DQN

’StarCraft II로 배우는 강화학습’ 웨비나

Sep 8, 2020

박석

Agenda

1. DQN
- Naive DQN
- Fixed Q Target
- Experience Replay
2. DQN variants
- Double DQN
- Prioritized Experience Replay -
Dueling DQN
- Rainbow

Model-based RL vs. Model-free RL

Model-based RL Model-free RL 특징

환경모델이 있음 환경모델이 없음

…

장점 • high ‘Sample Efficiency’ • high ‘Transferability’

• Useful under No Env. Model • Useful under Complex Task

단점 • high ‘Computing Cost’ • high ‘Model Error’

• Huge Training Data
• Hard under Multi Task/Same Env.

DP, Dyna-Q, Trajectory Sampling, RTDP, MBA, NVE, MBPO, GPS, iLQR,

SARSA, Q-learning, DQN, REINFORCE, PG, AC, PPO, DDPG, …

Value-based RL vs. Policy-based RL vs. Actor-Critic RL

Value-based RL Policy-based RL Actor-Critic RL

근사하여 Optimal Optimal Policy를 두가지의
Policy를 찾음. 찾음.
특징 Reward 함수를 Value-based,
Value 함수를 직접 근사하여 Policy based 장점을 모두 취함.

장점 • Low Variance • Low Bias 단점 • High

Bias • High Variance

DQN, DDQN, PER,
Dueling DQN,
Rainbow, R2D2, …
Hill Climbing,

REINFORCE,
PG, TRPO,
PPO, …

AC, A3C, A2C, GAE,
DDPG, SAC …

DRL – Value-based

Methods • DQN - Experience Replay, Fixed Q-Targets

• DDQN / PER / Dueling DQN / Rainbow

Optional References

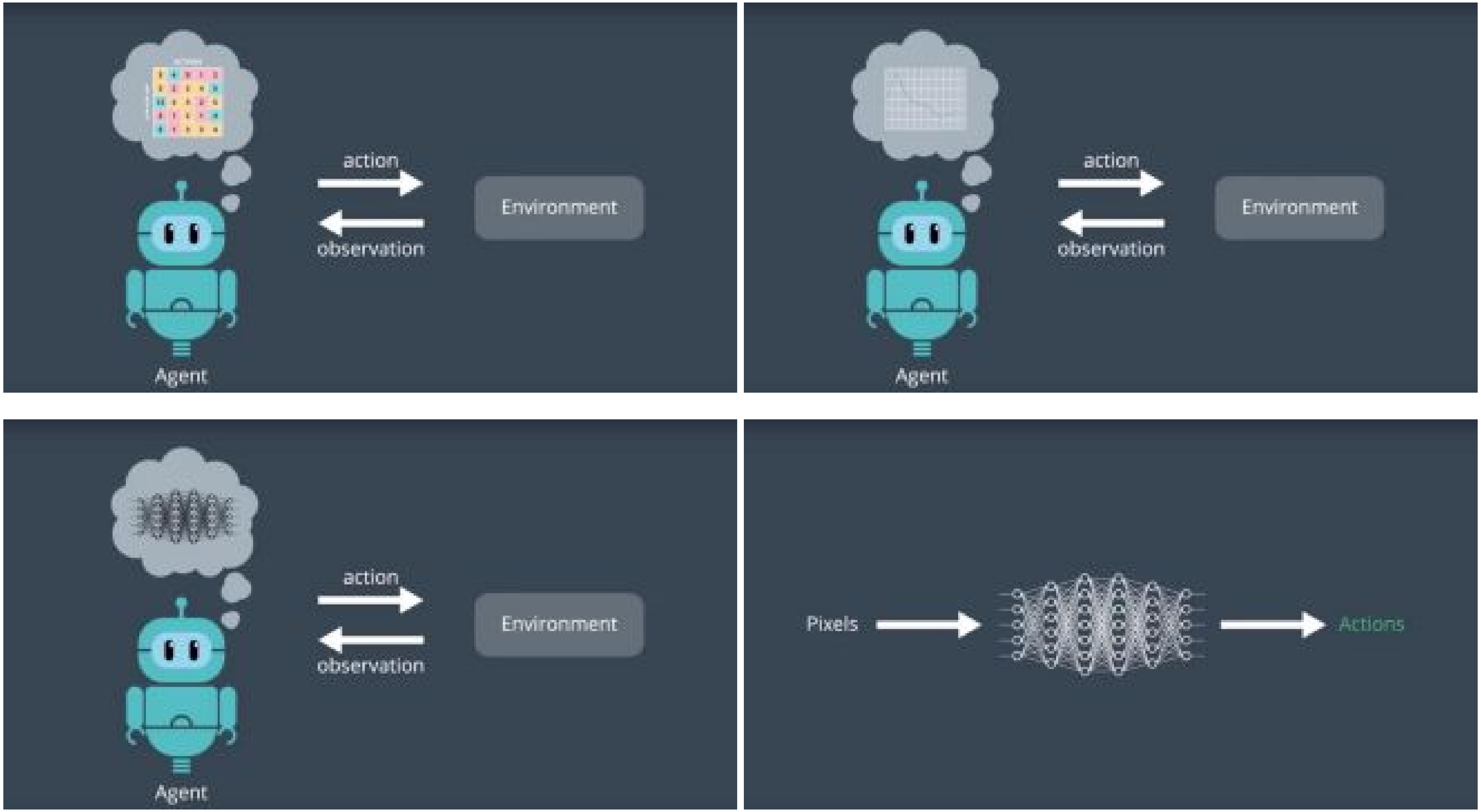
- Read this [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks.
- Read the [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.
- Learn more about Deep Q-Learning and Google DeepMind by watching this [video] { https://www.youtube.com/watch?v=xN1d3qHMIEQ }.

