Workshop

Reinforcement Learning

howest
MULTIMEDIA & CREATIVE TECHNOLOGIES

Overview

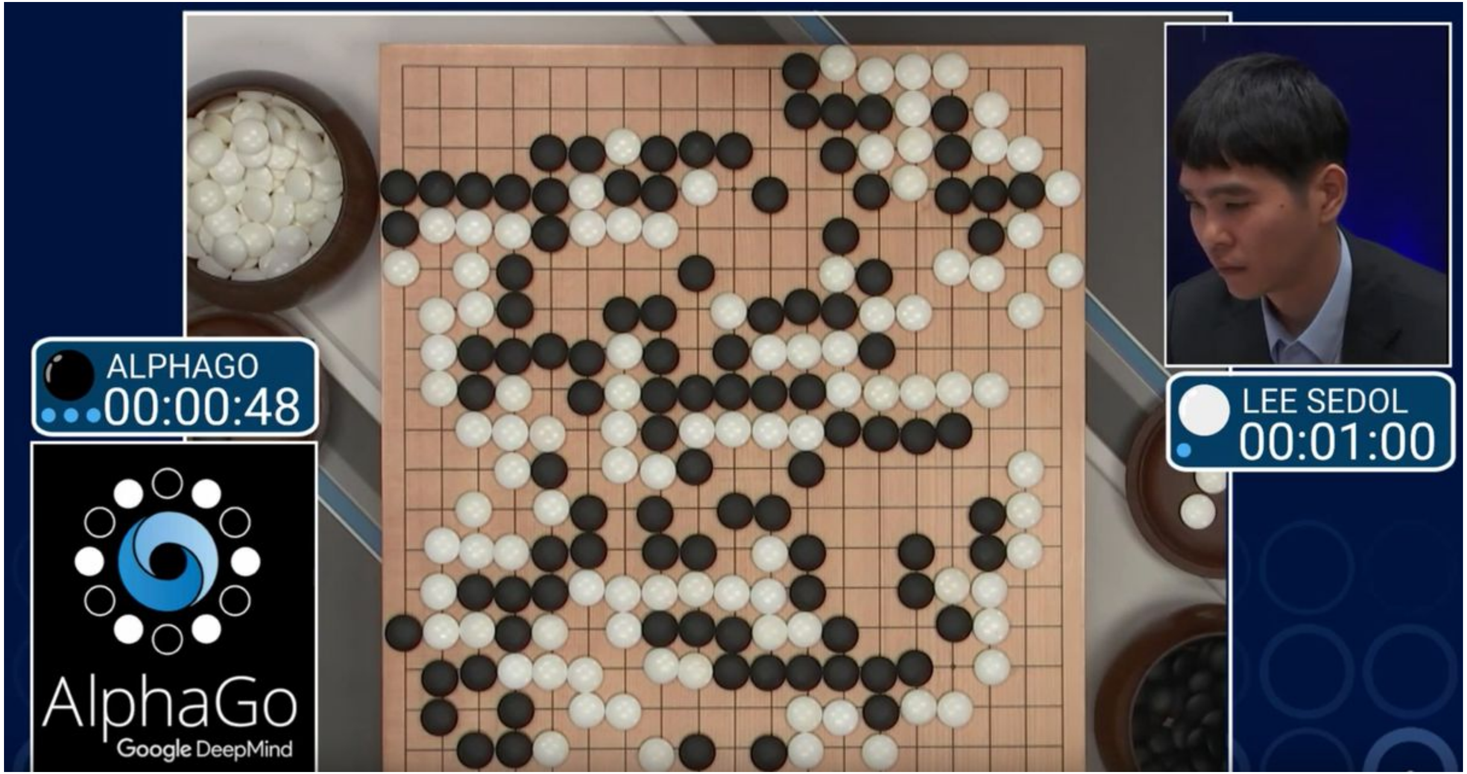
Introduction to RL

Reinforcement learning in context

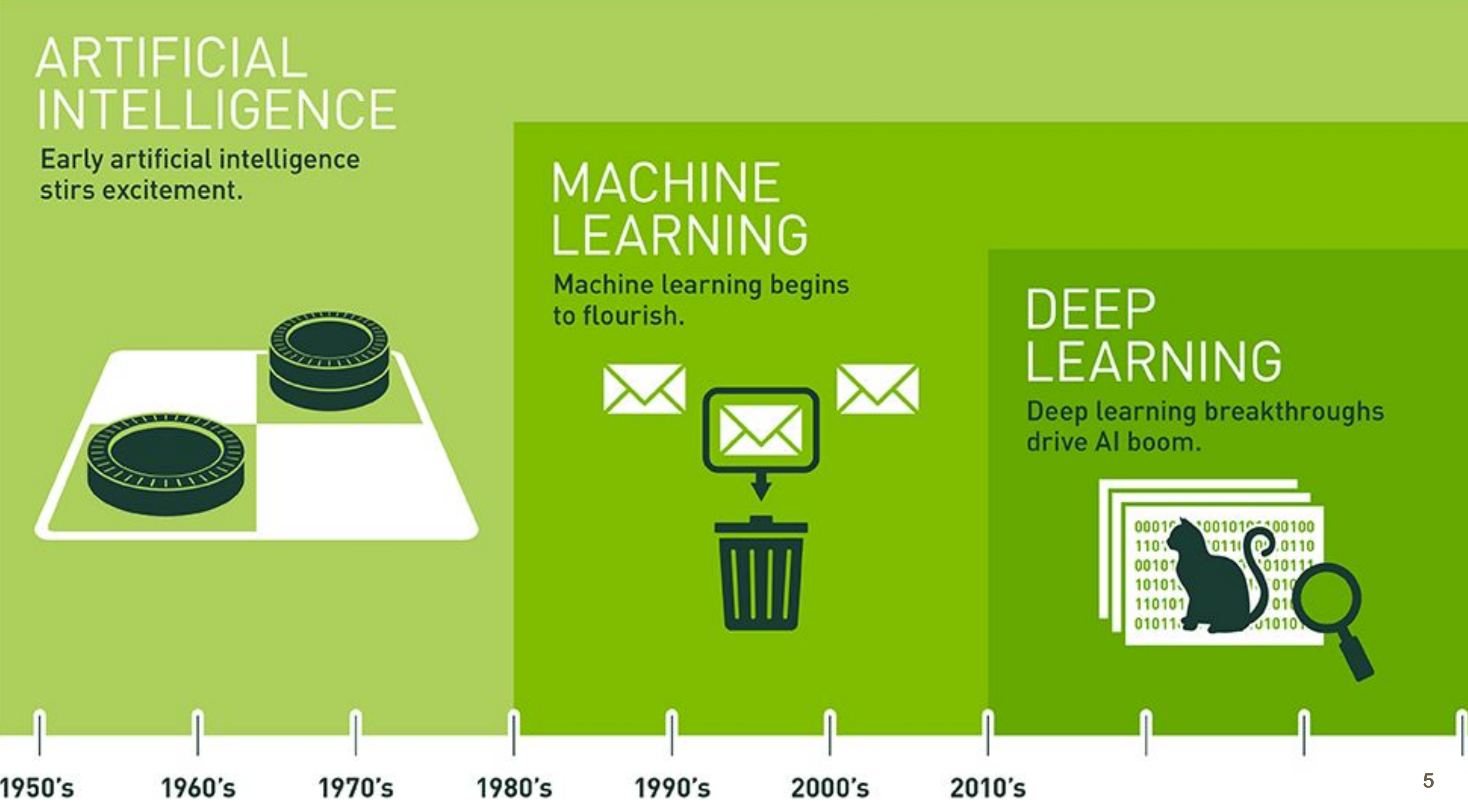
Practical implementation: Q-learning

Reinforcement learning in context

Examples



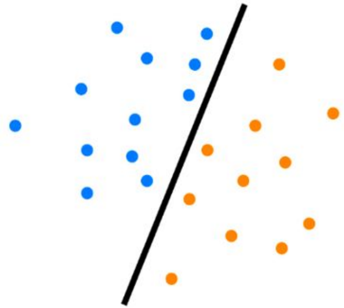
https://deepmind.com/blog/article/alphago-zero-starting-scratch


