

<https://introml.mit.edu/>

6.390 Intro to Machine Learning

Lecture 10: Markov Decision Processes

Shen Shen

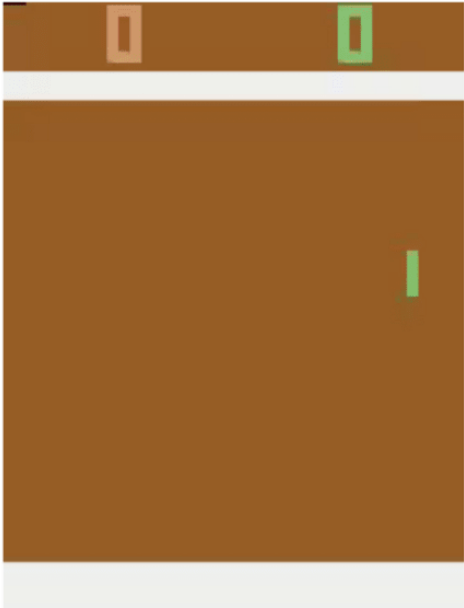
April 19, 2024

Outline

- Recap: Supervised Learning
- Markov Decision Processes
 - Mario example
 - Formal definition
 - Policy Evaluation
 - State-Value Functions: V -values
 - Finite horizon (recursion) and infinite horizon (equation)
 - Optimal Policy and Finding Optimal Policy
 - General tool: State-action Value Functions: Q -values
 - Value iteration

https://shenshen.mit.edu/demos/gifs/russ_toddler.gif

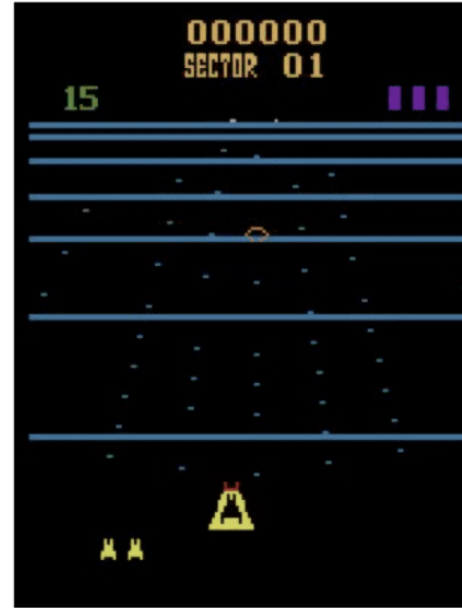
Toddler demo, Russ Tedrake thesis, 2004
(Uses vanilla policy gradient (actor-critic))



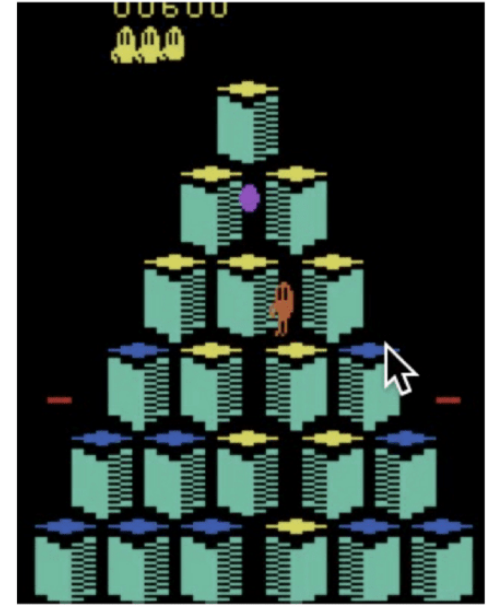
Pong



Enduro



Beamrider

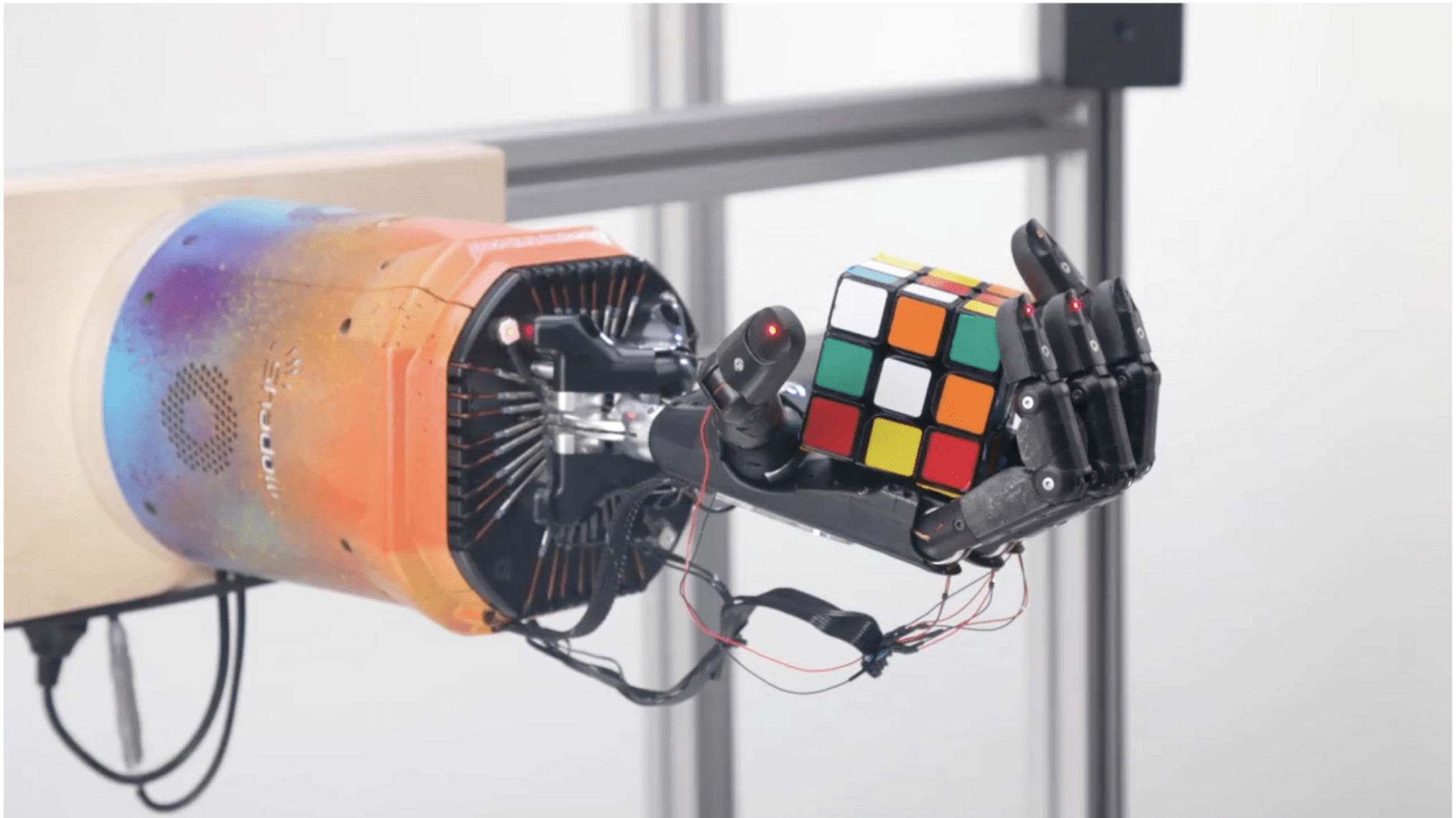


Q*bert

[Human-level control through deep reinforcement learning. Mnih et al. Nature 2015]