Deep RL for Robotics - Optional Resources

- Read [this article] { https://www.technologyreview.com/s/601045/this-factory-robot-learns-a-new-job overnight/ } if you'd like to learn more about how the Japanese robot company Fanuc uses deep RL to learn new tasks.
- [This robot] { https://www.cnet.com/news/robot-learns-via-trial-and-error-like-a-human/ } at UC Berkeley also uses deep RL to learn new skills.
- Learn how [Amazon is using deep RL] { https://medium.com/@teamrework/deep-learning-in-production warehousing-with-amazon-robotics-571e69fea721 } to make their warehouses more efficient.

Main Reference – Udacity DRL nanodegree contents (https://www.udacity.com/course/deep-reinforcement-learning-nanodegree--nd893) [Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto - Second Edition] http://incompleteideas.net/book/the-book.html

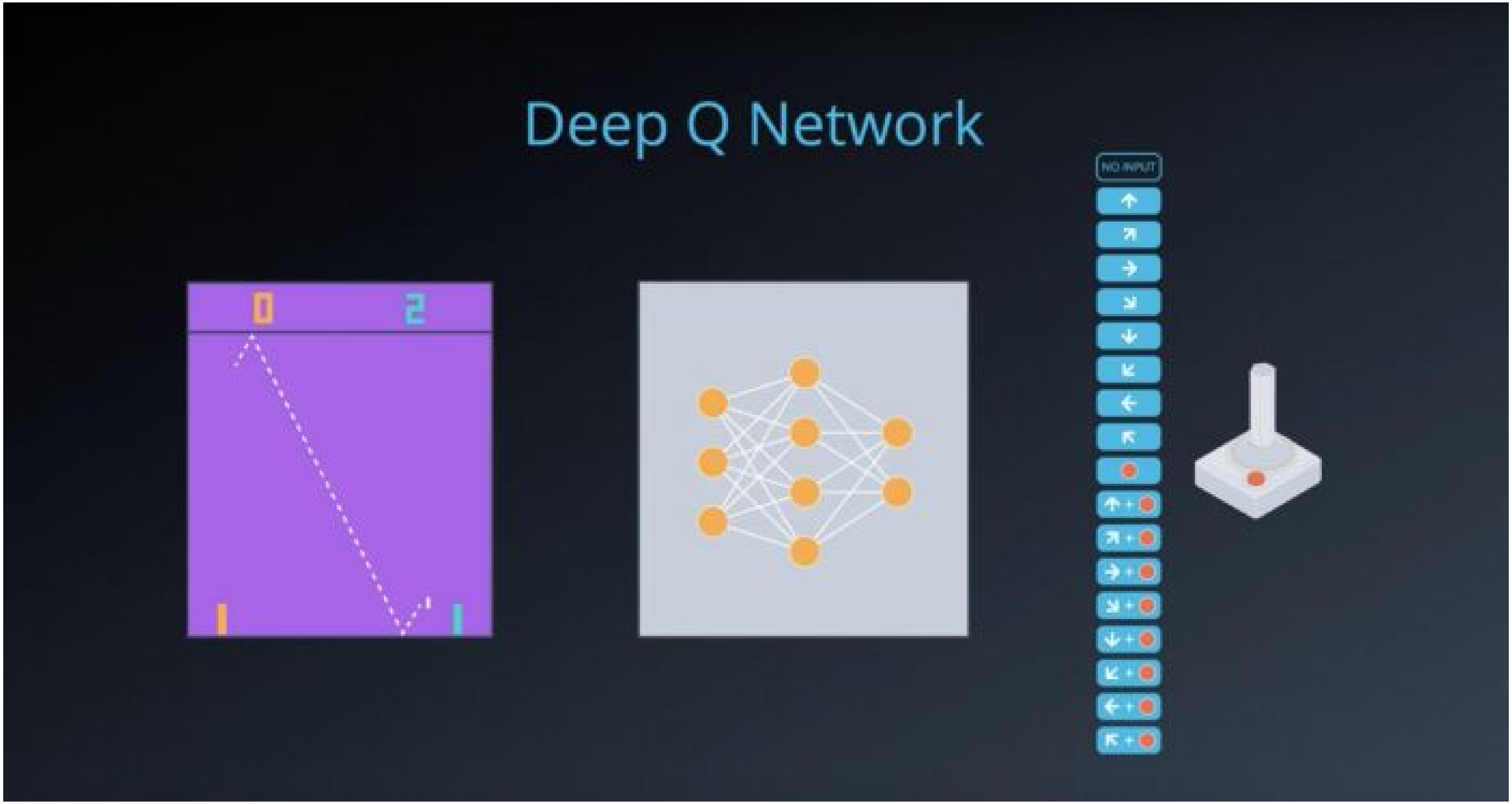
From RL to Deep RL



Main Reference – Udacity DRL nanodegree contents (https://www.udacity.com/course/deep-reinforcement-learning-nanodegree--nd893) [Reinforcement

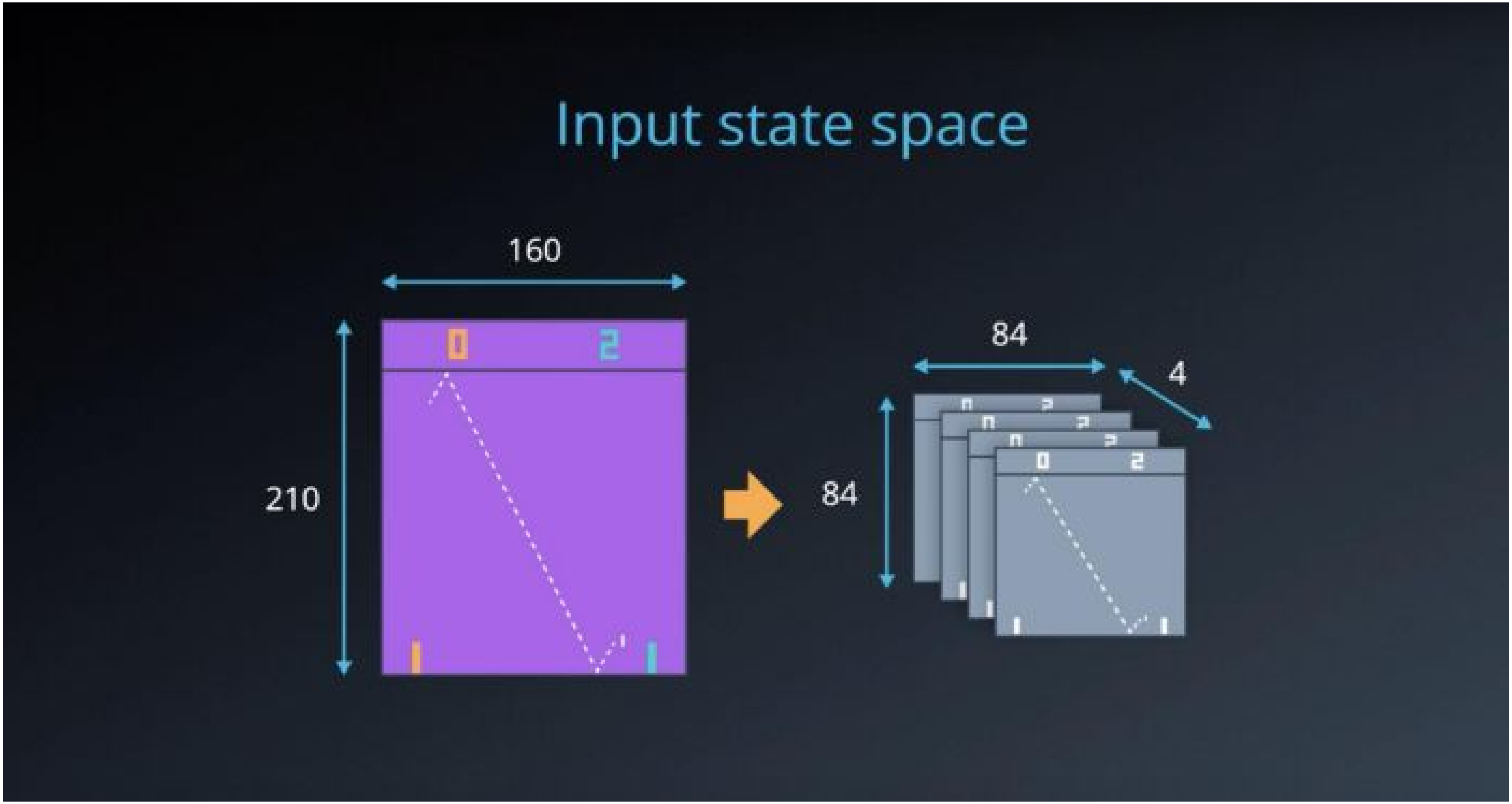
Learning: An Introduction by Richard S. Sutton and Andrew G. Barto - Second Edition] http://incompleteideas.net/book/the-book.html