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Overview learning algorithms

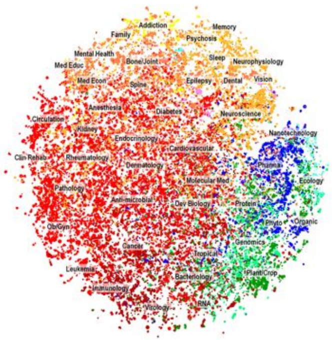
Supervised

Inputs with corresponding labels
Answers are provided
Task driven



Unsupervised

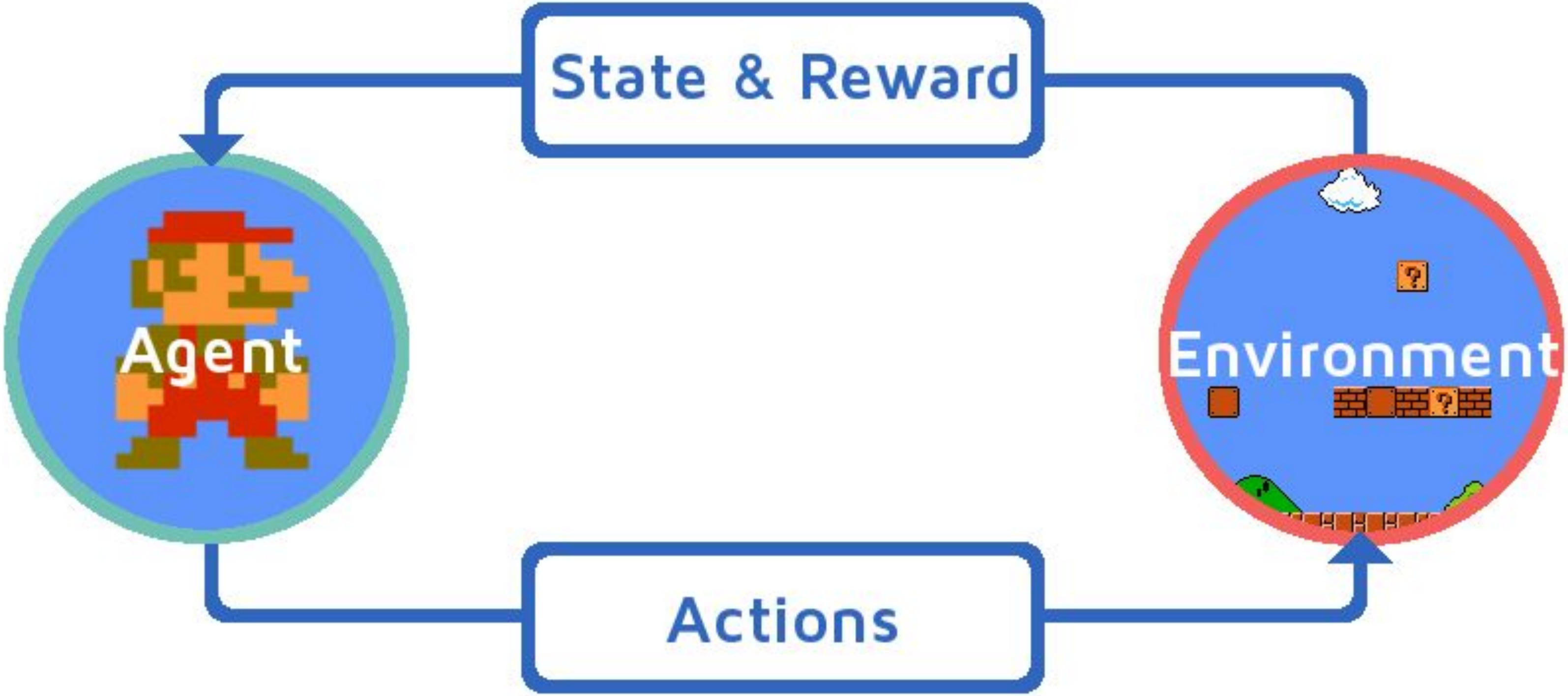
Corresponding labels are not provided
Data driven (clustering)



Reinforcement

Take the best actions in an environment to maximize rewards

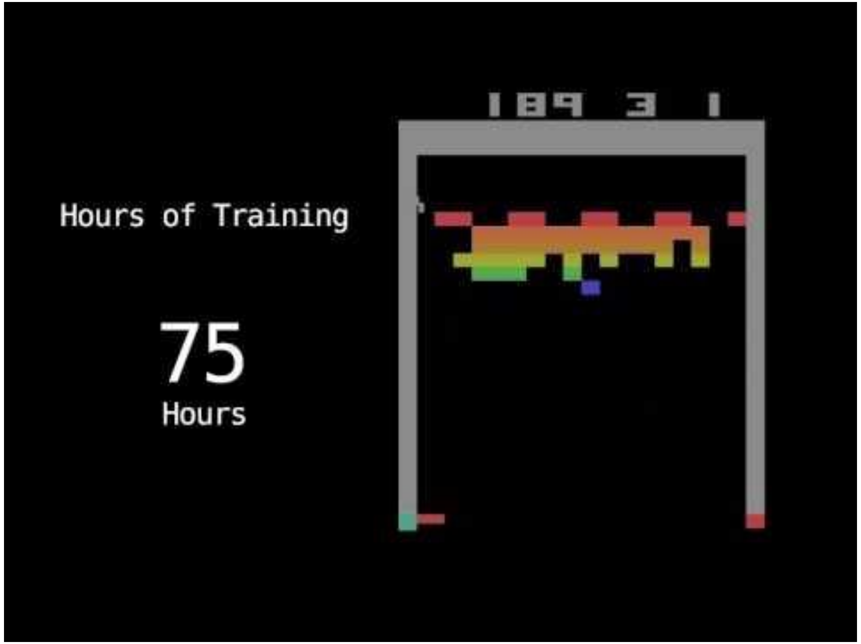
What is reinforcement learning?



What is reinforcement learning?