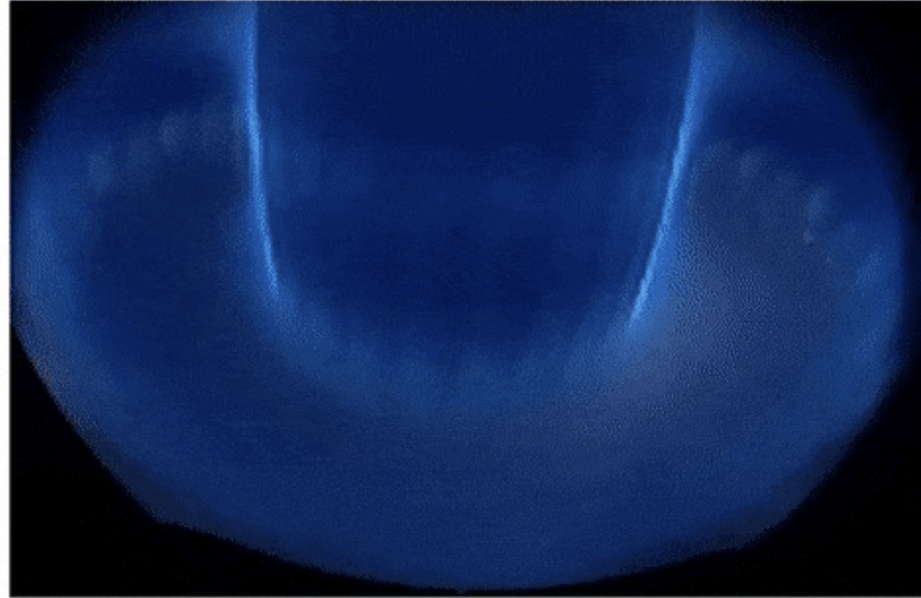
[Mastering the game of Go with deep neural networks and tree search. Silver et al. Nature 2016]



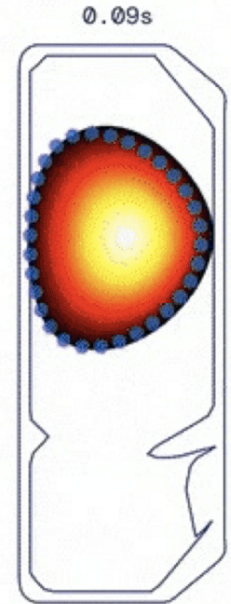
[Solving Rubik's cube with a robot hand. OpenAI. 2019]



Photo Credits: DeepMind and SPC/EPFL

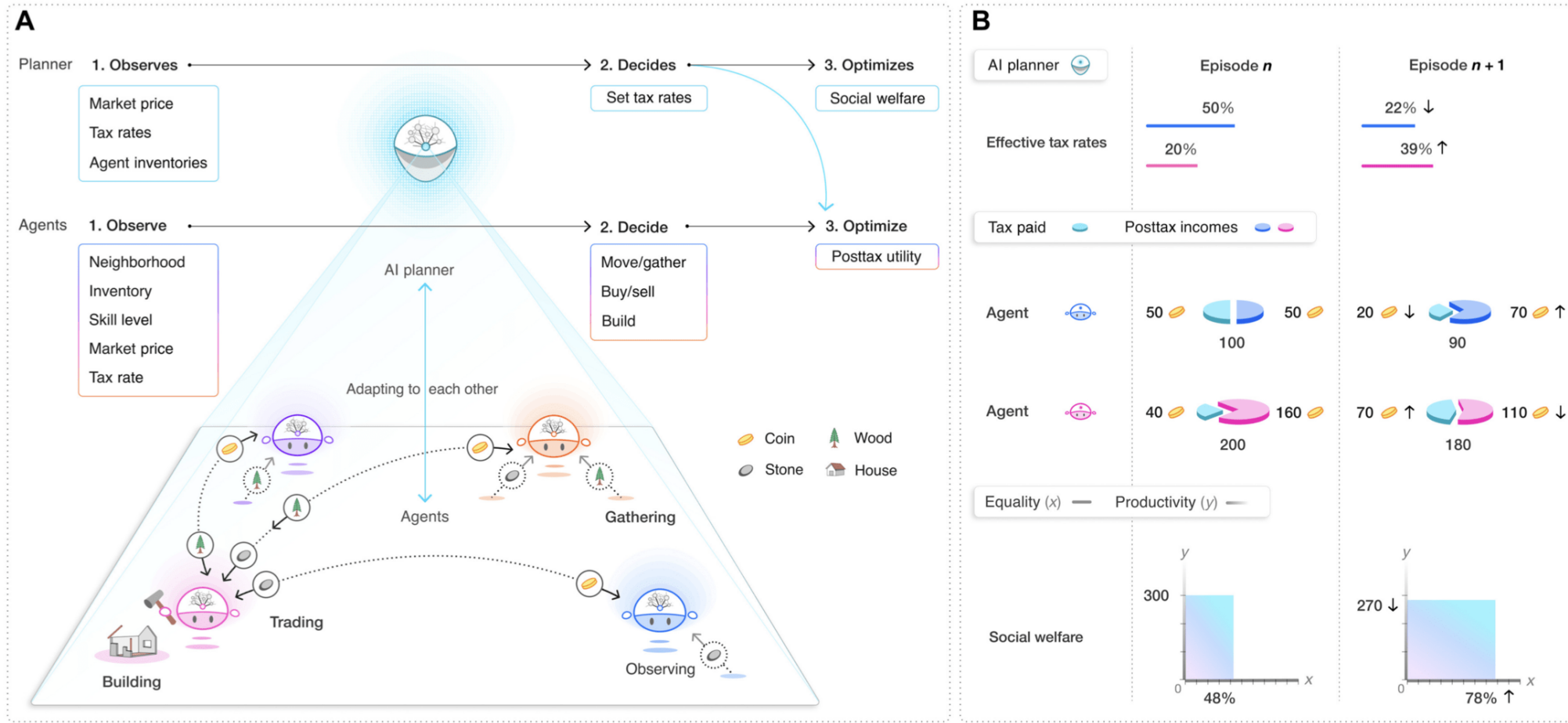


View from inside the tokamak



Plasma state reconstruction

[Magnetic control of tokamak plasmas through deep reinforcement learning. Degraeve et al. Nature 2022]



[The AI Economist: Taxation policy design via two-level deep multiagent RL. Zheng et al. Science 2022]

Optimizing risk-based breast cancer screening policies with reinforcement learning

