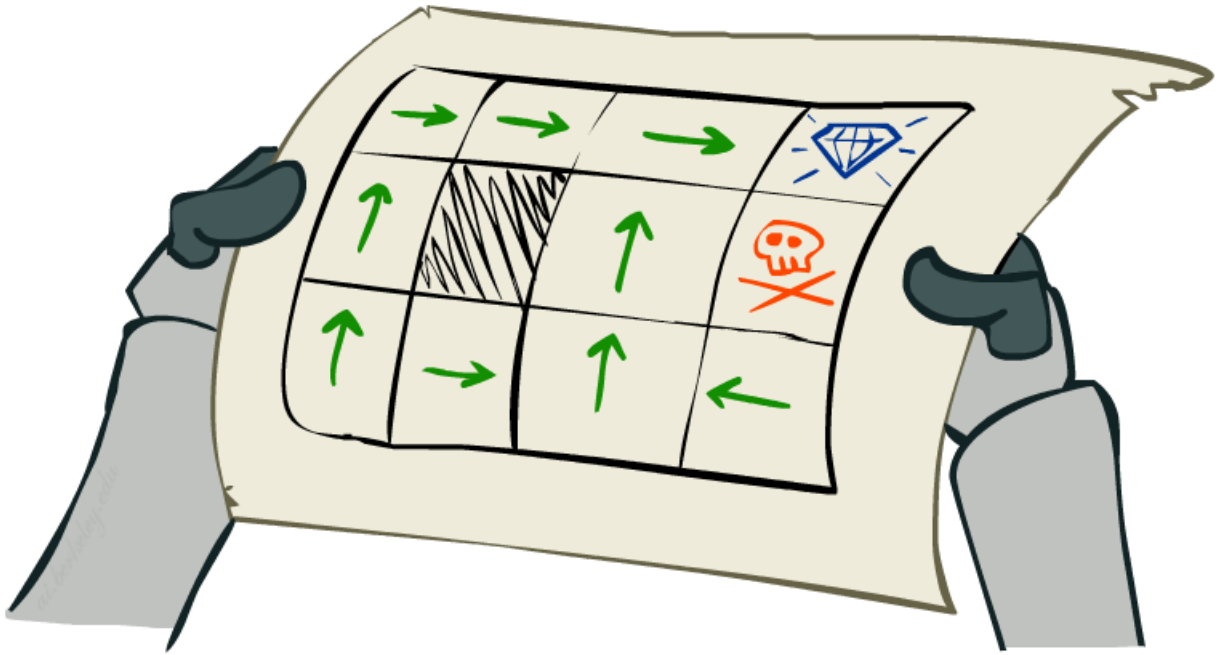
R(s) = -0.4



R(s) = -2.0

RIT

Solving MDPs



RIT

Optimal Quantities

- The value (utility) of a state s:

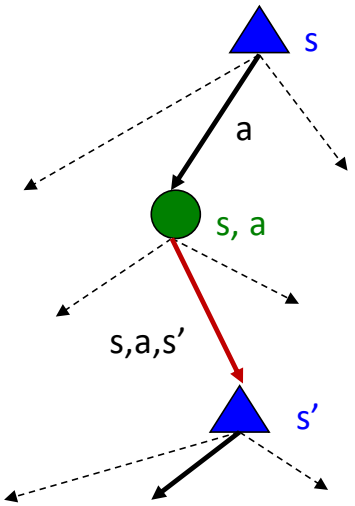
  $V^*(s)$ = expected utility starting in s and acting optimally

- The value (utility) of a q-state (s,a):

  $Q^*(s,a)$ = expected utility starting out having taken action a from state s and (thereafter) acting optimally

- The optimal policy:

  $\pi^*(s)$ = optimal action from state s



s is a *state*

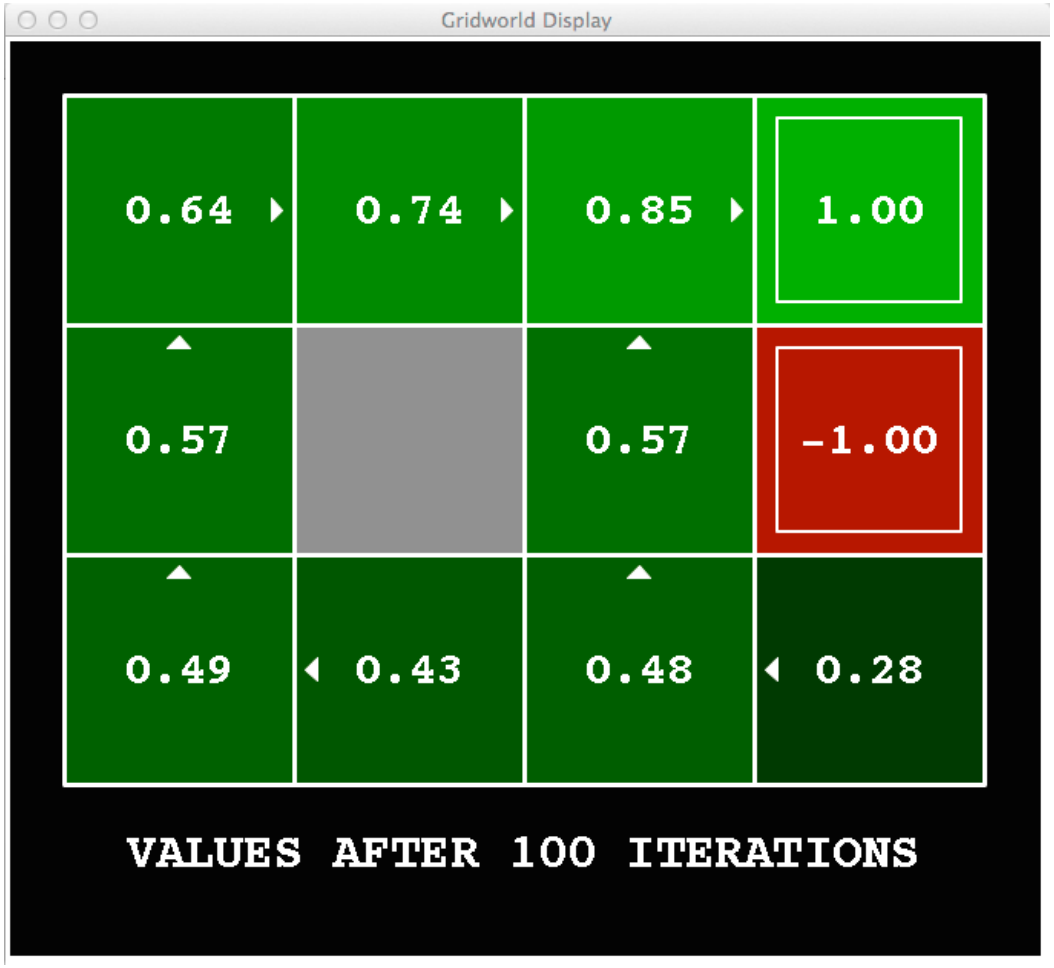
(s, a) is a *q-state*

(s,a,s') is a *transition*

[Demo – gridworld values (L8D4)]