Examples

https://www.youtube.com/watch?v=4MlZncshy1Q

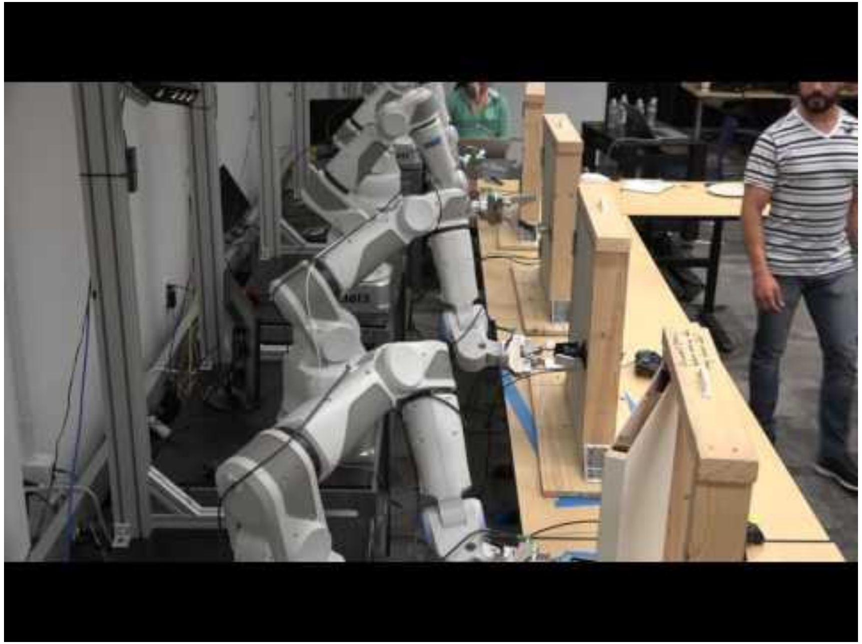


https://www.youtube.com/watch?v=eG1Ed8PTJ18


TURBO
SCORE
10500
LAPS
-/3
TIME
0:23
TURBO READY
MORE GAMES

Examples

https://www.youtube.com/watch?v=W_gxLKSsSIE



https://www.youtube.com/watch?v=ZBFwe1gF0FU

Examples

https://www.youtube.com/watch?v=VCdxqn0fcnE

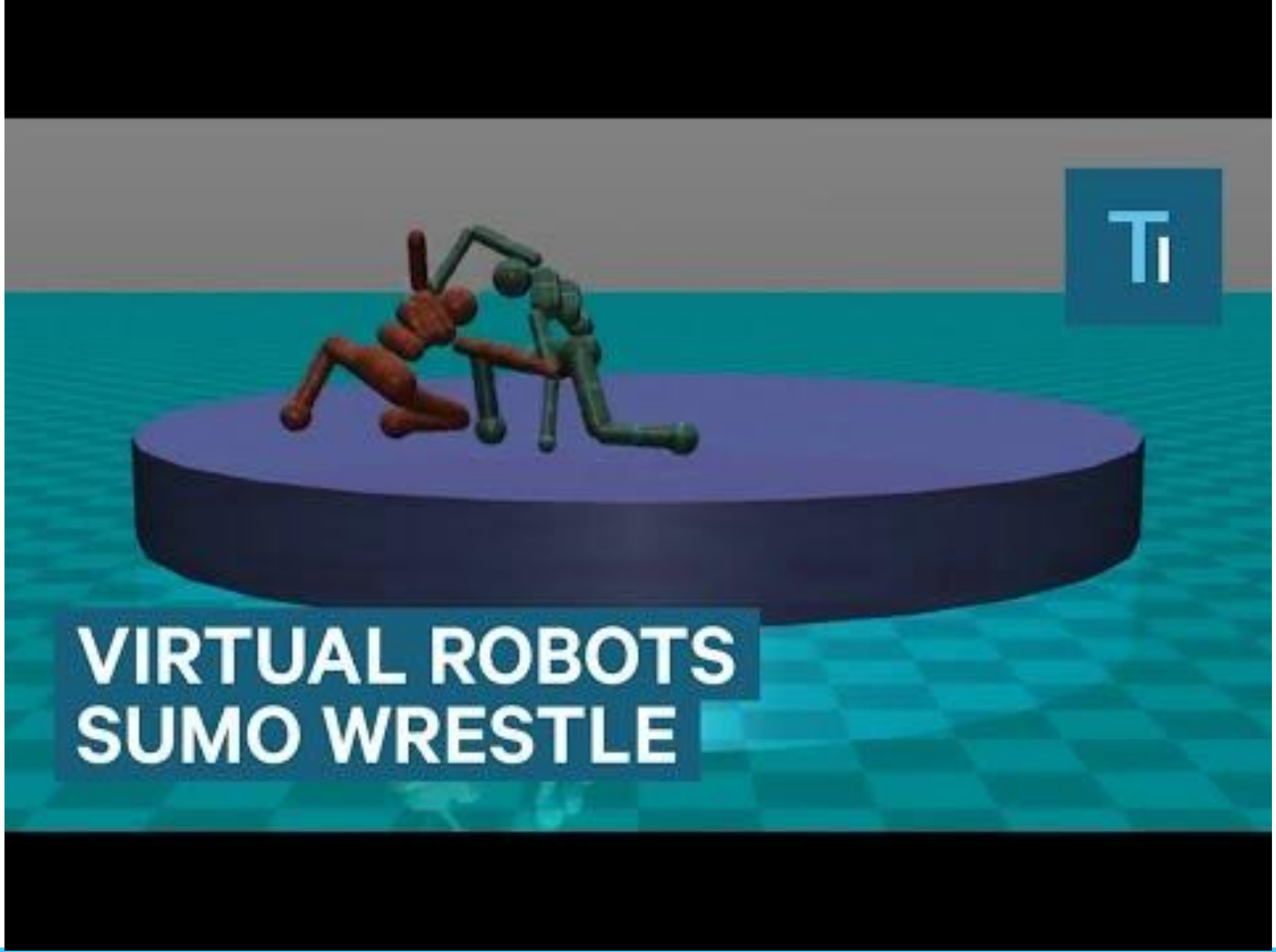
https://www.youtube.com/watch?v=opsmd5yuBF0

