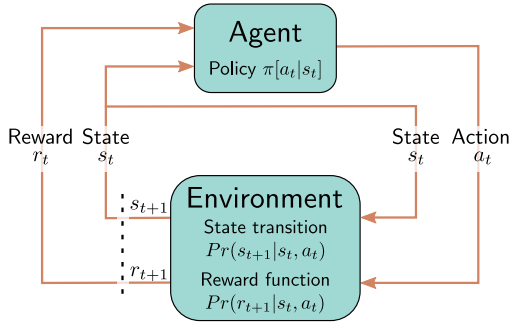
Deep Reinforcement Learning

CS 4277 Deep Learning

Kennesaw State University

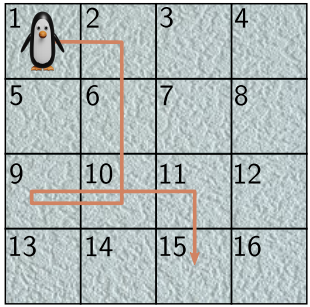
KENNESAW STATE UNIVERSITY

Reinforcement Learning



Markov Process

a)



$s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8$

$\boldsymbol{\tau} = [1, 2, 6, 10, 9, 10, 11, 15]$

b)

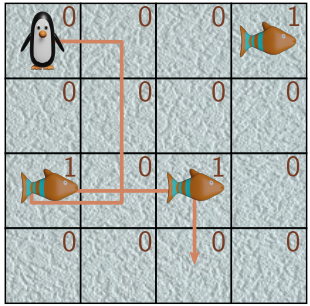
$$Pr(s_{t+1}|s_t)$$

s_t (columns) $\times$ s_{t+1} (rows)

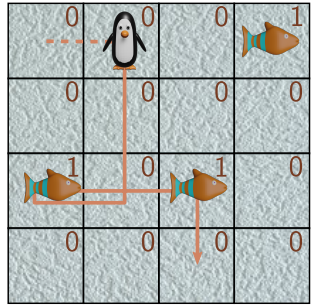
s_{t+1} \ s_t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0.33	0	0	0.33	0	0	0	0	0	0	0	0	0	0	0
2	0.5	0	0.33	0	0	0.25	0	0	0	0	0	0	0	0	0	0
3	0	0.33	0	0.5	0	0	0.25	0	0	0	0	0	0	0	0	0
4	0	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
5	0.5	0	0	0	0	0.25	0	0	0.33	0	0	0	0	0	0	0
6	0	0.33	0	0	0.33	0	0.25	0	0	0.25	0	0	0	0	0	0
7	0	0	0.33	0	0	0.25	0	0.33	0	0	0.25	0	0	0	0	0
8	0	0	0	0.5	0	0	0.25	0	0	0	0	0.33	0	0	0	0
9	0	0	0	0	0.33	0	0	0	0	0.25	0	0	0.5	0	0	0
10	0	0	0	0	0	0.25	0	0	0.33	0	0.25	0	0	0.33	0	0
11	0	0	0	0	0	0	0.25	0	0	0.25	0	0.33	0	0	0.33	0
12	0	0	0	0	0	0	0	0.33	0	0	0.25	0	0	0	0	0.5
13	0	0	0	0	0	0	0	0	0.33	0	0	0	0	0.33	0	0
14	0	0	0	0	0	0	0	0	0	0.25	0	0	0.5	0	0.33	0
15	0	0	0	0	0	0	0	0	0	0	0.25	0	0	0.33	0	0.5
16	0	0	0	0	0	0	0	0	0	0	0	0.33	0	0	0.33	0

Markov Reward Process

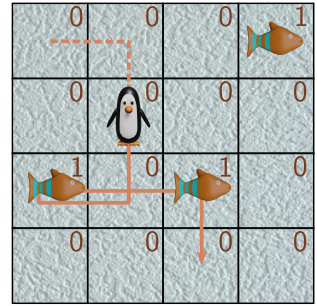
a) $G_1 = 0 + \gamma \cdot 0 + \gamma^2 \cdot 0 + \gamma^3 \cdot 0 + \gamma^4 \cdot 1 + \gamma^5 \cdot 0 + \gamma^6 \cdot 1 + \gamma^7 \cdot 0 = 1.19$



b) $G_2 = 0 + \gamma \cdot 0 + \gamma^2 \cdot 0 + \gamma^3 \cdot 1 + \gamma^4 \cdot 0 + \gamma^5 \cdot 1 + \gamma^6 \cdot 0 = 1.31$



c) $G_3 = 0 + \gamma \cdot 0 + \gamma^2 \cdot 1 + \gamma^3 \cdot 0 + \gamma^4 \cdot 1 + \gamma^5 \cdot 0 = 1.47$