RIT

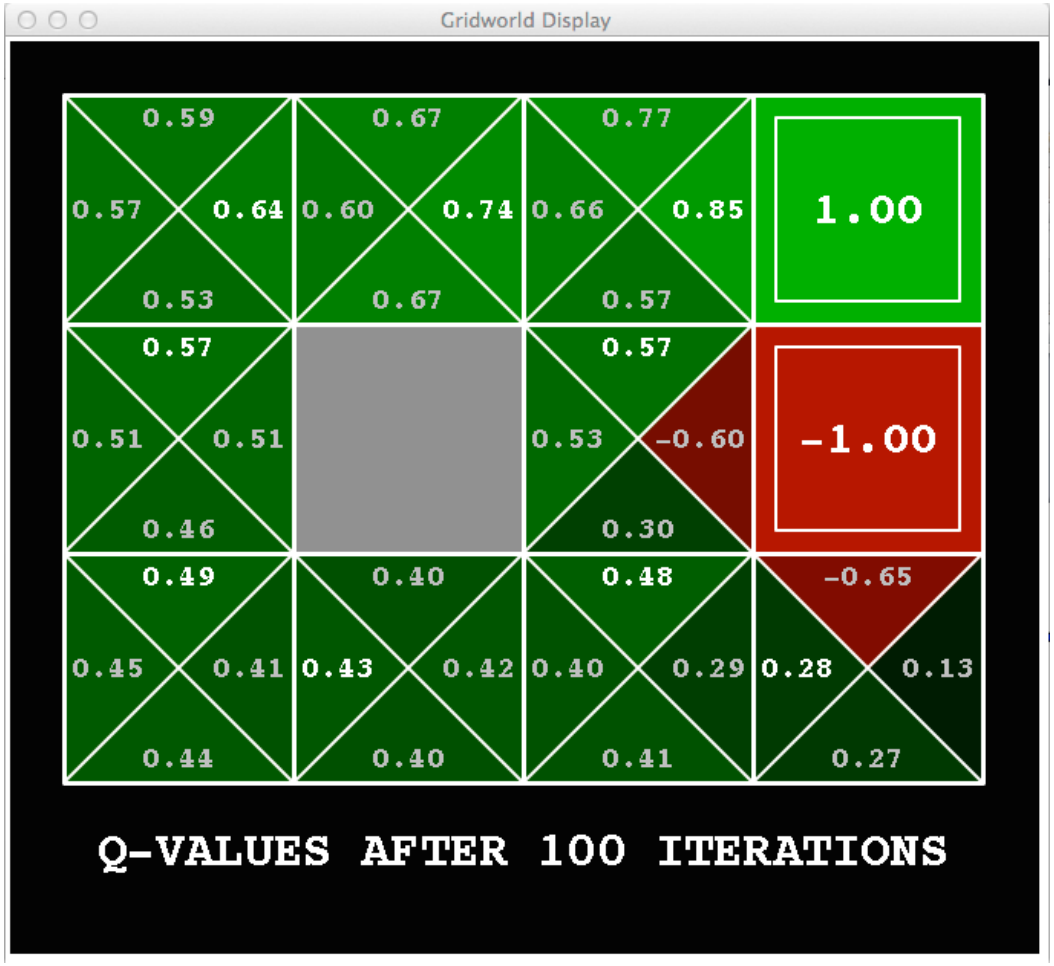
Gridworld V Values



Noise = 0.2
Discount = 0.9
Living reward = 0

RIT

Gridworld Q Values

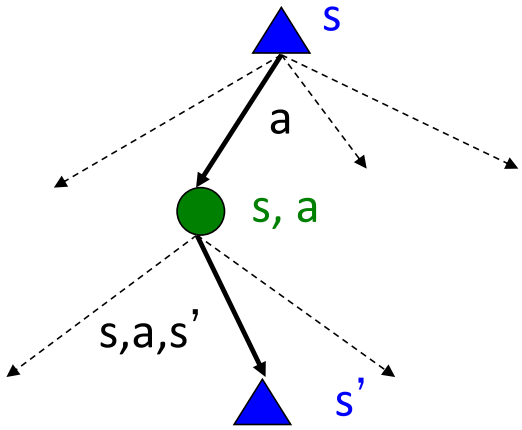


Noise = 0.2
Discount = 0.9
Living reward = 0

RIT

Values of States

- Fundamental operation: compute the (expectimax) value of a state
  - Expected utility under optimal action
  - Average sum of (discounted) rewards
  - This is just what expectimax computed!



- Recursive definition of value:

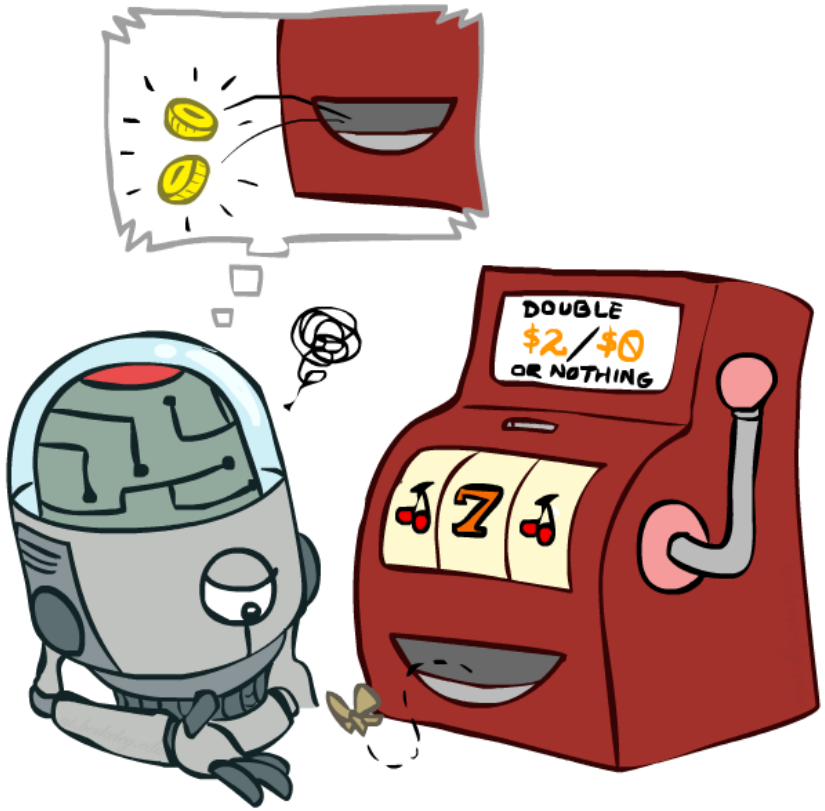
$$V^*(s) = \max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s') \right]$$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s') \right]$$

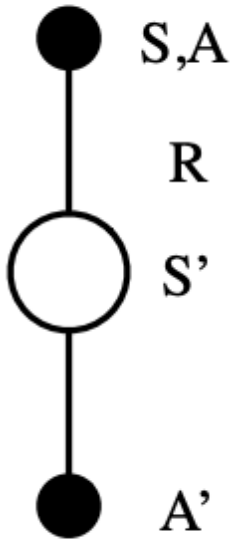
RIT

Model-Free Learning



RIT

Q-Learning: SARSA



$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right)$$

RIT

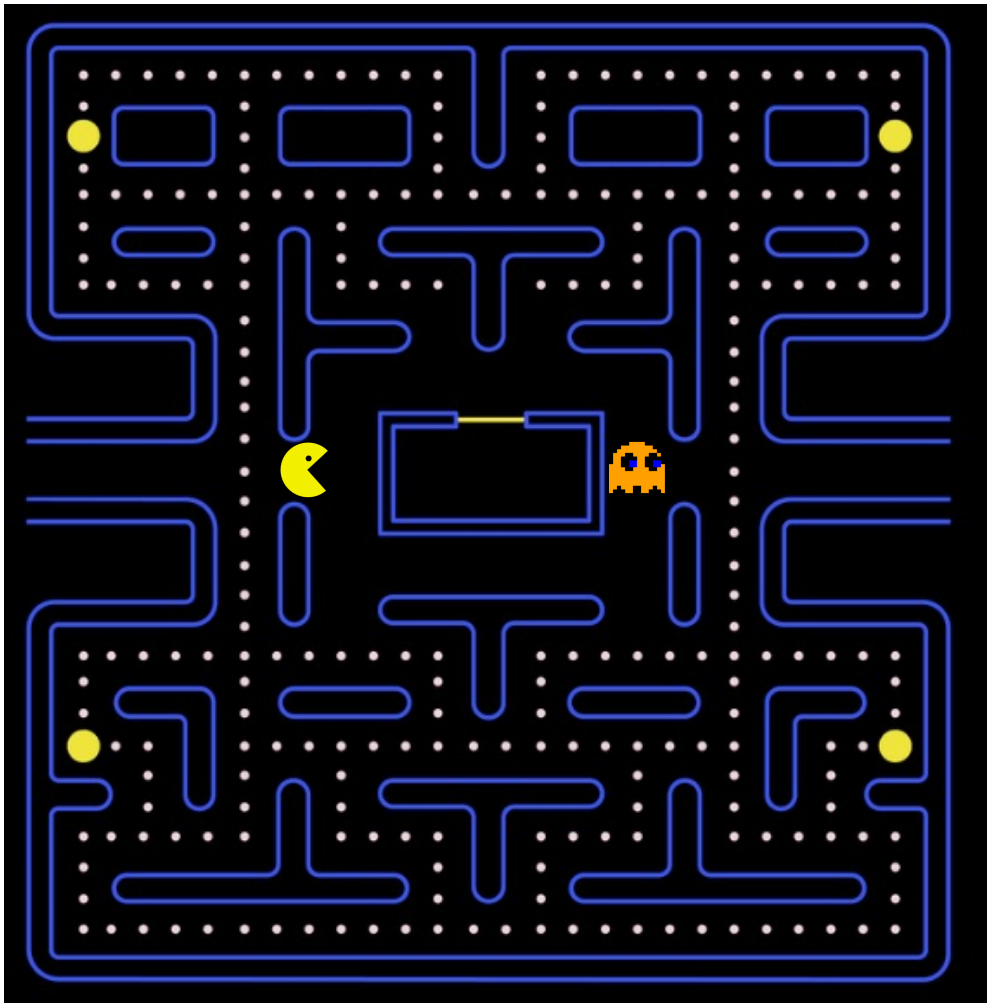
Q-Learning: SARSA

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Repeat (for each step of episode):
        Take action A, observe R, S'
        Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma Q(S', A') - Q(S, A)\big]$
        $S \leftarrow S'$; $A \leftarrow A'$;
    until S is terminal