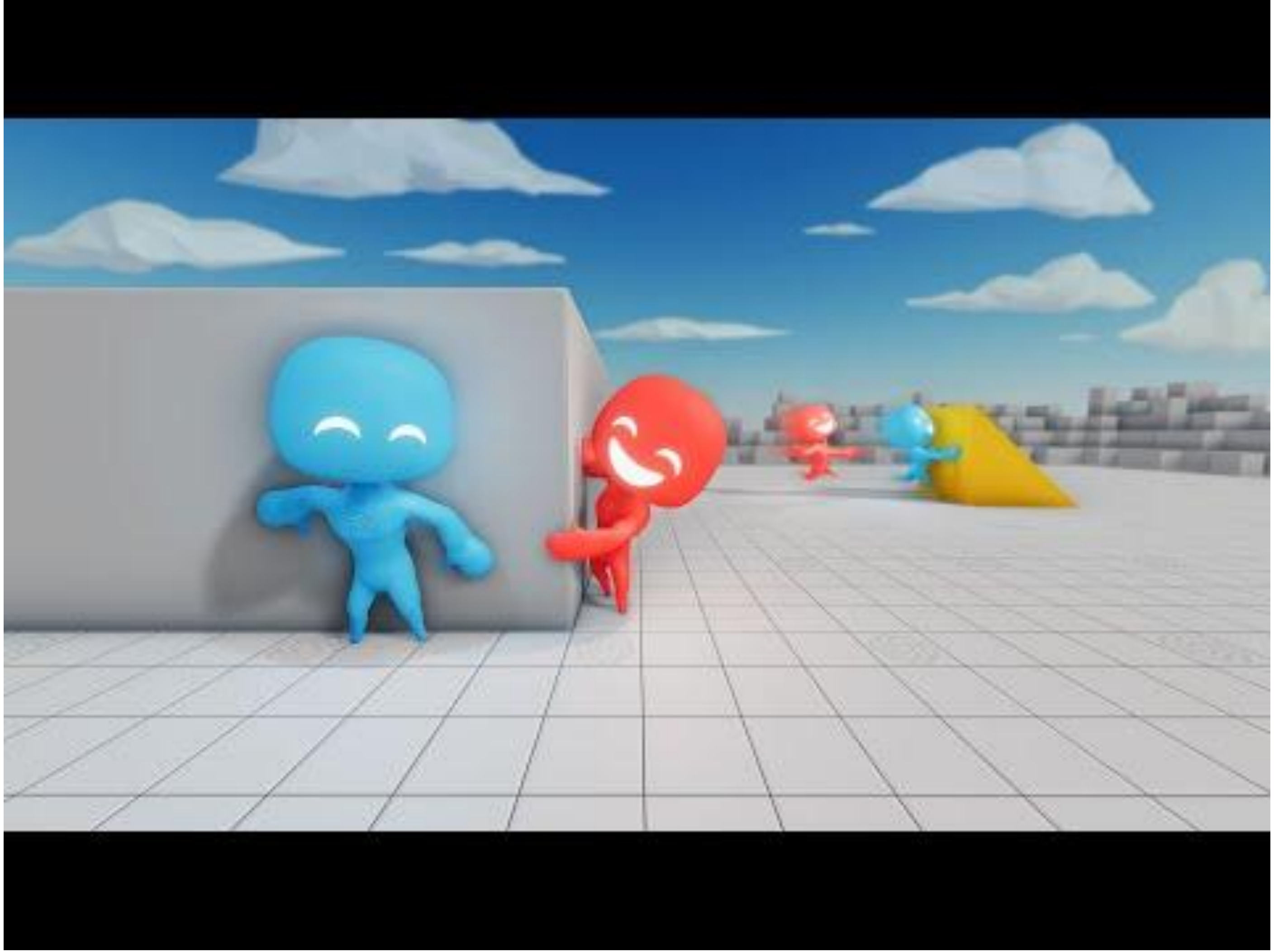

High PUE
ML Control On
ML Control Off
Low PUE

Reinforcement learning in humans

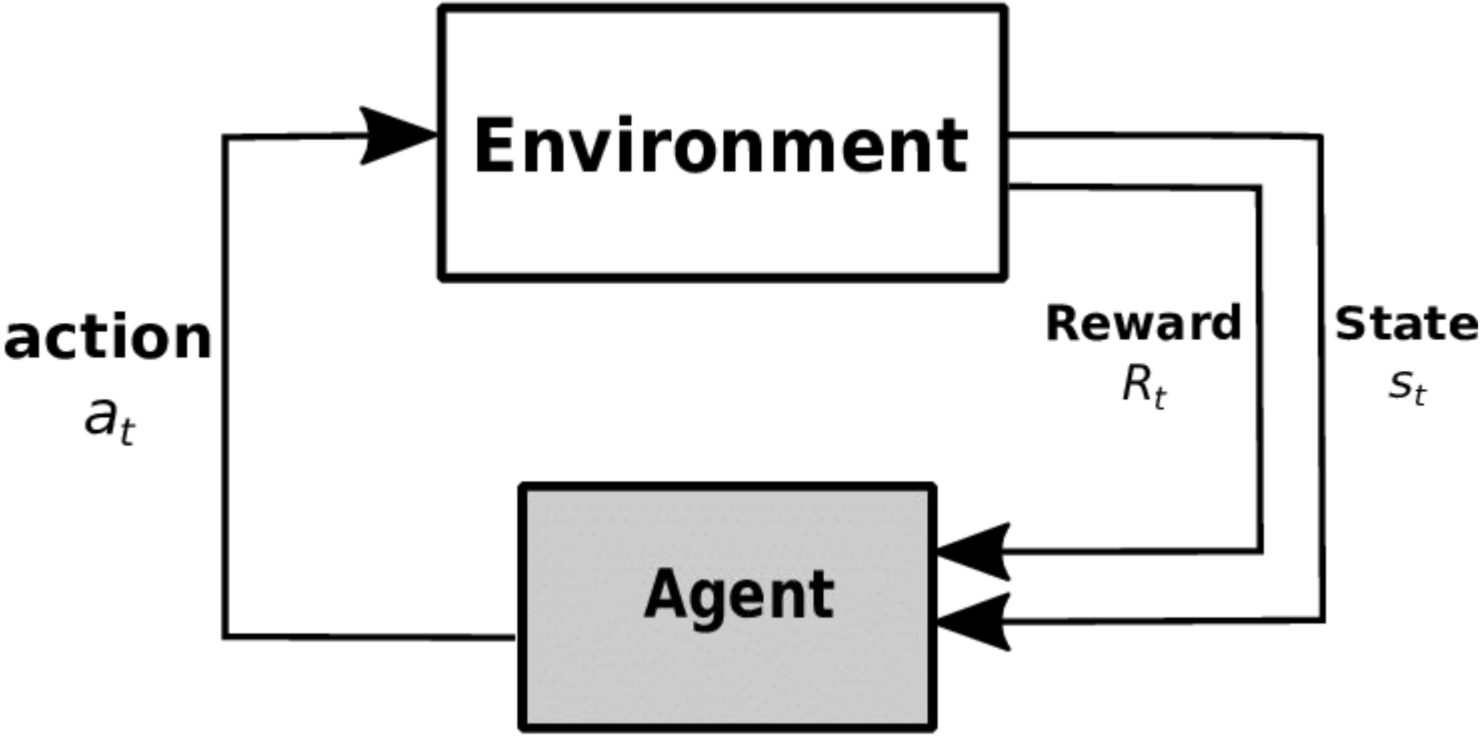
Examples

https://www.youtube.com/watch?v=gn4nRCC9TwQ


VIRTUAL ROBOTS
SUMO WRESTLE



Reinforcement learning terminologies



Reinforcement learning taxonomy

Value-based

- **SARSA
- **DQN: Deep Q-Networks
- **Double DQN
- **DQN + Prioritized Experience Replay
- QT-OPT

Policy-based

- **REINFORCE

Model-based

- iLQR: Iterative Linear Quadratic Regulator
- MPC: Model Predictive Control
- MCTS: Monte Carlo Tree Search

Combined methods
Value and policy

- **Actor-Critic
  - A2C[1], GAE[2], A3C[3]
- TRPO: Trust Region Policy Optimization
- **PPO: Proximal Policy Optimization
- SAC: Soft Actor-Critic

Combined methods
Model + value and / or policy

- Dyna-Q / Dyna-AC
- AlphaZero
- I2A: Imagination Augmented Agents
- VPN: Value Prediction Networks

**: discussed in this book
1. A2C: Advantage Actor-Critic
2. A3C: Asynchronous Advantage Actor-Critic
3. GAE: Actor-Critic with Generalized Advantage Estimation

Q-learning

Q-learning

What is Q-learning?

- The objective of Q-learning is to find a policy that is optimal in the sense that the expected value of the total reward over all successive steps is **the maximum achievable**.

- The goal of Q-learning is to **find the optimal policy by learning the optimal Q-values** for each **state-action pair**.

- The Q-learning algorithm iteratively updates the Q-values for each state-action pair using the Bellman equation until the Q-function converges to the optimal Q-function, q*. This approach is called **value iteration**.

- Q-learning **converges to optimal Q-values** if all states are visited by the agent for an infinite amount of times.

- Q-learning is **off-policy**

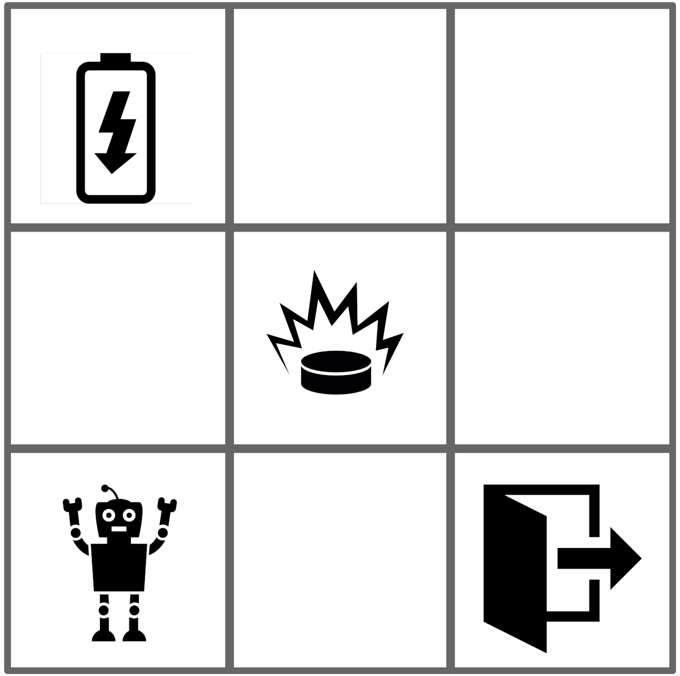
Q-learning

Updating the Q-values

$$\boxed{Q(s_t, a_t)} \leftarrow (1-\alpha) \cdot \boxed{\underbrace{Q(s_t, a_t)}_{\text{old value}}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Q-learning

Example



The goal for the robot (agent) is to **find the exit**:

1 step = -10 points.