Deep Q Networks



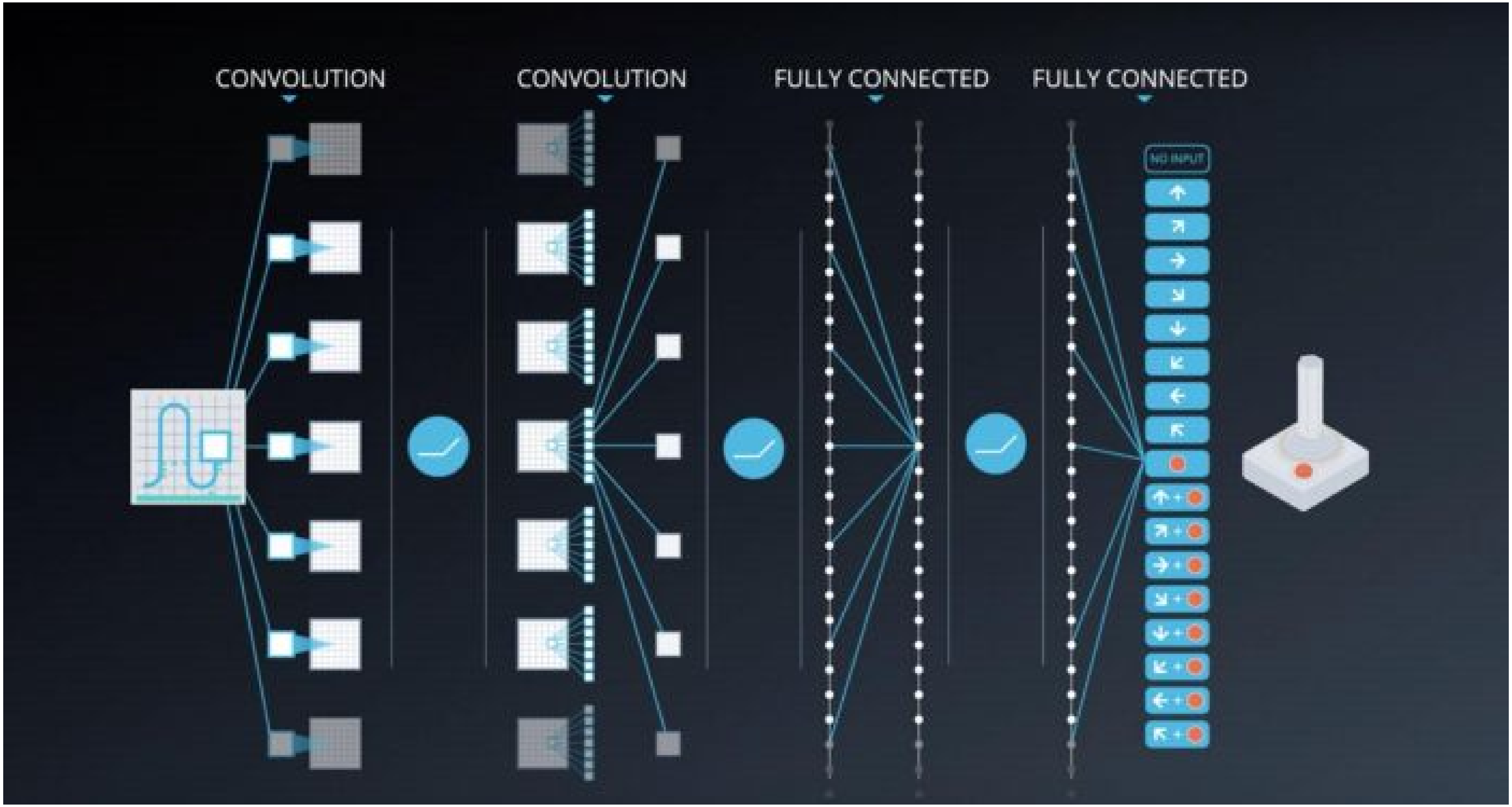
Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Deep Q Networks