Adam Yala , Peter G. Mikhael, Constance Lehman, Gigin Lin, Fredrik Strand, Yung-Liang Wan, Kevin Hughes, Siddharth Satuluru, Thomas Kim, Imon Banerjee, Judy Gichoya, Hari Trivedi & Regina Barzilay

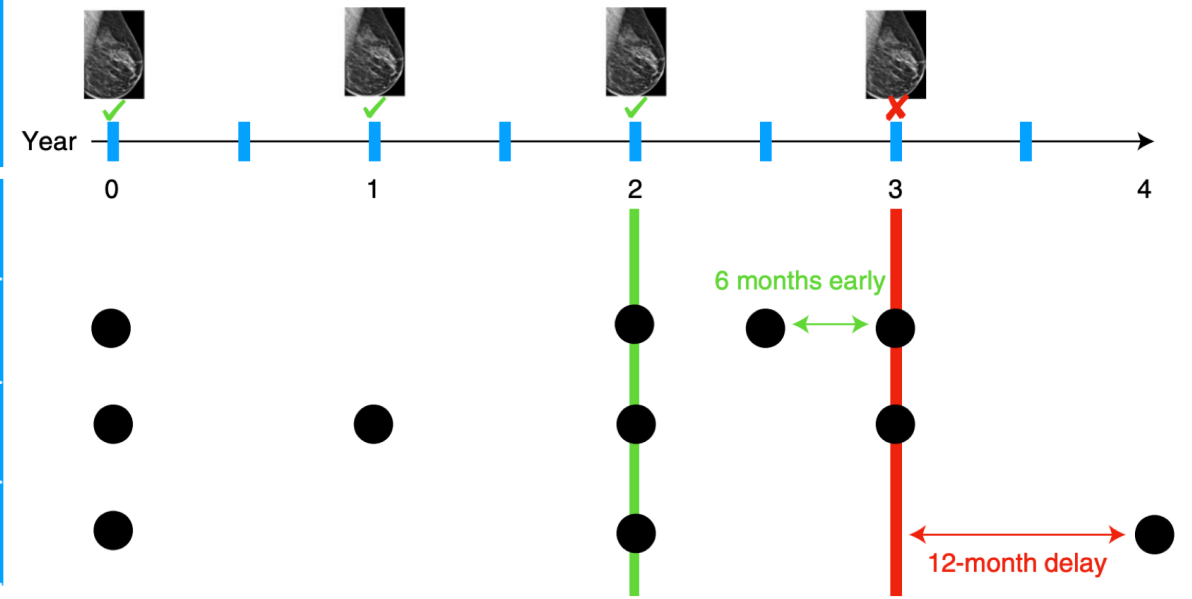
Nature Medicine 28, 136–143 (2022) | Cite this article

8291 Accesses | 24 Citations | 67 Altmetric | Metrics

Abstract

Screening programs must balance the benefit of early detection with the cost of overscreening. Here, we introduce a novel reinforcement learning-based framework for personalized screening, Tempo, and demonstrate its efficacy in the context of breast cancer. We trained our risk-based screening policies on a large screening mammography dataset from Massachusetts General Hospital (MGH; USA) and validated this dataset in held-out patients from MGH and external datasets from Emory University (Emory; USA), Karolinska Institute (Karolinska; Sweden) and Chang Gung Memorial Hospital (CGMH; Taiwan). Across all test sets, we find that the Tempo policy combined with an image-based artificial intelligence (AI) risk model is significantly more efficient than current regimens used in clinical practice in terms of simulated early detection per screen frequency. Moreover, we show that the same Tempo policy can be easily adapted to a wide range of possible screening preferences, allowing clinicians to select their desired trade-off between early detection and screening costs without training new policies. Finally, we demonstrate that Tempo policies based on AI-based risk models outperform Tempo policies based on less accurate clinical risk models. Altogether, our results show that pairing AI-based risk models with agile AI-designed screening policies has the potential to improve screening programs by advancing early detection while reducing overscreening.

- Retrospective patient trajectory
- Recommended trajectories
- Tempo-mirai
- Annual
- Biennial



Discovering faster matrix multiplication algorithms with reinforcement learning

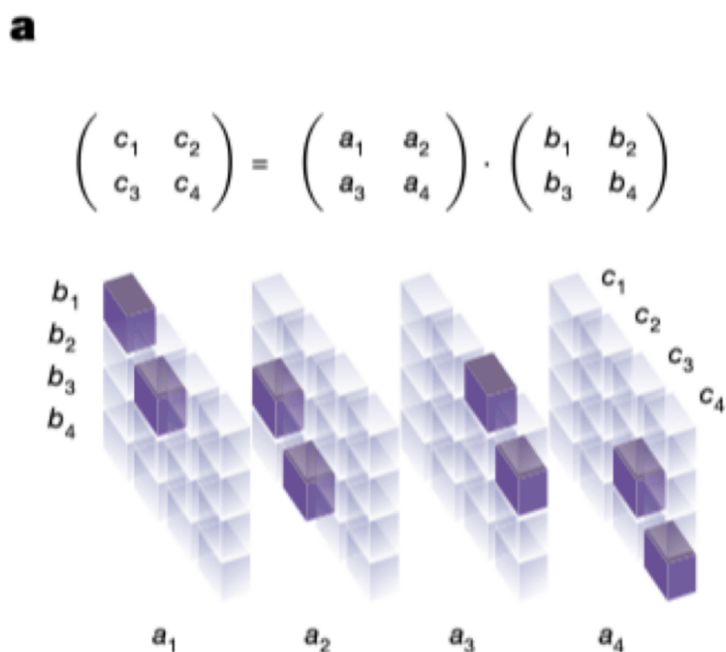
<https://doi.org/10.1038/s41586-022-05172-4>

Received: 2 October 2021

Accepted: 2 August 2022

Published online: 5 October 2022

Alhussein Fawzi^{1,2✉}, Matej Balog^{1,2}, Aja Huang^{1,2}, Thomas Hubert^{1,2}, Bernardino Romera-Paredes^{1,2}, Mohammadamin Barekatin¹, Alexandre Francisco J. R. Ruiz¹, Julian Schrittwieser¹, Grzegorz Swirszcz¹, David Si & Pushmeet Kohli¹



b

$$\begin{aligned} m_1 &= (a_1 + a_4)(b_1 + b_4) \\ m_2 &= (a_3 + a_4)b_1 \\ m_3 &= a_1(b_2 - b_4) \\ m_4 &= a_4(b_3 - b_1) \\ m_5 &= (a_1 + a_2)b_4 \\ m_6 &= (a_3 - a_1)(b_1 + b_2) \\ m_7 &= (a_2 - a_4)(b_3 + b_4) \\ c_1 &= m_1 + m_4 - m_5 + m_7 \\ c_2 &= m_3 + m_5 \\ c_3 &= m_2 + m_4 \\ c_4 &= m_1 - m_2 + m_3 + m_6 \end{aligned}$$