Charging = + 10 points.

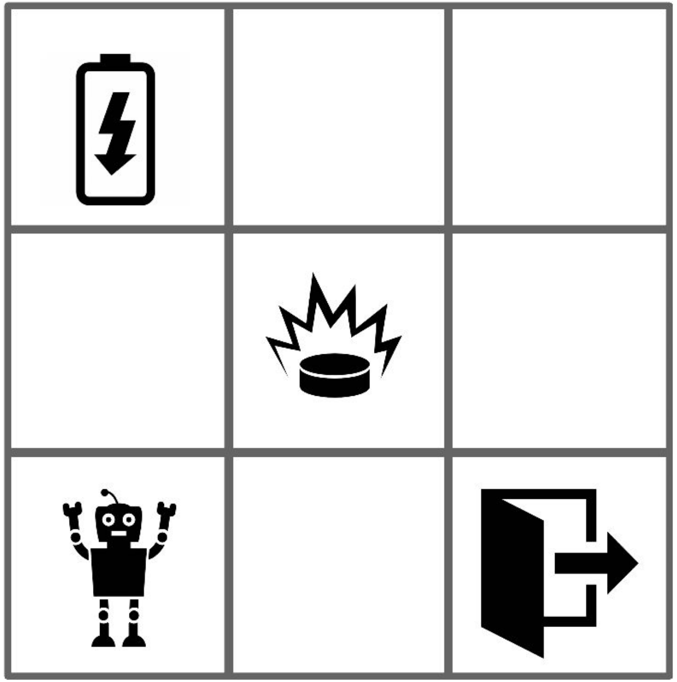
Reaching the exit = +100 points and episode ends.

Stepping on a landmine = -100 points and episode ends.

For the purpose of the example the robot will only explore the environment (= only taking random actions) and not yet exploit it's knowledge of the environment.

Q-learning

Q-table

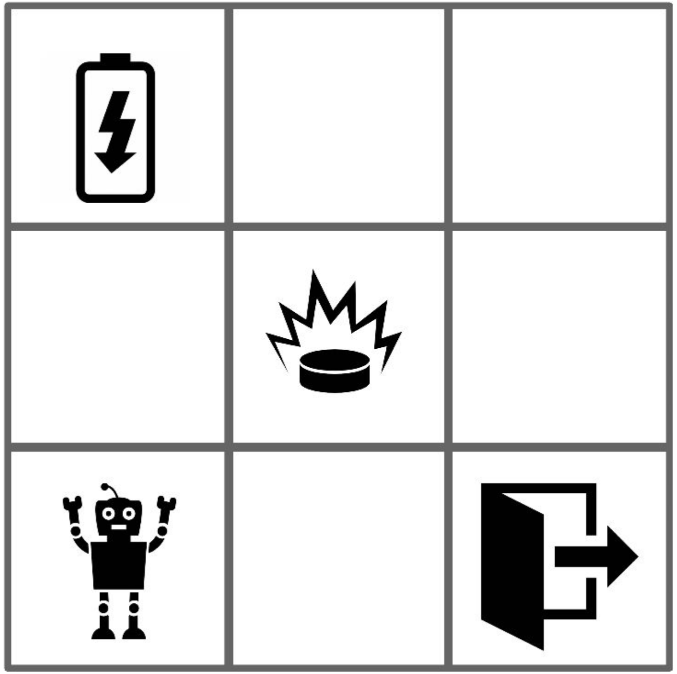


Q-table

	Left	Right	Up	Down
Charging	X		X	
Empty cel 1			X	
Empty cel 2		X	X	
Empty cel 3	X			
Land mine	X	X	X	X
Empty cel 4		X		
Start	X			X
Empty cel 5				X
Exit	X	X	X	X

Q-learning

Q-table initialization (with zeros)

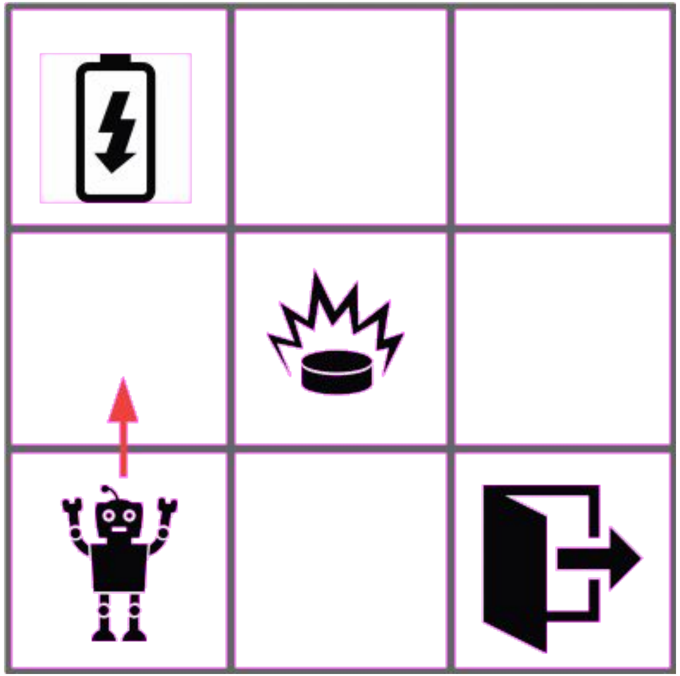


Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	0	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	0	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action



Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	0	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	0	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Robot takes random action ‘up’

From ‘starting state’ to state ‘empty cell 3’

Updating the Q-values with α=0.7 and γ=0.8:

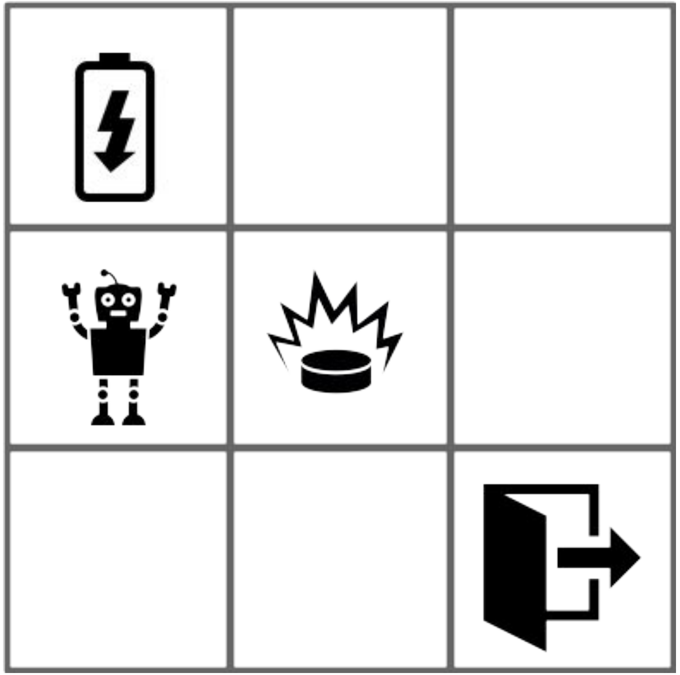
$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_Q(s_{t+1}, a))$$