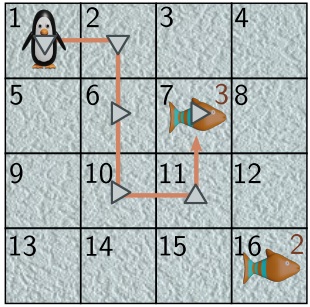


$s_1\ r_2\ s_2\ r_3\ s_3\ r_4\ s_4\ r_5\ s_5\ r_6\ s_6\ r_7\ s_7\ r_8\ s_8\ r_9$

$\tau = [1, 0, 2, 0, 6, 0, 10, 0, 9, 1, 10, 0, 11, 1, 15, 0]$

Markov Decision Process

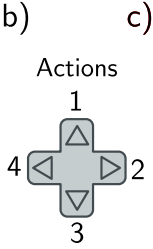
a)



$s_1\ a_1\ r_2\ s_2\ a_2\ r_3\ s_3\ a_3\ r_4\ s_4\ a_4\ r_5\ s_5\ a_5\ r_6\ s_6\ a_6\ r_7$

$\boldsymbol{\tau} = [1, 3, 0, 2, 3, 0, 6, 2, 0, 10, 2, 0, 11, 1, 0, 7, 2, 3]$

b)



c)

$$Pr(s_{t+1}|s_t=6, a_t=1) = \begin{bmatrix} 0 \\ 0.5 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

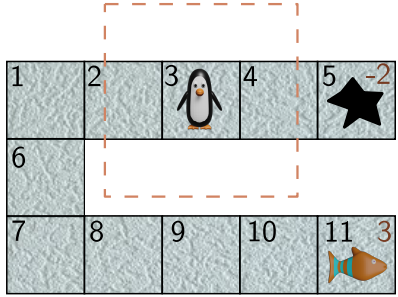
$$Pr(s_{t+1}|s_t=6, a_t=2) = \begin{bmatrix} 0 \\ 0.17 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Pr(s_{t+1}|s_t=6, a_t=3) = \begin{bmatrix} 0 \\ 0.17 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Pr(s_{t+1}|s_t=6, a_t=4) = \begin{bmatrix} 0 \\ 0.17 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0.17 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

KENNESAW STATE UNIVERSITY

Parially-Observable MDP (POMDP)

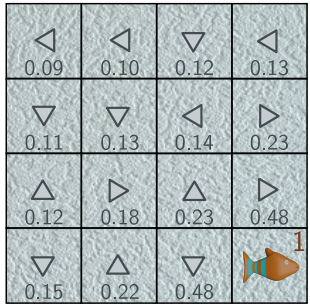


Policy

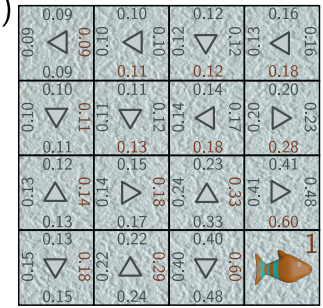
$$\pi(a|s)$$

State and Action Values

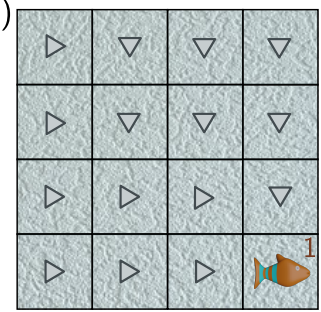
a)



b)



c)