c

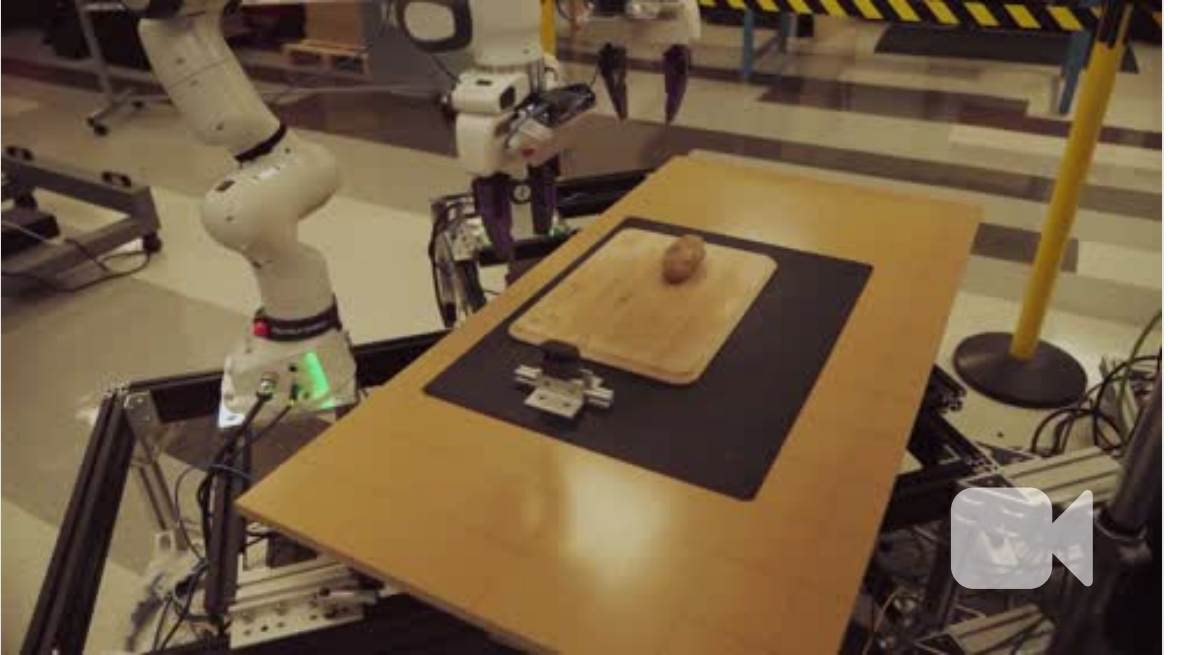
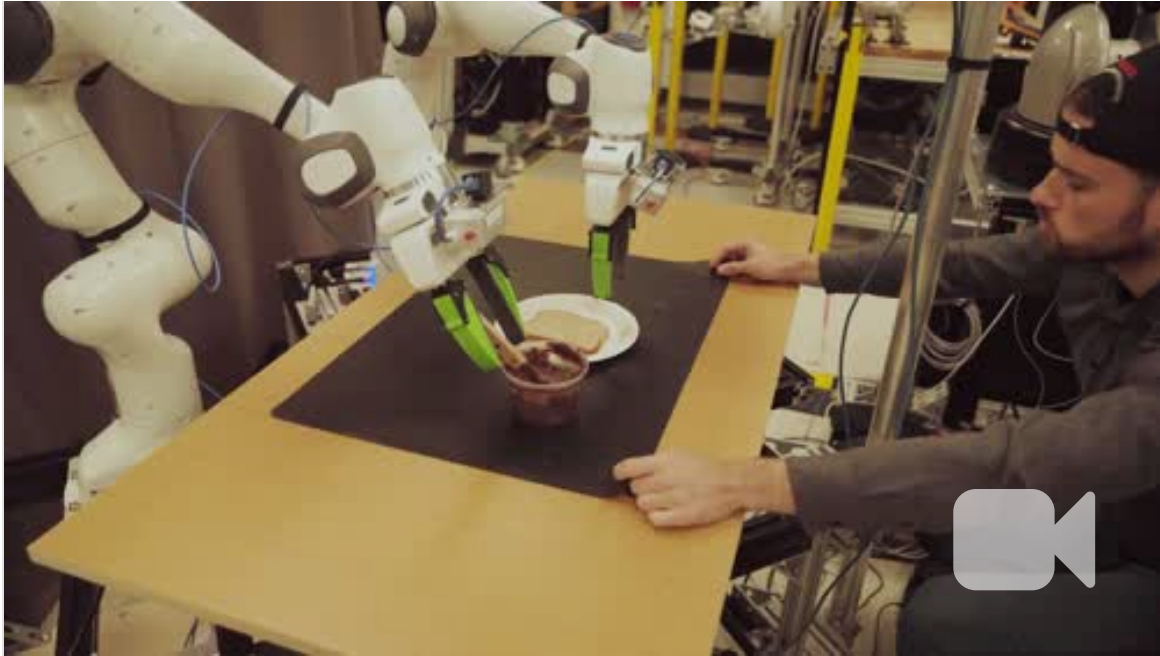
$$\mathbf{U} = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} \text{---} \\ \text{---} \\ \text{---} \end{pmatrix}$$

Size (n, m, p)	Best method known	Best rank known	AlphaTensor rank Modular	rank Standard
(2, 2, 2)	(Strassen, 1969) ²	7	7	7
(3, 3, 3)	(Laderman, 1976) ¹⁵	23	23	23
(4, 4, 4)	(Strassen, 1969) ² (2, 2, 2) \otimes (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971) ¹⁶	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971) ¹⁶	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971) ¹⁶	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971) ¹⁶	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971) ¹⁶	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971) ¹⁶	40	40	40
(3, 3, 4)	(Smirnov, 2013) ¹⁸	29	29	29
(3, 3, 5)	(Smirnov, 2013) ¹⁸	36	36	36
(3, 4, 4)	(Smirnov, 2013) ¹⁸	38	38	38
(3, 4, 5)	(Smirnov, 2013) ¹⁸	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021) ¹⁹	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) \otimes (2, 1, 1)	80	76	76