$$= (1-0.7) \cdot 0 + 0.7 \cdot (-10 + 0.8 \cdot 0)$$

$$= -7$$

Q-learning

Updating the Q-table

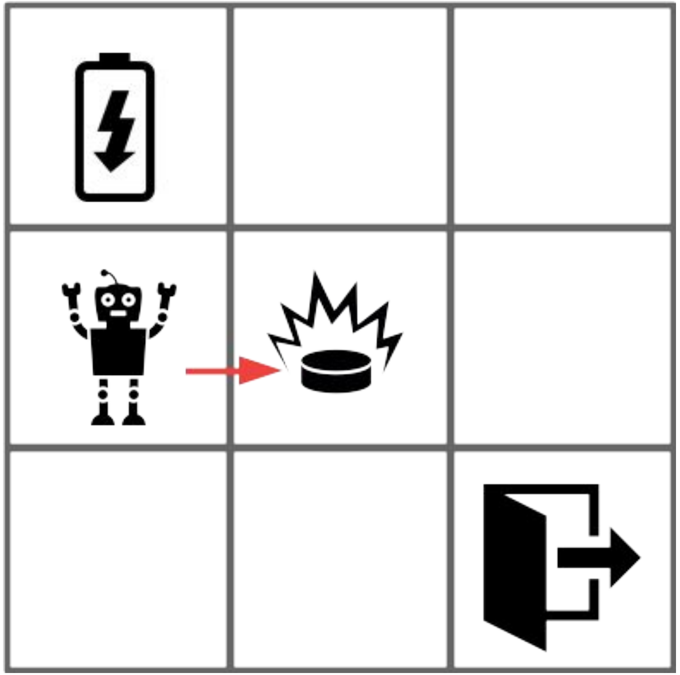


Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	0	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-7	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action



Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	0	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-7	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Robot takes random action 'right'

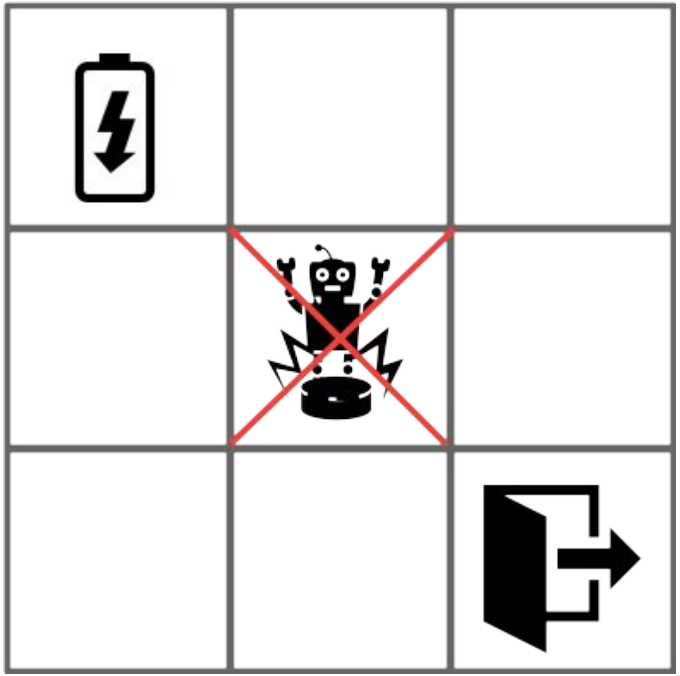
From state 'empty cell 3' to state 'landmine'

Updating the Q-values with α=0.7 and γ=0.8:

$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_Q(s_{t+1}, a))$$
$$= (1-0.7) \cdot 0 + 0.7 \cdot ((-10-100) + 0.8 \cdot 0)$$
$$= -77$$

Q-learning

Updating the Q-table



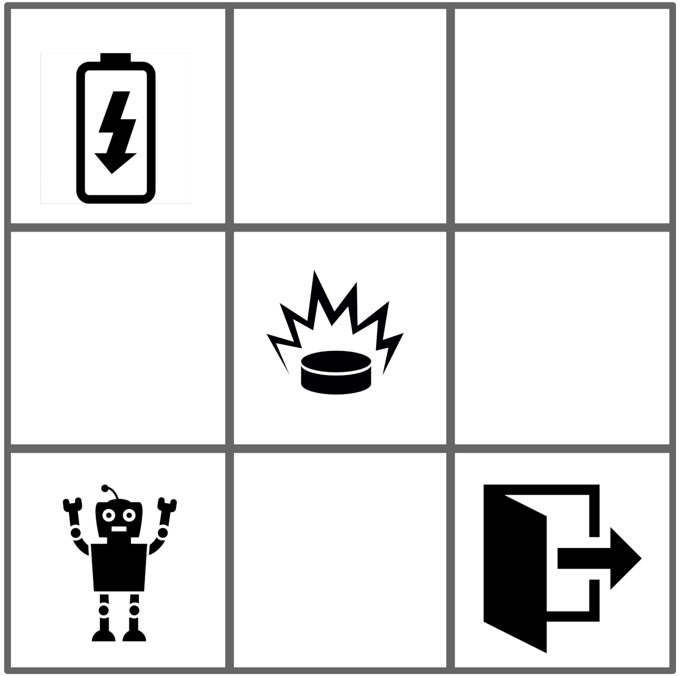
END OF THE EPISODE

Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-7	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Start new episode

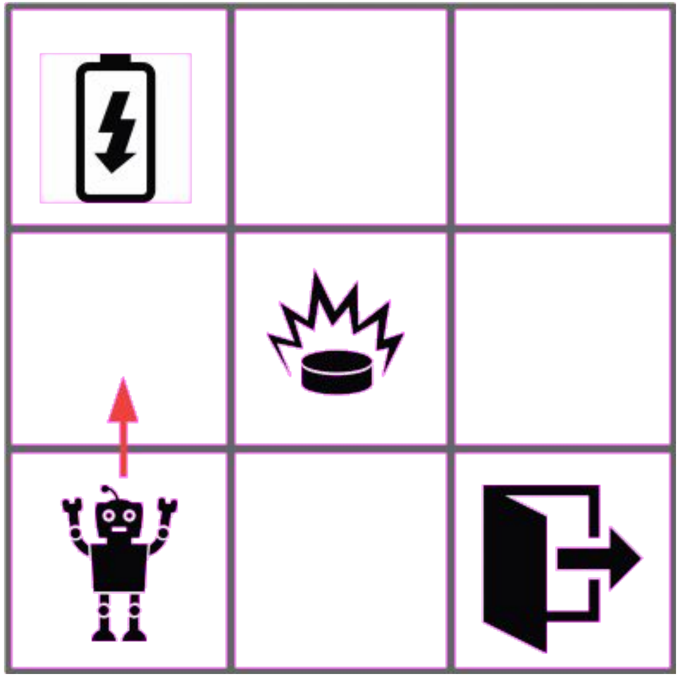


Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-7	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action



Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-7	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Robot takes random action ‘up’

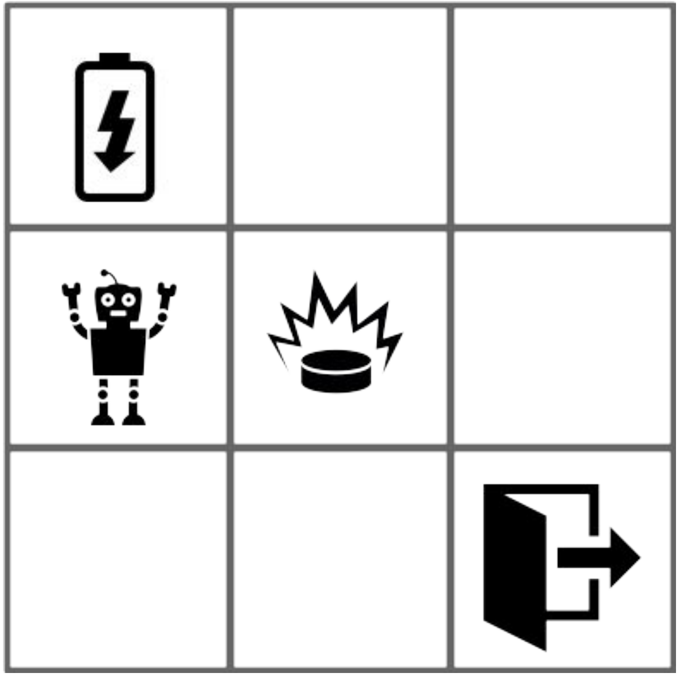
From ‘starting state’ to state ‘empty cell 3’