RIT

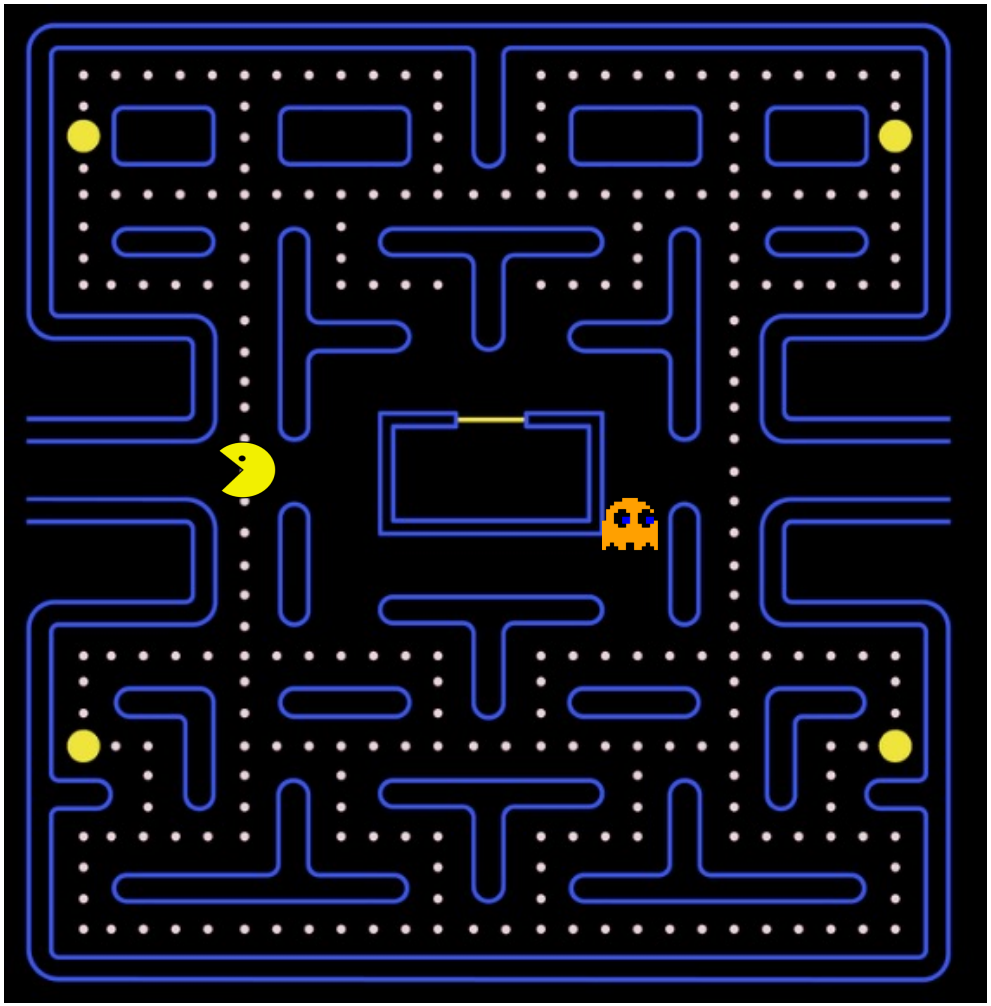
Pacman Example



State?
Actions?
Reward?
Gamma?
Alpha?
Q-values Initial?

1. Take action according to policy

RIT Pacman Example



State?
Actions?
Reward?
Gamma?
Alpha?
Q-values Initial?

1. Take action according to policy
2. Observe R, S'
3. Update Q(s,a)

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right)$$

RIT Always Exploit: Greedy Action Selection

"Behind one door is tenure - behind the other is flipping burgers at McDonald's."

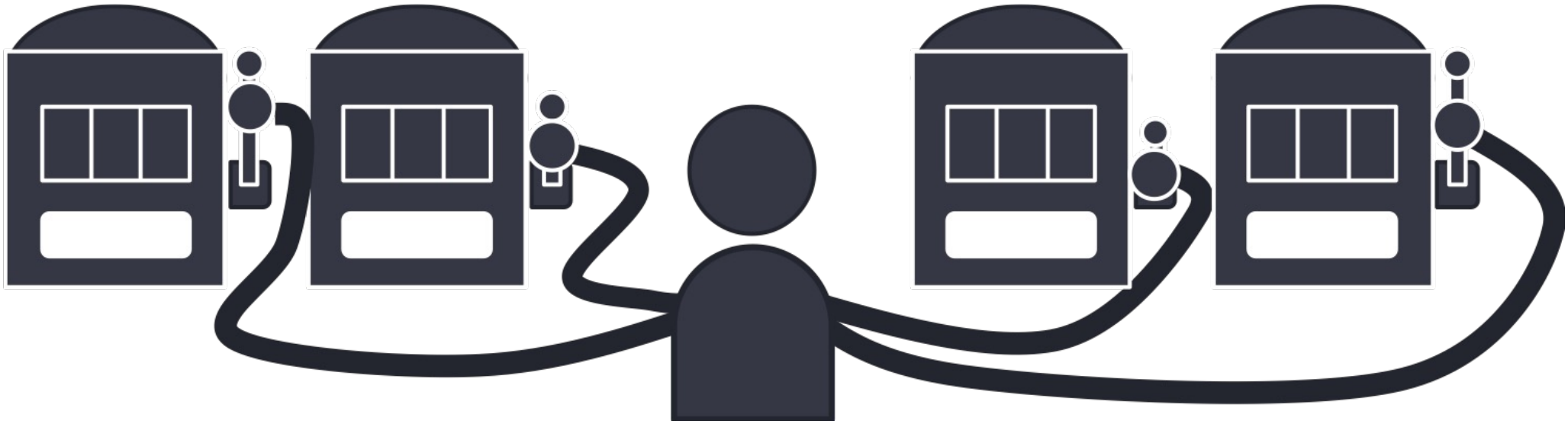
- There are two doors in front of you.
- You open the left door and get reward 0
  $V(left) = 0$
- You open the right door and get reward $+1$
  $V(right) = +1$
- You open the right door and get reward $+3$
  $V(right) = +2$
- You open the right door and get reward $+2$
  $V(right) = +2$

$\vdots$

- Are you sure you've chosen the best door?