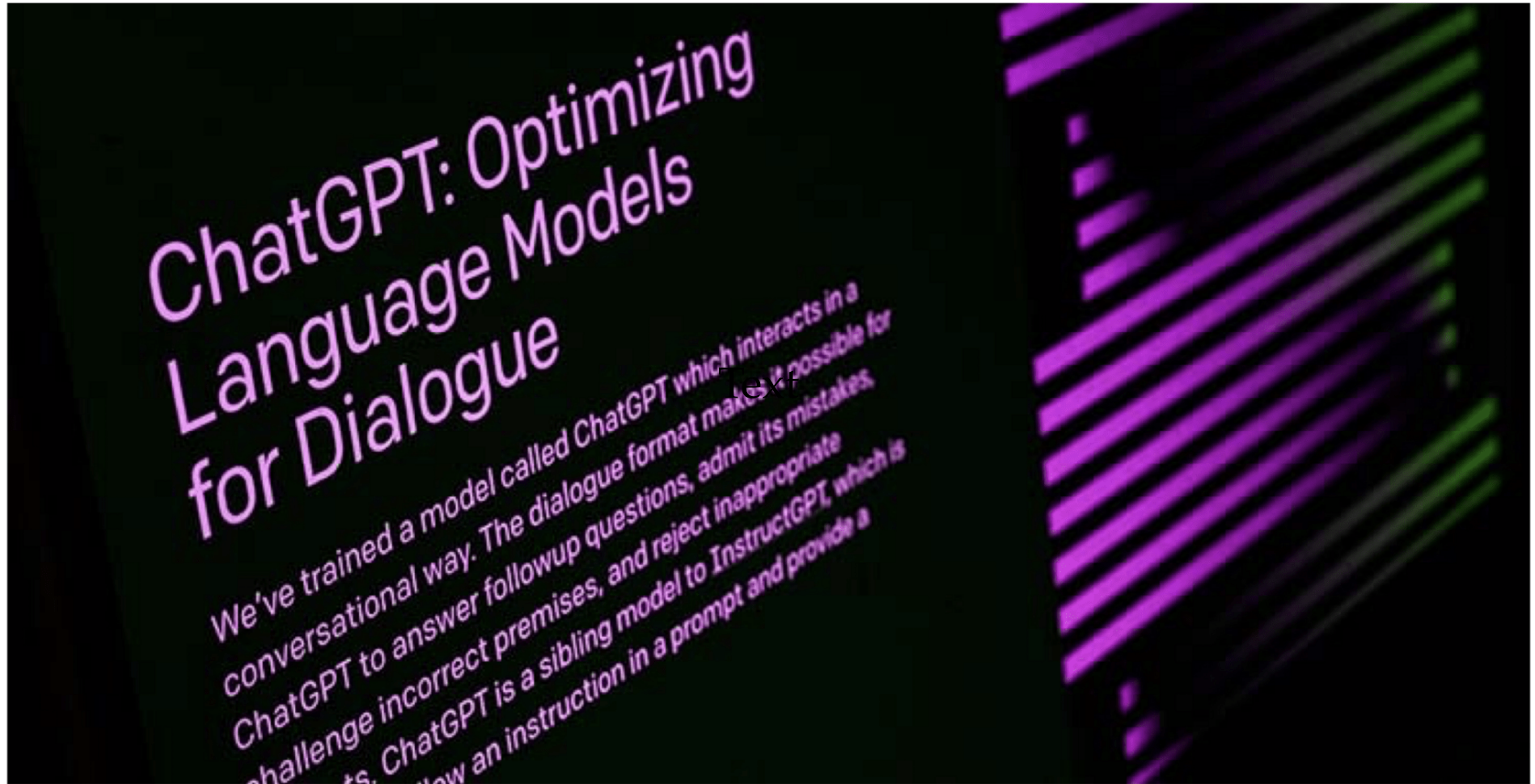
X



(The demo won't embed in PDF. But the direct link below works.)

<https://learning-to-paint.github.io>

Reinforcement Learning with Human Feedback



[Aligning language models to follow instructions. Ouyang et al. 2022]

Markov Decision Processes

- Foundational tools and concept to understand RL.
- Research area initiated in the 1950s (Bellman), known under various names (in various communities):
 - Stochastic optimal control (Control theory)
 - Stochastic shortest path (Operations research)
 - Sequential decision making under uncertainty (Economics)
 - Dynamic programming, control of dynamical systems (under uncertainty)
 - Reinforcement learning (Artificial Intelligence, Machine Learning)
- A rich variety of (accessible & elegant) theory / math, algorithms, and applications / illustrations
- As a result, quite a large variations of notations.
- We will use the most RL-flavored notation



Running example: Mario in a grid-world

1	2	3
4	5	6
7	8	9

- 9 possible **states**
- 4 possible **actions**: {Up \uparrow , Down \downarrow , Left \leftarrow , Right \rightarrow }
- almost all **transitions** are deterministic:
 - Normally, actions take Mario to the “intended” state.
 - E.g., in state (7), action “ \uparrow ” gets to state (4)
 - If an action would've taken us out of this world, stay put
 - E.g., in state (9), action “ \rightarrow ” gets back to state (9)
 - except, in state (6), action “ \uparrow ” leads to two possibilities:
 - 20% chance ends in (2)
 - 80% chance ends in (3)



example cont'd



1	2	3
4	5	6
7	8	9

Transitions from state 2: 80% to state 3 (up), 20% to state 4 (down)

		1
		1 1
		1
		-10
		-10 -10
		-10

reward of being in state s , taking action a

- (state, action) pair can get Mario rewards:

- In state (3), any action gets reward +1 
- In state (6), any action gets reward -10 
- Any other (state, action) pairs get reward 0

actions: {Up ↑, Down ↓, Left ←, Right →}

- goal is to find a gameplay strategy for Mario, to
 - get maximum sum of rewards
 - get these rewards **as soon as possible**

Definition and Goal

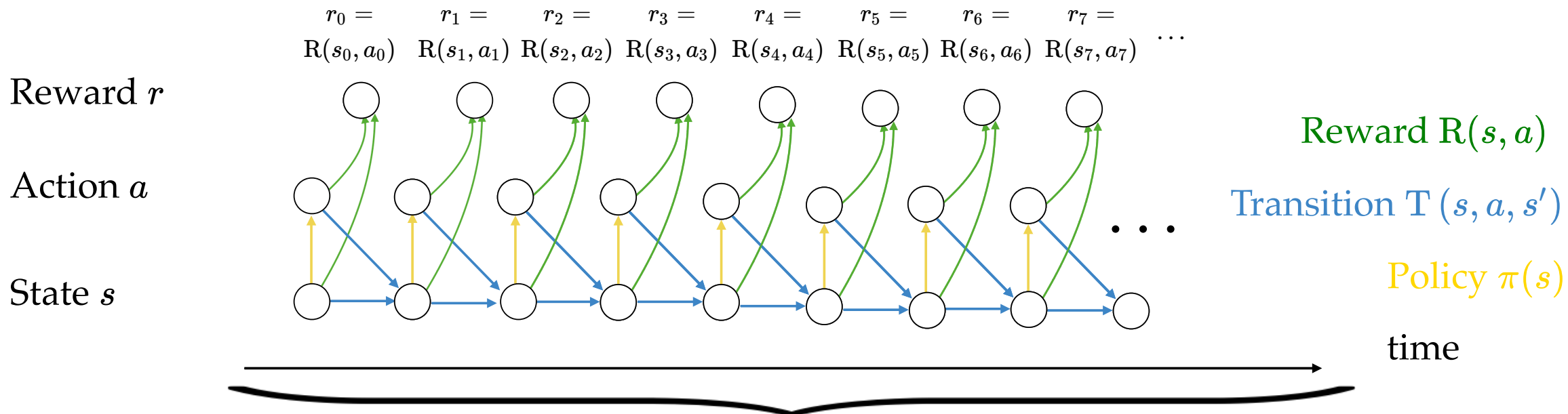
- \mathcal{S} : state space, contains all possible states s .
- \mathcal{A} : action space, contains all possible actions a .
- $T(s, a, s')$: the probability of transition from state s to s' when action a is taken.
- $R(s, a)$: a function that takes in the (state, action) and returns a reward.
- $\gamma \in [0, 1]$: discount factor, a scalar.

- $\pi(s)$: policy, takes in a state and returns an action.