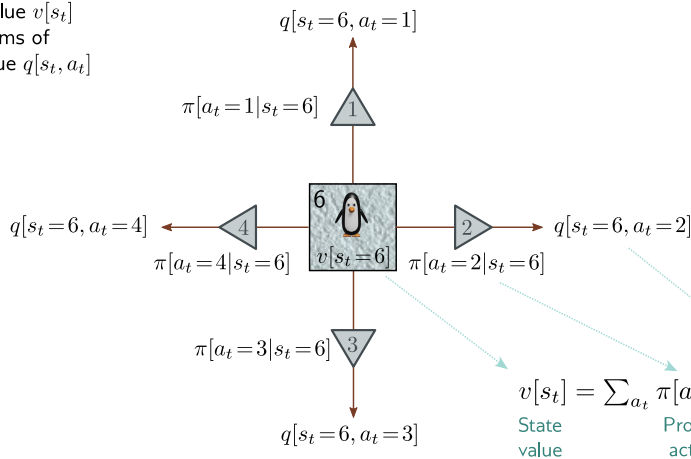


Optimal Policy

$$\pi(a|s)$$

Bellman Equations

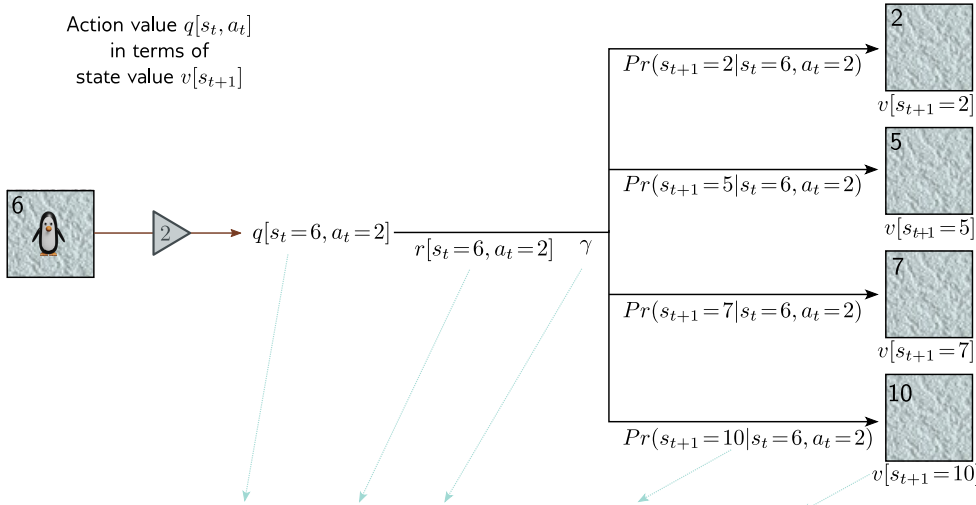
State value $v[s_t]$ in terms of action value $q[s_t, a_t]$