RIT

Multi-Armed Bandit



RIT Exploration/Exploitation: Epsilon-Greedy

Algorithm 2: Epsilon-Greedy Action Selection

Data: Q: Q-table generated so far, : a small number, S: current state

Result: Selected action

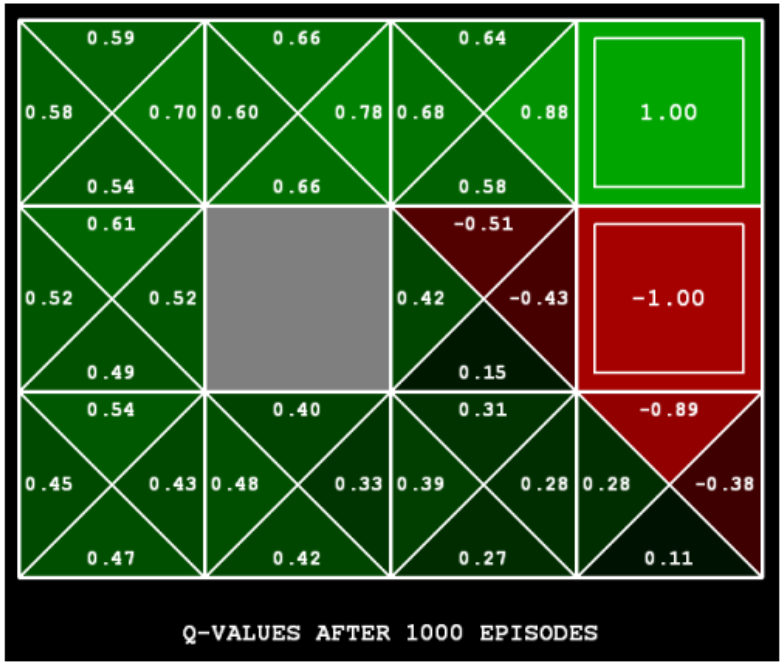
Function *SELECT-ACTION(Q, S, ϵ)* **is**

```
    n ← uniform random number between 0 and 1;
    if n < ϵ then
        A ← random action from the action space;
    else
        A ← maxQ(S,.);
    end
    return selected action A;
end
```

RIT

Q-Tables

- Q-learning produces tables of q-values:



RIT

Problems with Q-Learning

RIT

Problems with Q-Learning

- In realistic situations, we cannot possibly learn about every single state!
  - Too many states to visit them all in training
  - Too many states to hold the q-tables in memory

RIT

Problems with Q-Learning

- In realistic situations, we cannot possibly learn about every single state!
  - Too many states to visit them all in training
  - Too many states to hold the q-tables in memory
- Instead, we want to generalize:
  - Learn about some small number of training states from experience
  - Generalize that experience to new, similar states
  - This is a fundamental idea in machine learning, and we'll see it over and over again

RIT

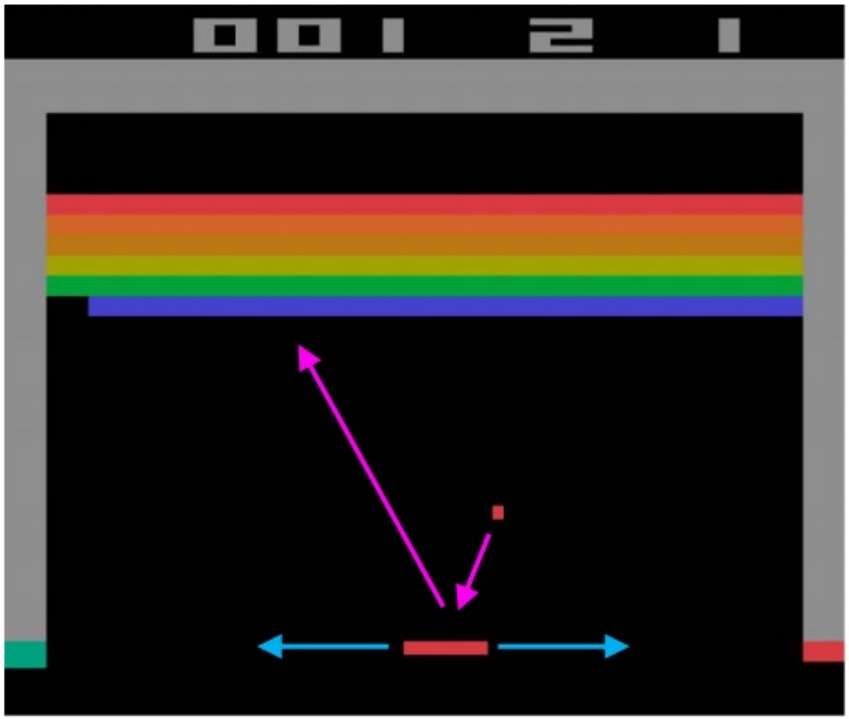
Pacman States

- Let's say we discover through experience that this state is bad:

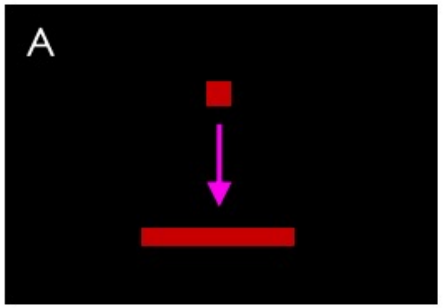
RIT

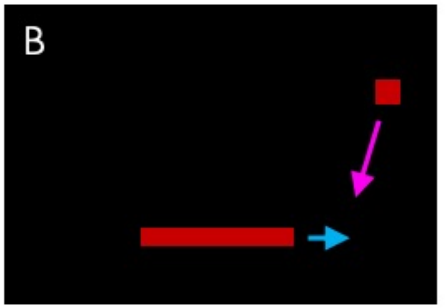
Deep Q-Learning

Example: Atari Breakout



It can be very difficult for humans to accurately estimate Q-values



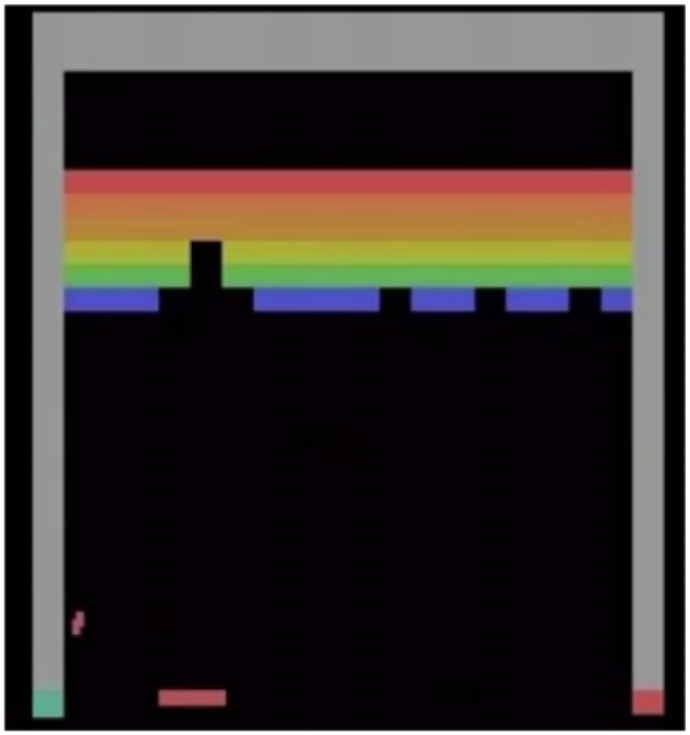


Which (**s**, **a**) pair has a higher Q-value?