Ultimate goal of an MDP: Find the "best" policy π .

Sidenote:

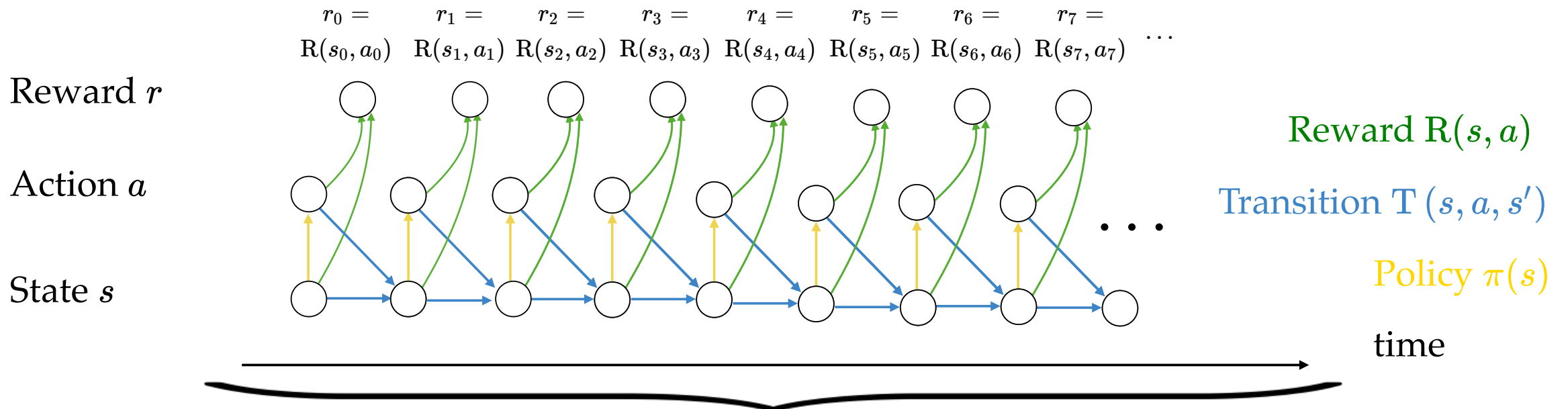
- In 6.390, $R(s, a)$ is deterministic and bounded.
- In 6.390, $\pi(s)$ is deterministic.
- In this week, \mathcal{S} and \mathcal{A} are discrete set, i.e. have finite elements (in fact, typically quite small)



a trajectory (aka an experience or rollout) $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$

how "good" is a trajectory?

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \gamma^3 R(s_3, a_3) + \gamma^4 R(s_4, a_4) + \gamma^5 R(s_5, a_5) + \gamma^6 R(s_6, a_6) + \gamma^7 R(s_7, a_7) \dots$$



a trajectory (aka an experience or rollout) $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$

- Now, suppose h the horizon (how many time steps), and s_0 the initial state are given.
- Also, recall the rewards $R(s, a)$ and policy $\pi(s)$ are deterministic.
- There would still be randomness in a trajectory, due to stochastic transition.
- That is, we cannot just evaluate

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \gamma^3 R(s_3, a_3) + \gamma^4 R(s_4, a_4) + \gamma^5 R(s_5, a_5) + \gamma^6 R(s_6, a_6) + \gamma^7 R(s_7, a_7) \dots$$



For a given policy $\pi(s)$, the finite-horizon horizon- h (state) value functions are:

$$V_{\pi}^h(s) := \mathbb{E} \left[\sum_{t=0}^{h-1} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s, \pi \right], \forall s, h$$

$$\mathbb{E} \left[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \gamma^3 R(s_3, a_3) + \gamma^4 R(s_4, a_4) + \gamma^5 R(s_5, a_5) + \gamma^6 R(s_6, a_6) + \gamma^7 R(s_7, a_7) \dots \right]$$

- **expected** sum of discounted rewards, for starting in state s , following policy $\pi(s)$, for horizon h .
- expectation w.r.t. stochastic transition.
- horizon-0 values are all 0.
- value is a long-term thing, reward is a one-time thing.



example: evaluating the "always ↑" policy

Recall:

1	2	3
4	5	6
7	8	9

$\pi(s) = \text{"\u2191"}, \forall s$

$R(3, \uparrow) = 1$

$R(6, \uparrow) = -10$

$R(s, \uparrow) = 0$ for all other seven states

$$V_{\pi}^h(s) := \mathbb{E} \left[\sum_{t=0}^{h-1} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s, \pi \right], \forall s, h$$

Suppose $\gamma = 0.9$

$$\mathbb{E} \left[\underbrace{R(s_0, a_0) + .9R(s_1, a_1) + (.9)^2 R(s_2, a_2) \dots}_{h \text{ terms inside}} \right]$$

- Horizon $h = 0$; nothing happens

$V_{\pi}^0(s)$

0	0	0
0	0	0
0	0	0

$V_{\pi}^1(s)$

- Horizon $h = 1$: simply receiving the rewards

0	0	1
0	0	-10
0	0	0

Recall:

1	2	3
4	5	6
7	8	9

$\pi(s) = \text{“}\uparrow\text{”}, \forall s$

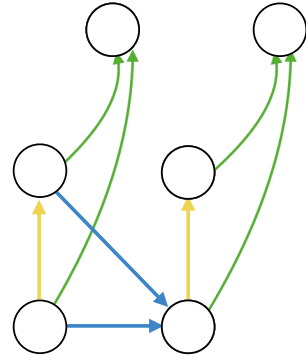
$R(3, \uparrow) = 1$

$R(6, \uparrow) = -10$

$\gamma = 0.9$

- Horizon $h = 2$

$R(s_0, a_0) \quad \gamma R(s_1, a_1)$



$$V_{\pi}^h(s) := \mathbb{E} \left[\sum_{t=0}^{h-1} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s, \pi \right]$$

$$\mathbb{E} \left[R(s_0, a_0) + .9R(s_1, a_1) \right]$$

2 terms inside

$V_{\pi}^2(s)$

Recall:

1	2	3
4	5	6
7	8	9

$\pi(s) = \text{“}\uparrow\text{”}, \forall s$

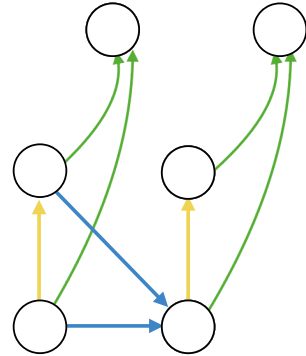
$R(3, \uparrow) = 1$

$R(6, \uparrow) = -10$

$\gamma = 0.9$

• Horizon $h = 2$

$R(s_0, a_0) \quad \gamma R(s_1, a_1)$



$$V_{\pi}^h(s) := \mathbb{E} \left[\sum_{t=0}^{h-1} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s, \pi \right]$$

$$\mathbb{E} \left[R(s_0, a_0) + .9R(s_1, a_1) \right]$$

2 terms inside

if $s_0 = 1$, receive $R(1, \uparrow) + \gamma R(1, \uparrow)$

if $s_0 = 2$, receive $R(2, \uparrow) + \gamma R(2, \uparrow)$

if $s_0 = 3$, receive $R(3, \uparrow) + \gamma R(3, \uparrow)$

if $s_0 = 4$, receive $R(4, \uparrow) + \gamma R(1, \uparrow)$

if $s_0 = 5$, receive $R(5, \uparrow) + \gamma R(2, \uparrow)$

if $s_0 = 6$, receive $R(6, \uparrow) + \gamma[.2R(2, \uparrow) + .8R(3, \uparrow)]$