Updating the Q-values with α=0.7 and γ=0.8:

$$
\begin{aligned}
Q(s_t, a_t) &= (1-\alpha)\cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_Q(s_{t+1}, a)) \\
&= (1-0.7)\cdot(-7) + 0.7\cdot(-10 + 0.8\cdot \max(-77; 0; 0)) \\
&= 0.3\cdot(-7) + 0.7\cdot(-10 + 0.8\cdot 0) \\
&= -9.1
\end{aligned}
$$

Q-learning

Updating the Q-table



Q-table

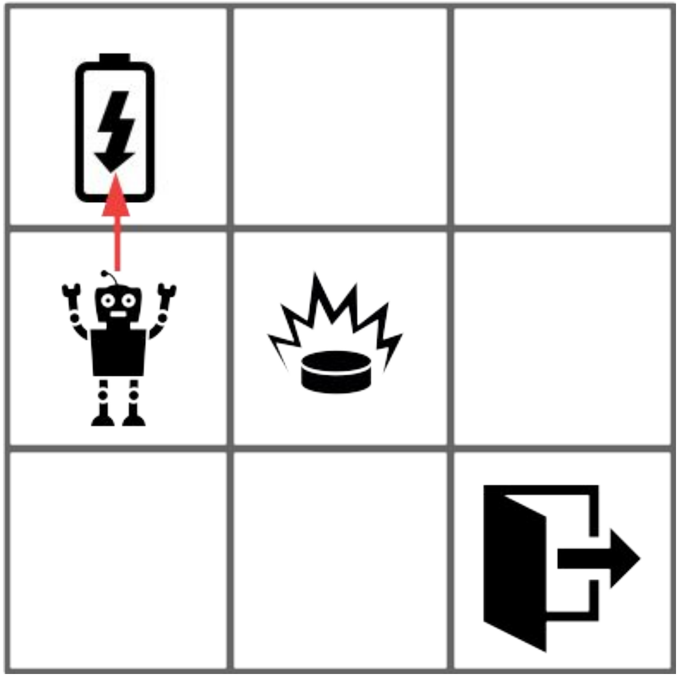
	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action

Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	0	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X



Robot takes random action 'up'

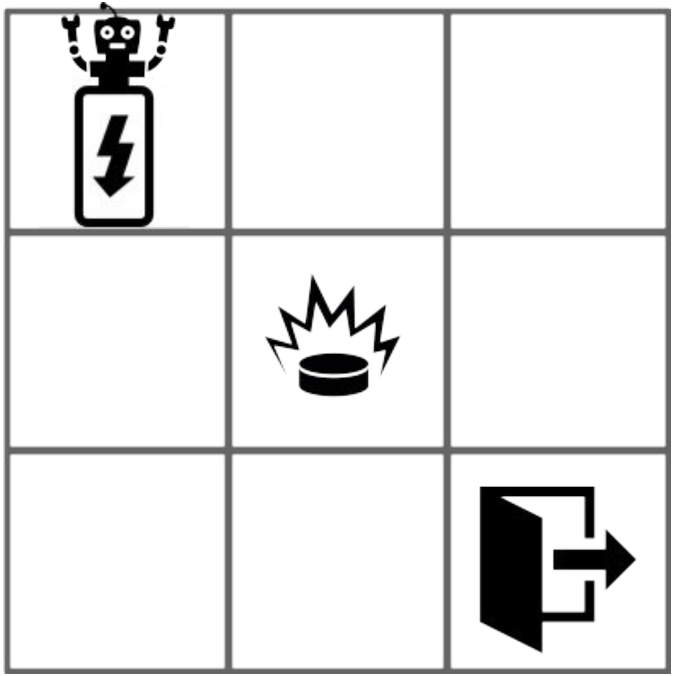
From 'empty cell 3' to state 'charging'

Updating the Q-values with α=0.7 and γ=0.8:

$$
\begin{aligned}
Q(s_t, a_t) &= (1-\alpha)\cdot Q(s_t,a_t) + \alpha\cdot\left(r_t + \gamma\cdot \max_{Q}(s_{t+1}, a)\right) \\
&= (1-0.7)\cdot 0 + 0.7\cdot(-10+10+0.8\cdot\max(0;0)) \\
&= 0.3\cdot(-7) + 0.7\cdot(0+0.8\cdot 0) \\
&= -2.1
\end{aligned}
$$

Q-learning

Updating the Q-table

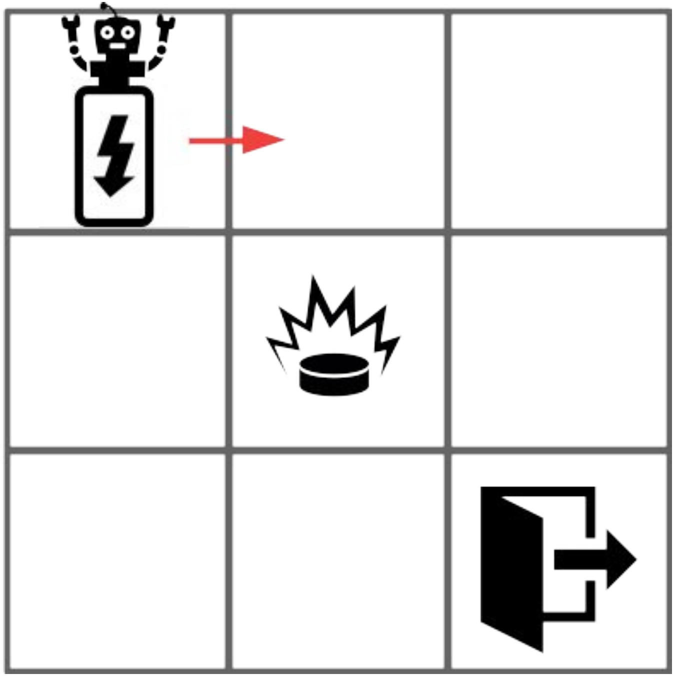


Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	-2.1	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action



Robot takes random action ‘right’

From ‘charing’ to state ‘empty cell 1’

Updating the Q-values with α=0.7 and γ=0.8:

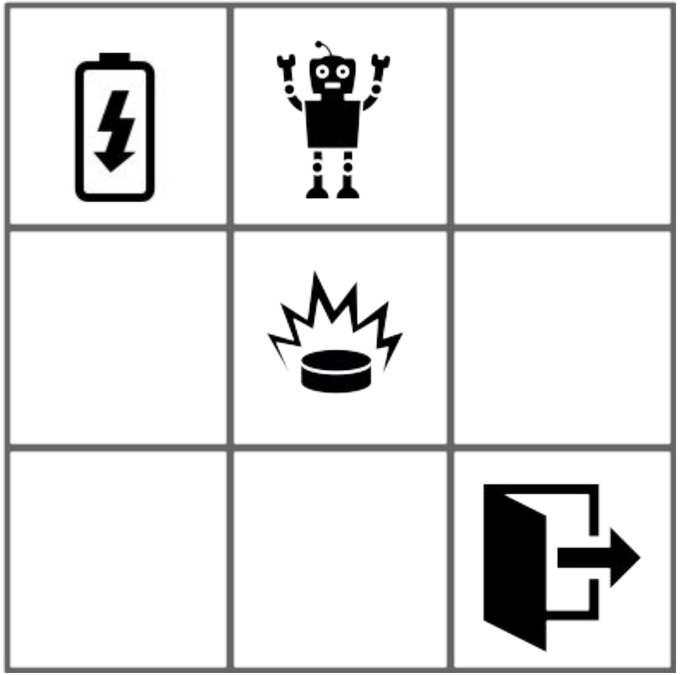
$$
\begin{aligned}
Q(s_t, a_t) &= (1-\alpha)\cdot Q(s_t, a_t) + \alpha \cdot \left(r_t + \gamma \cdot \max_Q(s_{t+1}, a)\right) \\
&= (1-0.7)\cdot 0 + 0.7 \cdot (-10 + 0.8 \cdot \max(0;0;0)) \\
&= 0 + 0.7 \cdot (-10 + 0.8 \cdot 0) \\
&= -7
\end{aligned}
$$