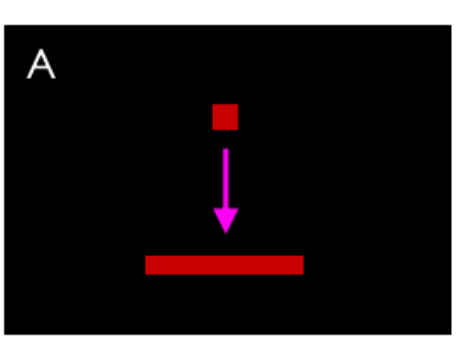
RIT

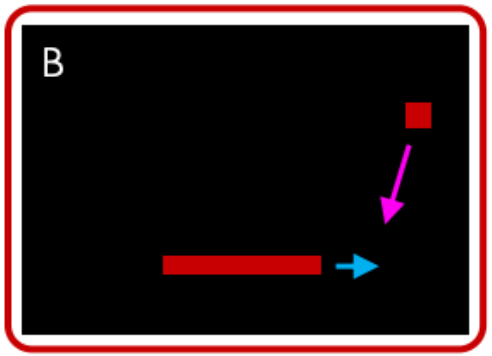
Example

Example: Atari Breakout - Middle



It can be very difficult for humans to accurately estimate Q-values



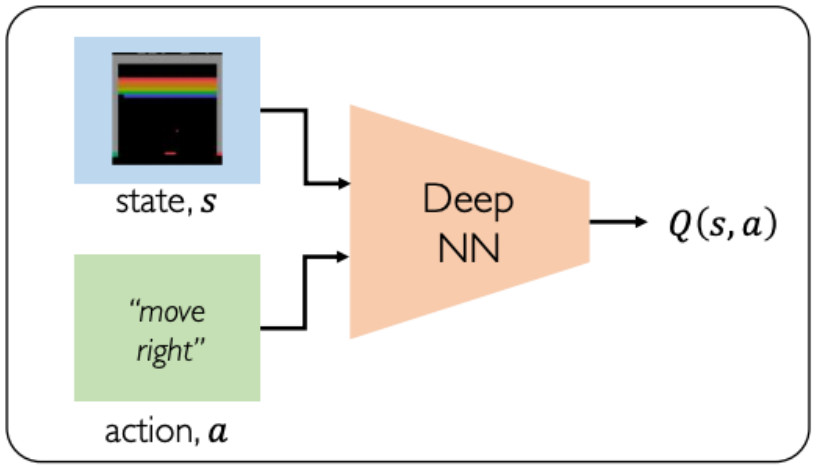


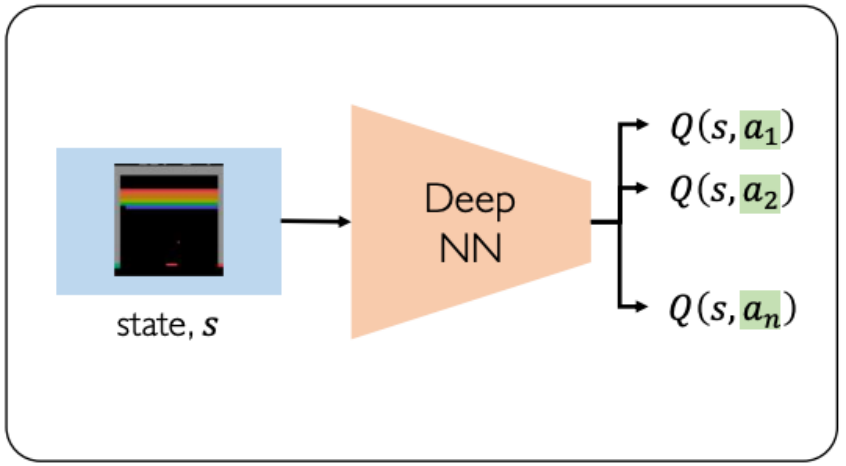
Which (**s**, **a**) pair has a higher Q-value?

RIT

How to use Deep Learning

How can we use deep neural networks to model Q-functions?