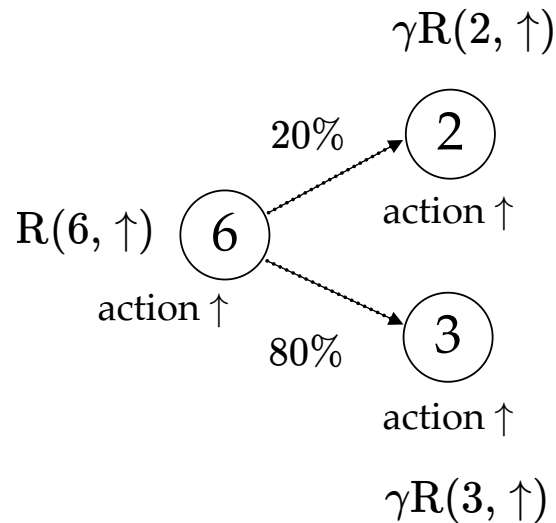
if $s_0 = 7$, receive $R(7, \uparrow) + \gamma R(4, \uparrow)$

if $s_0 = 8$, receive $R(8, \uparrow) + \gamma R(5, \uparrow)$

if $s_0 = 9$, receive $R(9, \uparrow) + \gamma R(6, \uparrow)$

$V_{\pi}^2(s)$

0	0	1.9
0	0	-9.28
0	0	-9



Recall:

1	2	3
4	5	6
7	8	9

$\pi(s) = \text{“}\uparrow\text{”}, \forall s$

$R(3, \uparrow) = 1$

$R(6, \uparrow) = -10$

$\gamma = 0.9$

 $V_{\pi}^0(s)$

0	0	0
0	0	0
0	0	0

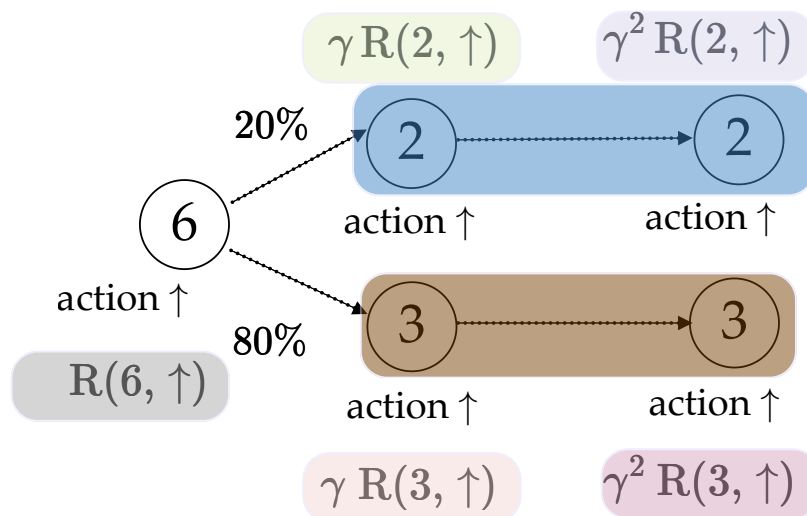
 $V_{\pi}^1(s)$

0	0	1
0	0	-10
0	0	0

 $V_{\pi}^2(s)$

0	0	1.9
0	0	-9.28
0	0	-9

Now, let's think about $V_{\pi}^3(6)$



$$\begin{aligned}
 V_{\pi}^3(6) &= R(6, \uparrow) + 20\% \left[\gamma R(2, \uparrow) + \gamma^2 R(2, \uparrow) \right] + 80\% \left[\gamma R(3, \uparrow) + \gamma^2 R(3, \uparrow) \right] \\
 &= R(6, \uparrow) + 20\% \gamma \left[R(2, \uparrow) + \gamma R(2, \uparrow) \right] + 80\% \gamma \left[R(3, \uparrow) + \gamma R(3, \uparrow) \right] \\
 &= R(6, \uparrow) + 20\% \gamma V_{\pi}^2(2) + 80\% \gamma V_{\pi}^2(3)
 \end{aligned}$$

Bellman Recursion

expected sum of discounted rewards, for starting in state s , follow policy $\pi(s)$ for horizon h

$$V_{\pi}^h(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s'} \mathbf{T}(s, \pi(s), s') V_{\pi}^{h-1}(s')$$

immediate reward, for being in state s and taking the action given by policy $\pi(s)$

$(h - 1)$ horizon values at a next state s'

weighted by the probability of getting to that next state s'

discounted by γ

finite-horizon policy evaluation

For a given policy $\pi(s)$, the finite-horizon horizon- h (state) value functions are:

$$V_{\pi}^h(s) := \mathbb{E} \left[\sum_{t=0}^{h-1} \gamma^t \mathbf{R}(s_t, \pi(s_t)) \mid s_0 = s, \pi \right], \forall s$$

Bellman recursion

$$V_{\pi}^h(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s'} \mathbf{T}(s, \pi(s), s') V_{\pi}^{h-1}(s'), \forall s$$

infinite-horizon policy evaluation

For any given policy $\pi(s)$, the infinite-horizon (state) value functions are

$$V_{\pi}(s) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{R}(s_t, \pi(s_t)) \mid s_0 = s, \pi \right], \forall s$$

γ is now necessarily < 1 for convergence too

Bellman equation

$$V_{\pi}(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s'} \mathbf{T}(s, \pi(s), s') V_{\pi}(s'), \forall s$$

- $|\mathcal{S}|$ many linear equations

Optimal policy π^*

- Definition of π^* : for any given horizon h (possibly infinite horizon), $V_{\pi^*}^h(s) \geq V_{\pi}^h(s)$ for all $s \in \mathcal{S}$ and for all possible policy π .
- For a fixed MDP, optimal values $V_{\pi^*}^h(s)$ must be unique.
- Optimal policy π^* might not be unique. (Think e.g. symmetric)
- In finite horizon, optimal policy depends on horizon.
- In infinite horizon, horizon no longer matter. Exist a stationary optimal policy.

(Optimal) state-action value functions $Q^h(s, a)$

$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

V values vs. Q values

- V is defined over state space; Q is defined over (state, action) space.
- Any policy can be evaluated to get V values; whereas Q per our definition, has the sense of "tail optimality" baked in.
- $V_{\pi^*}^h(s)$ can be derived from $Q^h(s, a)$, and vice versa.
- Q is easier to read "optimal actions" from.



example: recursively finding $Q^h(s, a)$

$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

$Q^0(s, a)$

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

$Q^1(s, a)$

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Transitions: 2 to 3 (80%), 2 to 5 (20%), 5 to 6 (100%)

$R(s, a)$

		1
		1
		-10
		-10
		-10



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states (1-9). A dashed arrow points from state 3 to state 2, labeled '20%'. A solid arrow points from state 3 to state 6, labeled '80%'.