Q-table

	Left	Right	Up	Down
Charging	X	0	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	-2.1	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Updating the Q-table



Q-table

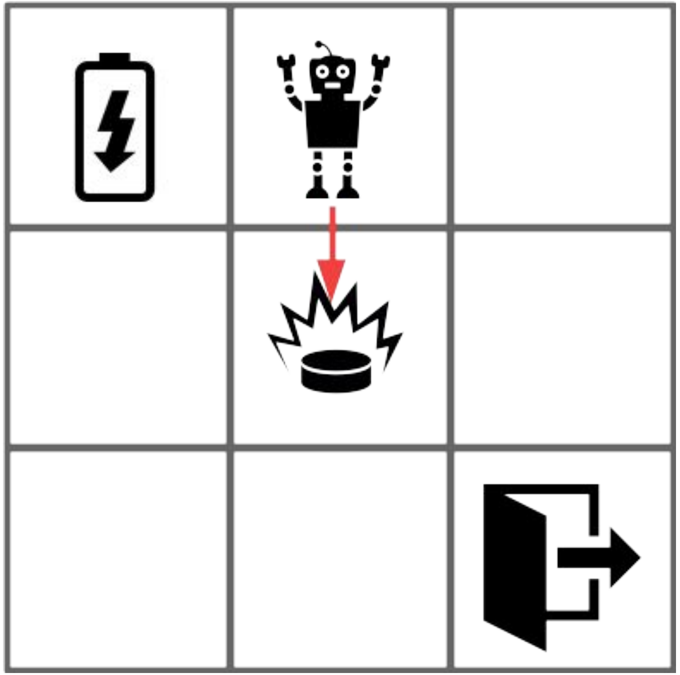
	Left	Right	Up	Down
Charging	X	-7	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	-2.1	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Agent is taking a random action

Q-table

	Left	Right	Up	Down
Charging	X	-7	X	0
Empty cel 1	0	0	X	0
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	-2.1	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X



Robot takes random action ‘Down’

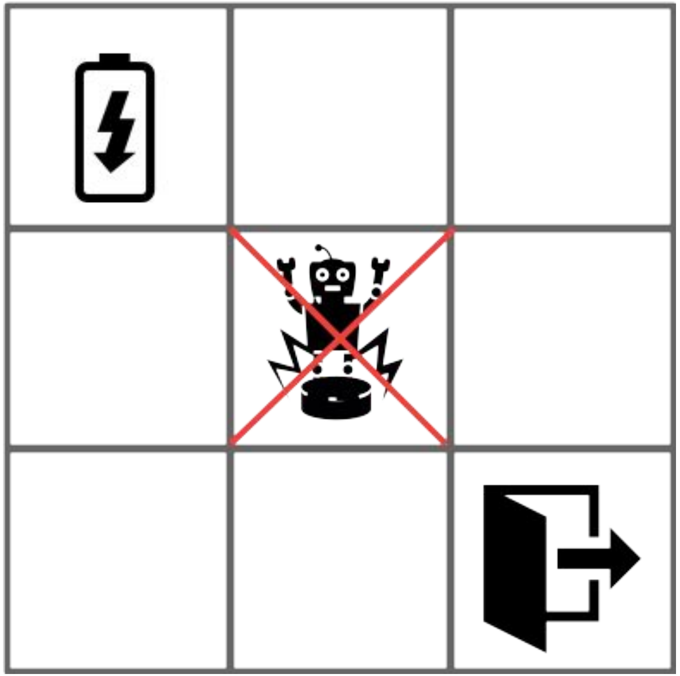
From ‘empty cell 1’ to state ‘landmine’

Updating the Q-values with α=0.7 and γ=0.8:

$$
\begin{aligned}
Q(s_t, a_t) &= (1-\alpha)\cdot Q(s_t, a_t) + \alpha \cdot (r_t + \gamma \cdot \max_Q(s_{t+1}, a)) \\
&= (1-0.7)\cdot 0 + 0.7 \cdot (-10 - 100 + 0.8 \cdot 0) \\
&= 0 + 0.7 \cdot (-110) \\
&= -77
\end{aligned}
$$

Q-learning

Updating the Q-table



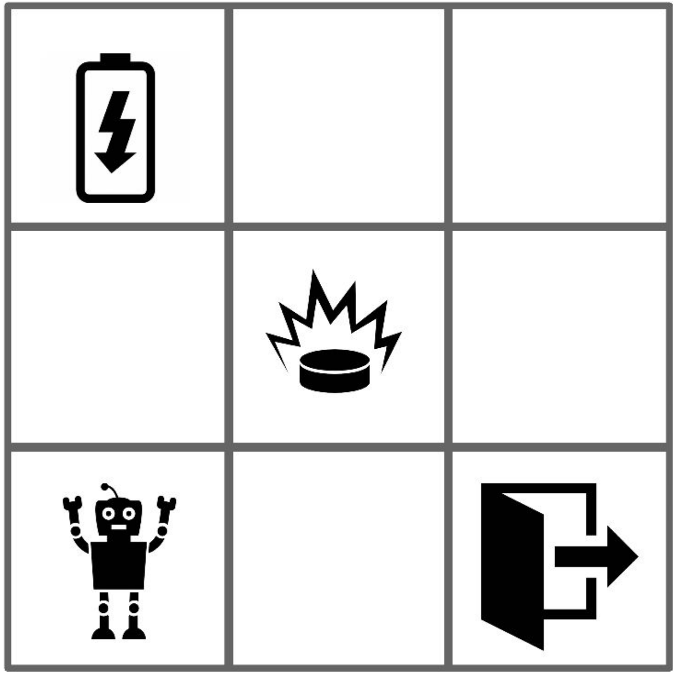
END OF THE EPISODE

Q-table

	Left	Right	Up	Down
Charging	X	-7	X	0
Empty cel 1	0	0	X	-77
Empty cel 2	0	X	X	0
Empty cel 3	X	-77	-2.1	0
Land mine	X	X	X	X
Empty cel 4	0	X	0	0
Start	X	0	-9.1	X
Empty cel 5	0	0	0	X
Exit	X	X	X	X

Q-learning

Suppose after many episode we end up with the following Q-table



Q-table

	Left	Right	Up	Down
Charging	X	1.4	X	-6.4
Empty cel 1	-0.4	8.0	X	-86.7
Empty cel 2	-4.2	X	X	16.3
Empty cel 3	X	-86.4	7.1	-0.48
Land mine	X	X	X	X
Empty cel 4	-82.9	X	-8.7	78.4
Start	X	32.4	18.9	X
Empty cel 5	-8.4	89.1	-86.7	X
Exit	X	X	X	X

During exploitation the agent will follow the state-actions with the highest Q-values: Start -> Empty cell 5 -> Exit

SARSA

Cliff walking problem

- Q-learning will converge to the optimal path (but also more risky path)
- SARSA will converge to the safest path