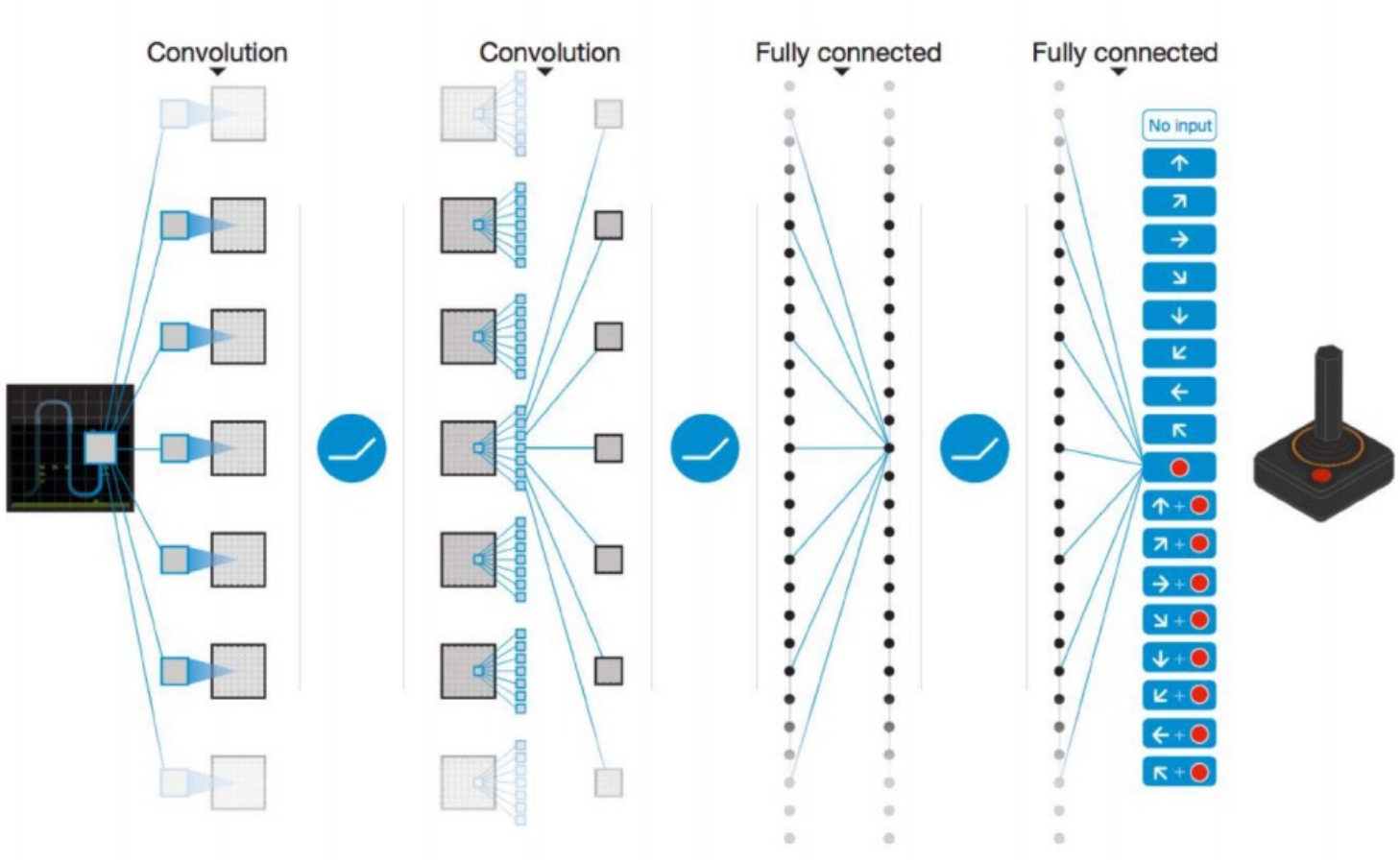




$$\mathcal{L} = \mathbb{E}\left[\left\|\overbrace{\left(r + \gamma \max_{a'} Q(s', a')\right)}^{\text{target}} - \overbrace{Q(s, a)}^{\text{predicted}}\right\|^2\right]$$

RIT

Deep-Q Learning Architecture



RIT Q-Learning Limitations

Complexity:

- Can model scenarios where the action space is discrete and small
- Cannot handle continuous action spaces

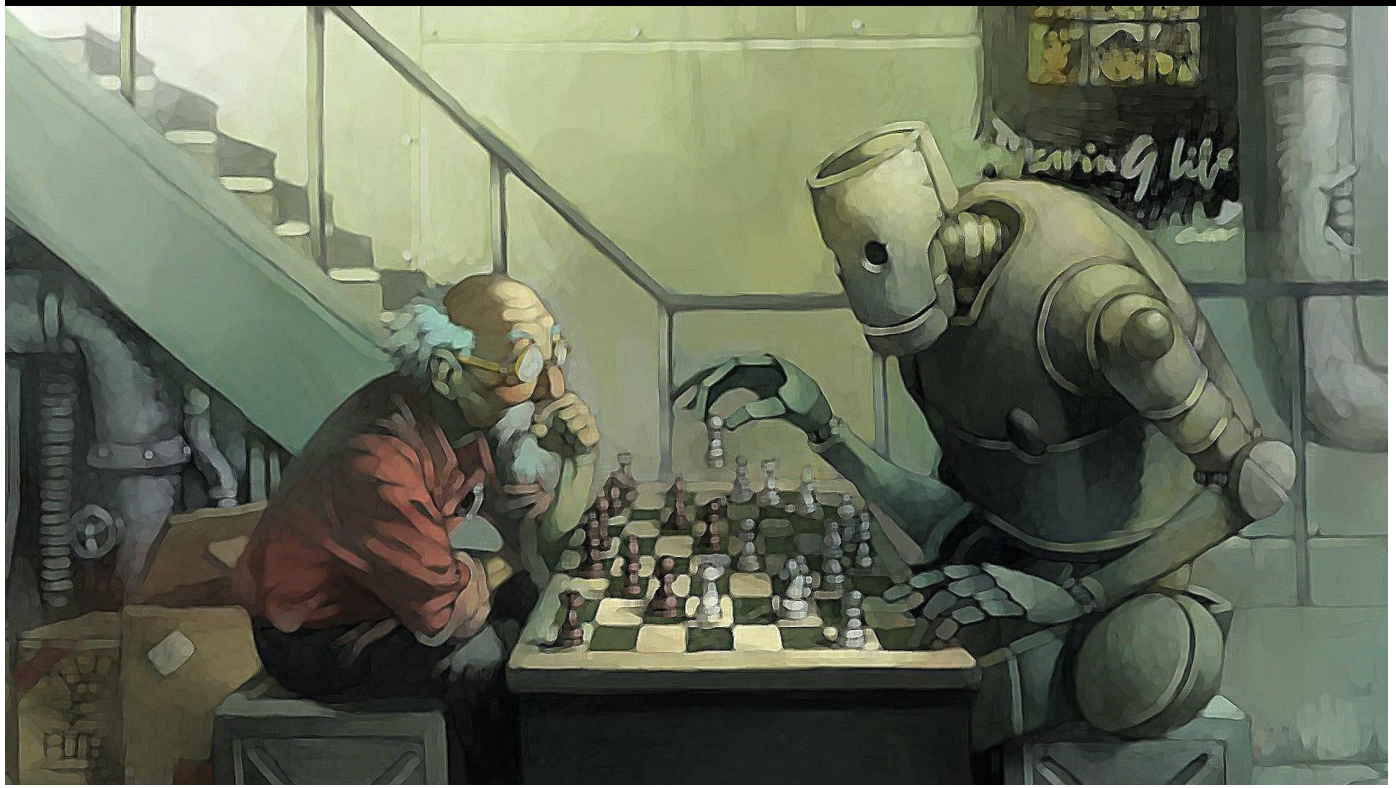
IMPORTANT:
Imagine you want to predict steering wheel angle of a car!