$$v[s_t] = \sum_{a_t} \pi[a_t|s_t] \cdot q[s_t, a_t]$$

State value — Prob of action — Action value

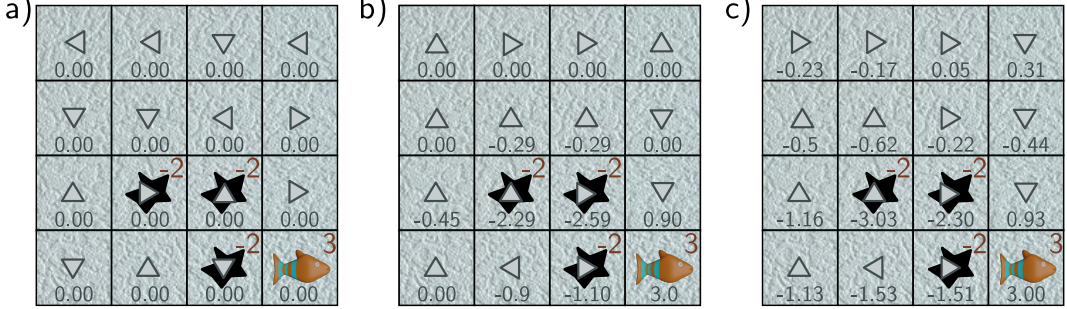
Bellman Equations 2



KENNESAW STATE UNIVERSITY

Tabular RL

Dynamic Programming



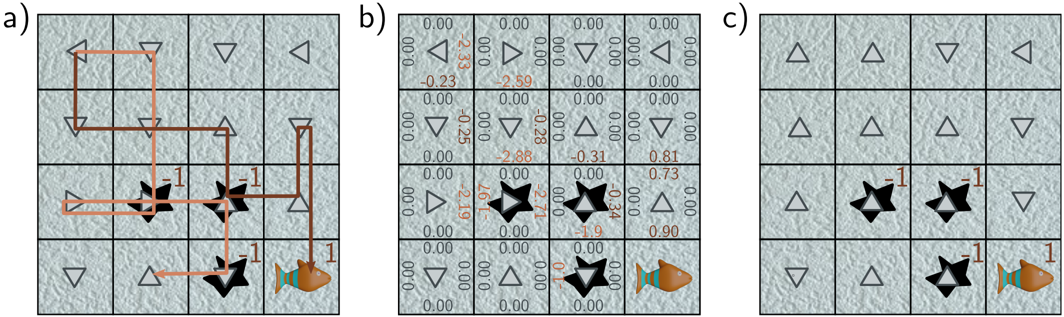
Policy Iteration

$$\pi(a|s)$$

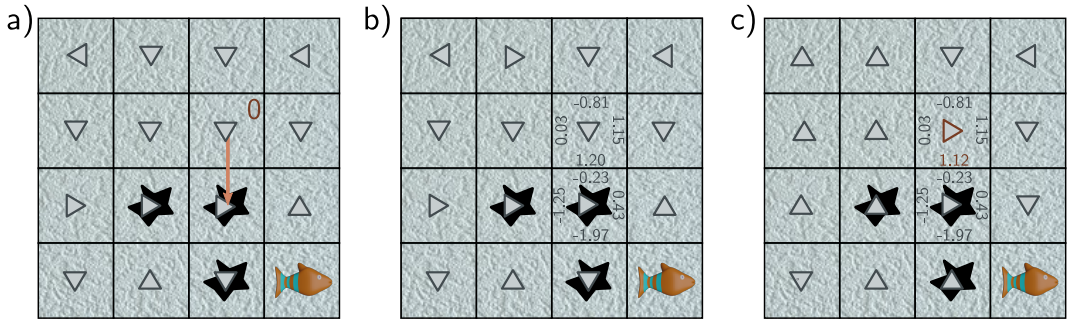
Value Iteration

$$\pi(a|s)$$

Monte Carlo Methods



Temporal Difference Methods



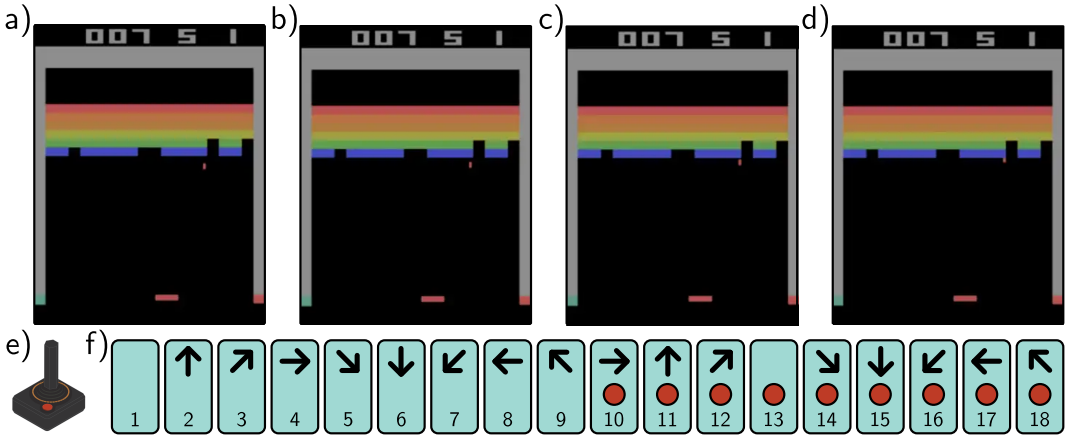
$$q[s_t, a_t] \leftarrow q[s_t, a_t] + \alpha\Big(r[s_t, a_t] + \gamma \cdot \max_a \big[q[s_{t+1}, a]\big] - q[s_t, a_t]\Big)$$

$$1.12 \leftarrow 1.20 + 0.1\Big(0.0 + 0.9 \cdot \max\big[-0.23, 0.43, -1.97, -1.25\big] - 1.20\Big)$$