Let's consider $Q^2(3, \rightarrow)$

- receive $R(3, \rightarrow)$

- next state $s' = 3$, act **optimally** for the remaining one timestep
 - receive $\max_{a'} Q^1(3, a')$

$$Q^2(3, \rightarrow) = R(3, \rightarrow) + \gamma \max_{a'} Q^1(3, a')$$

$$= 1 + .9 \max_{a'} Q^1(3, a')$$

$$= 1.9$$

$$Q^1(s, a) = R(s, a)$$

0	0	0	1	1
0	0	0	0	1
0	0	0	-10	-10
0	0	0	0	-10
0	0	0	0	0
0	0	0	0	0

$$Q^2(s, a)$$

				1.9



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states (1-9). A dashed arrow points from state 2 to state 3, labeled '80%'. A solid arrow points from state 2 to state 5, labeled '20%'.

Let's consider $Q^2(3, \uparrow)$

- receive $R(3, \uparrow)$

- next state $s' = 3$, act **optimally** for the remaining one timestep
 - receive $\max_{a'} Q^1(3, a')$

$$Q^2(3, \uparrow) = R(3, \uparrow) + \gamma \max_{a'} Q^1(3, a')$$

$$= 1 + .9 \max_{a'} Q^1(3, a')$$

$$= 1.9$$

$$Q^1(s, a) = R(s, a)$$

0	0	0	1	1
0	0	0	1	1
0	0	0	-10	-10
0	0	0	-10	-10
0	0	0	0	0
0	0	0	0	0

$$Q^2(s, a)$$

			1.9	1.9



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Special transition: 3 to 2 (80%), 3 to 5 (20%)

Let's consider $Q^2(3, \leftarrow)$

- receive $R(3, \leftarrow)$

- next state $s' = 2$, act **optimally** for the remaining one timestep

- receive $\max_{a'} Q^1(2, a')$

- $Q^2(3, \leftarrow) = R(3, \leftarrow) + \gamma \max_{a'} Q^1(2, a')$

$$= 1 + .9 \max_{a'} Q^1(2, a')$$

$$= 1$$

$$Q^1(s, a) = R(s, a)$$

0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	-10	-10
0	0	0	0	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$$Q^2(s, a)$$

				1.9	1.9
				1	1.9



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Transitions: 3 to 2 (80%), 3 to 5 (20%), 5 to 6 (100%)

Let's consider $Q^2(3, \downarrow)$

- receive $R(3, \downarrow)$

- next state $s' = 6$, act **optimally** for the remaining one timestep

- receive $\max_{a'} Q^1(6, a')$

$$Q^2(3, \downarrow) = R(3, \downarrow) + \gamma \max_{a'} Q^1(6, a')$$

$$= 1 + .9 \max_{a'} Q^1(6, a')$$

$$= -8$$

$$Q^1(s, a) = R(s, a)$$

0	0	0	1	1
0	0	0	0	1
0	0	0	-10	-10
0	0	0	-10	-10
0	0	0	0	0
0	0	0	0	0

$$Q^2(s, a)$$

			1.9	1.9
			1	-8



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. An arrow points from state 6 to state 2, labeled '20%'. Another arrow points from state 6 to state 3, labeled '80%'.

$Q^1(s, a) = R(s, a)$

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Diagram showing a 3x3 grid of states. The middle row (states 4, 5, 6) is highlighted in yellow. The right column (states 2, 3, 6) is highlighted in purple. The values in the grid are 0 for all cells.

$Q^2(s, a)$

					1.9
				1	1.9
					-8
					-9.28

Diagram showing a 3x3 grid of states. The right column (states 2, 3, 6) is highlighted in purple. The values in the grid are: (1,6)=1.9, (2,6)=1.9, (3,6)=-8, (4,6)=-9.28. All other cells are empty.

- receive $R(6, \uparrow)$
- act **optimally** for one more timestep, at the next state s'
 - 20% chance, $s' = 2$, act optimally, receive $\max_{a'} Q^1(2, a')$
 - 80% chance, $s' = 3$, act optimally, receive $\max_{a'} Q^1(3, a')$

Let's consider $Q^2(6, \uparrow) = R(6, \uparrow) + \gamma[.2 \max_{a'} Q^1(2, a') + .8 \max_{a'} Q^1(3, a')]$

$$= -10 + .9[.2 * 0 + .8 * 1] = -9.28$$



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. State 2 is highlighted in yellow. A transition arrow points from state 6 to state 2, labeled '20%'. Another transition arrow points from state 6 to state 3, labeled '80%'.

$Q^1(s, a)$
= $R(s, a)$

0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	-10	-10
0	0	0	0	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$Q^2(s, a)$

				1.9	
				1	1.9
				-8	
				-9.28	

$$Q^2(6, \uparrow) = R(6, \uparrow) + \gamma[.2 \max_{a'} Q^1(2, a') + .8 \max_{a'} Q^1(3, a')]$$

in general $Q^h(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^{h-1}(s', a'), \forall s, a$



$Q^h(s, a)$ is the expected sum of discounted rewards for

- starting in state s ,
- take action a , for one step
- act **optimally** there afterwards for the remaining $(h - 1)$ steps

Recall: $\gamma = 0.9$

States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states (1-9). A transition from state 3 to state 5 is shown with a 20% probability (indicated by a diagonal arrow), and a transition from state 3 to state 6 is shown with an 80% probability (indicated by a vertical arrow).

$Q^1(s, a)$

0	0	0	1	1
0	0	0	1	1
0	0	0	-10	-10
0	0	0	-10	-10
0	0	0	0	0
0	0	0	0	0

$Q^2(s, a)$

		1.9
		1 1.9
		-8
		-9.28

what's the optimal action in state 3, with horizon 2, given by $\pi_2^*(3) = ?$

either up or right

in general

$$\pi_h^*(s) = \arg \max_a Q^h(s, a), \forall s, h$$

Given the finite horizon recursion

$$Q^h(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^{h-1}(s', a')$$

We should easily be convinced of the infinite horizon equation

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q(s', a')$$

Infinite-horizon Value Iteration

1. **for** $s \in \mathcal{S}, a \in \mathcal{A}$:
2. $Q_{\text{old}}(s, a) = 0$
3. **while** True:
4. **for** $s \in \mathcal{S}, a \in \mathcal{A}$:
5. $Q_{\text{new}}(s, a) \leftarrow R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q_{\text{old}}(s', a')$
6. **if** $\max_{s,a} |Q_{\text{old}}(s, a) - Q_{\text{new}}(s, a)| < \epsilon$:
7. **return** Q_{new}
8. $Q_{\text{old}} \leftarrow Q_{\text{new}}$

if instead of relying on line 6 (convergence criterion), we run the block of (line 4 and 5) for h times, then the returned values are exactly horizon- h Q values

We'd appreciate your [feedback](#) on the lecture.

Thanks!