Flexibility:

- Cannot learn stochastic policies since policy is deterministically computed from the Q function

RIT

Designing Rewards: Chess



Bad: +1 every piece taken

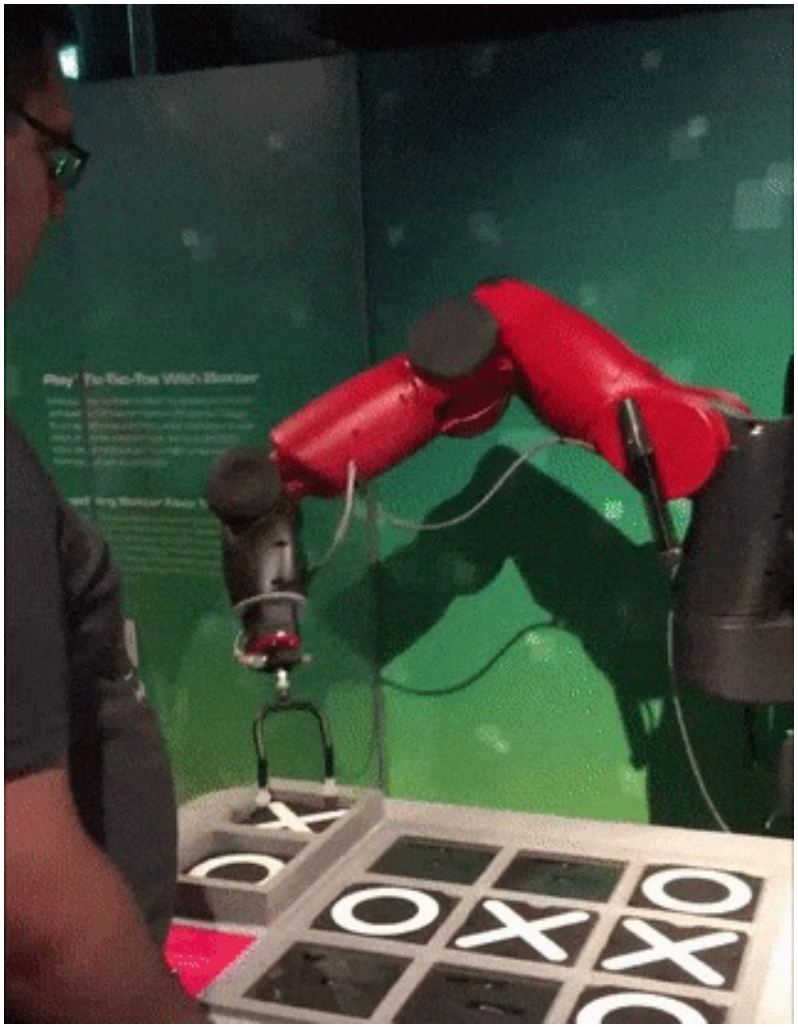
Good: +1 winning

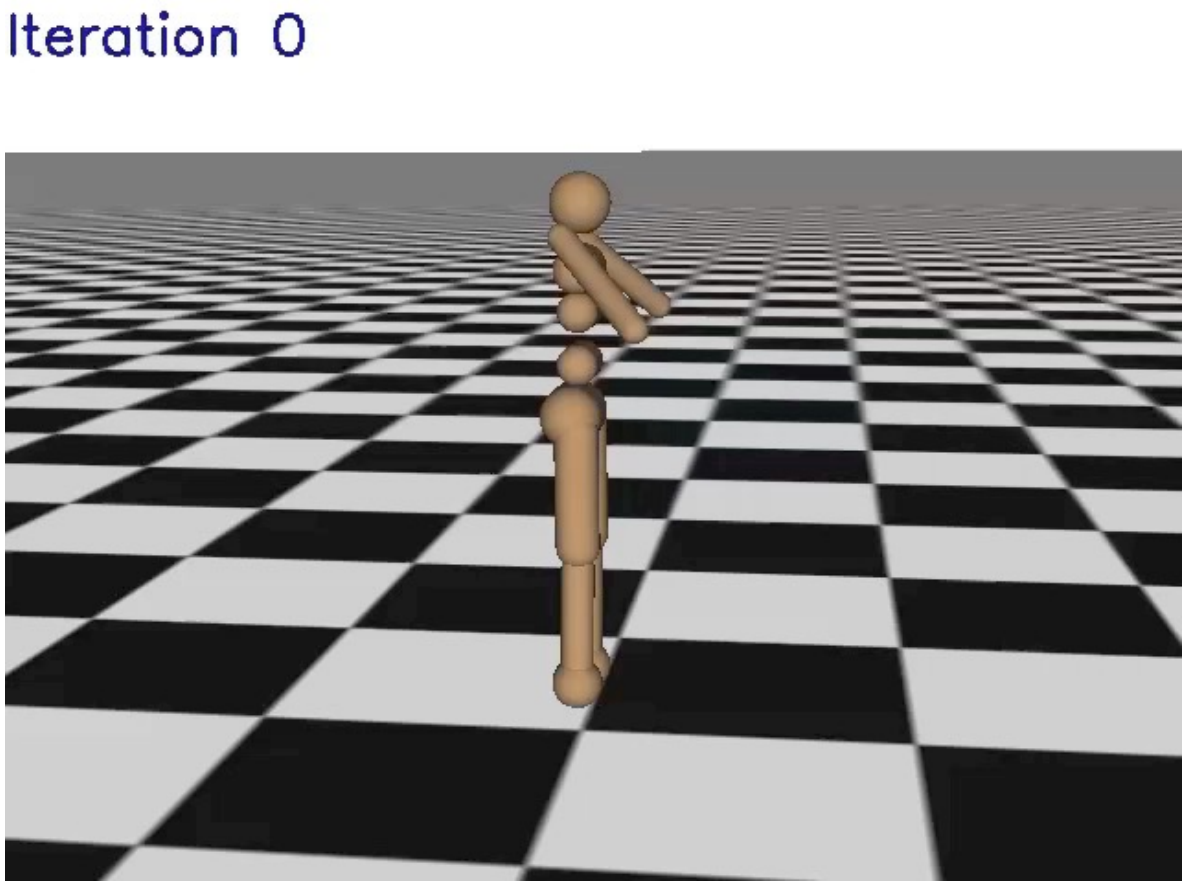

START
START
SCORE
0
LAPS
—/3
TIME
0:01
TURBO
MORE GAMES

RIT

RIT

RIT




Iteration 0

RIT

After 10 HRS of training


OpenAI
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts in a
conversational way. The dialogue format makes it possibl
ChatGPT to answer followup questions, admit its
hallenge incorrect premises, and reject
ts. ChatGPT is a sibling model to
to follow an instructio

RIT

Group Example

- Problem
- State Space
- Action Space
- Overall Objective
- Reward