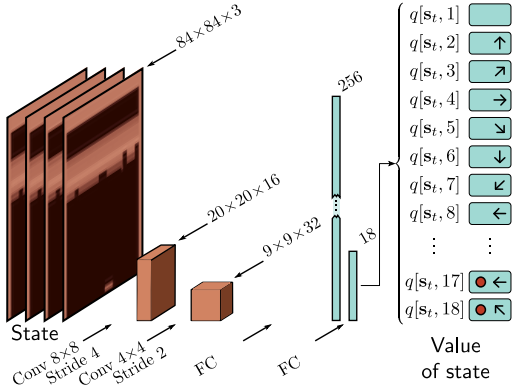
Fitted Q-Learning

$$\pi(a|s)$$

Deep Q-Networks for Playing Atari Games



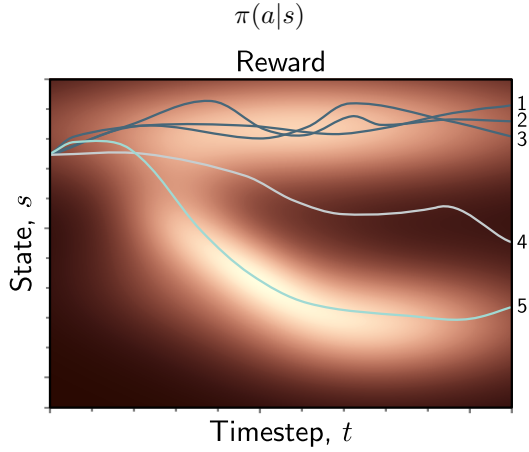
DQN Architecture



Double Q-Learning and Double Deep Q-Networks

$$\pi(a|s)$$

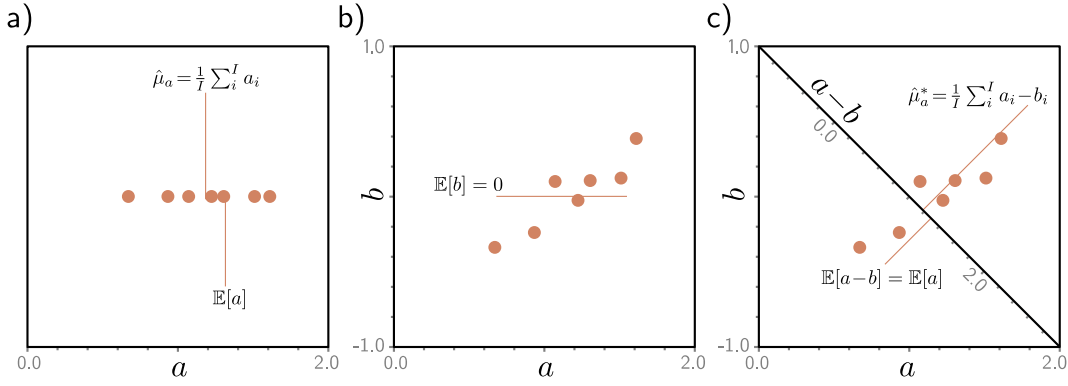
Policy Gradient Methods



REINFORCE Algorithm

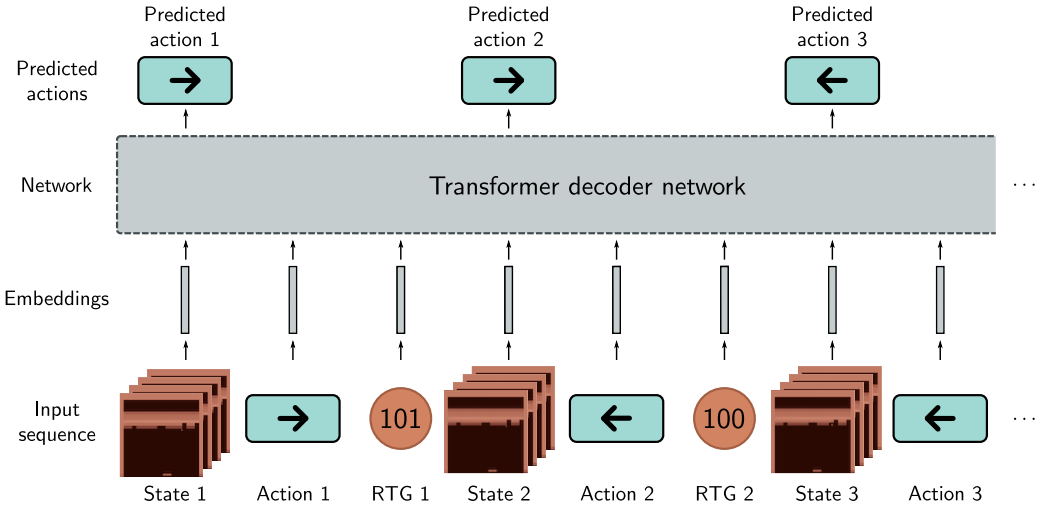
$$\pi(a|s)$$

Baselines


a)
$\hat{\mu}_a = \frac{1}{I}\sum_i^I a_i$
$\mathbb{E}[a]$
0.0
2.0
a
b)
1.0
-1.0
b
$\mathbb{E}[b] = 0$
0.0
2.0
a
c)
1.0
-1.0
b
$a-b$
0.0
$\hat{\mu}_a^* = \frac{1}{I}\sum_i^I a_i - b_i$
$\mathbb{E}[a-b] = \mathbb{E}[a]$
2.0
0.0
2.0
a

Actor-Critic Methods

Offline Reinforcement Learning



Closing Thoughts

Boom!