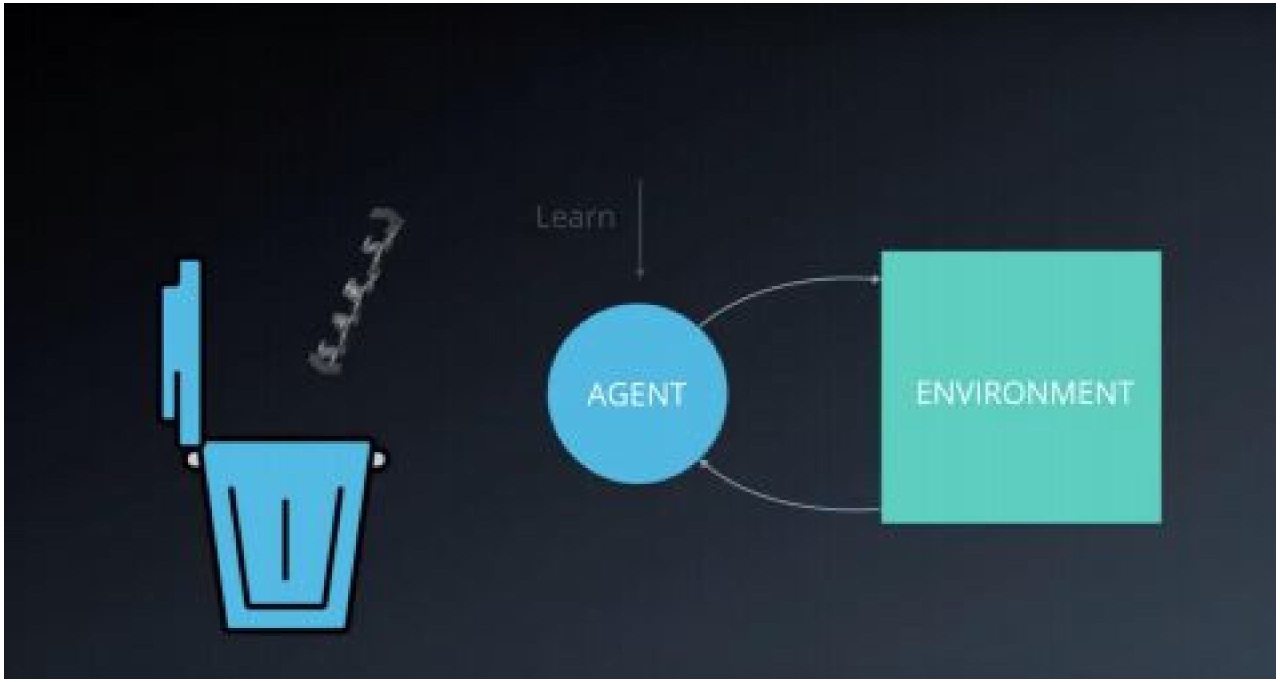
8 Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Deep Q Networks



Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

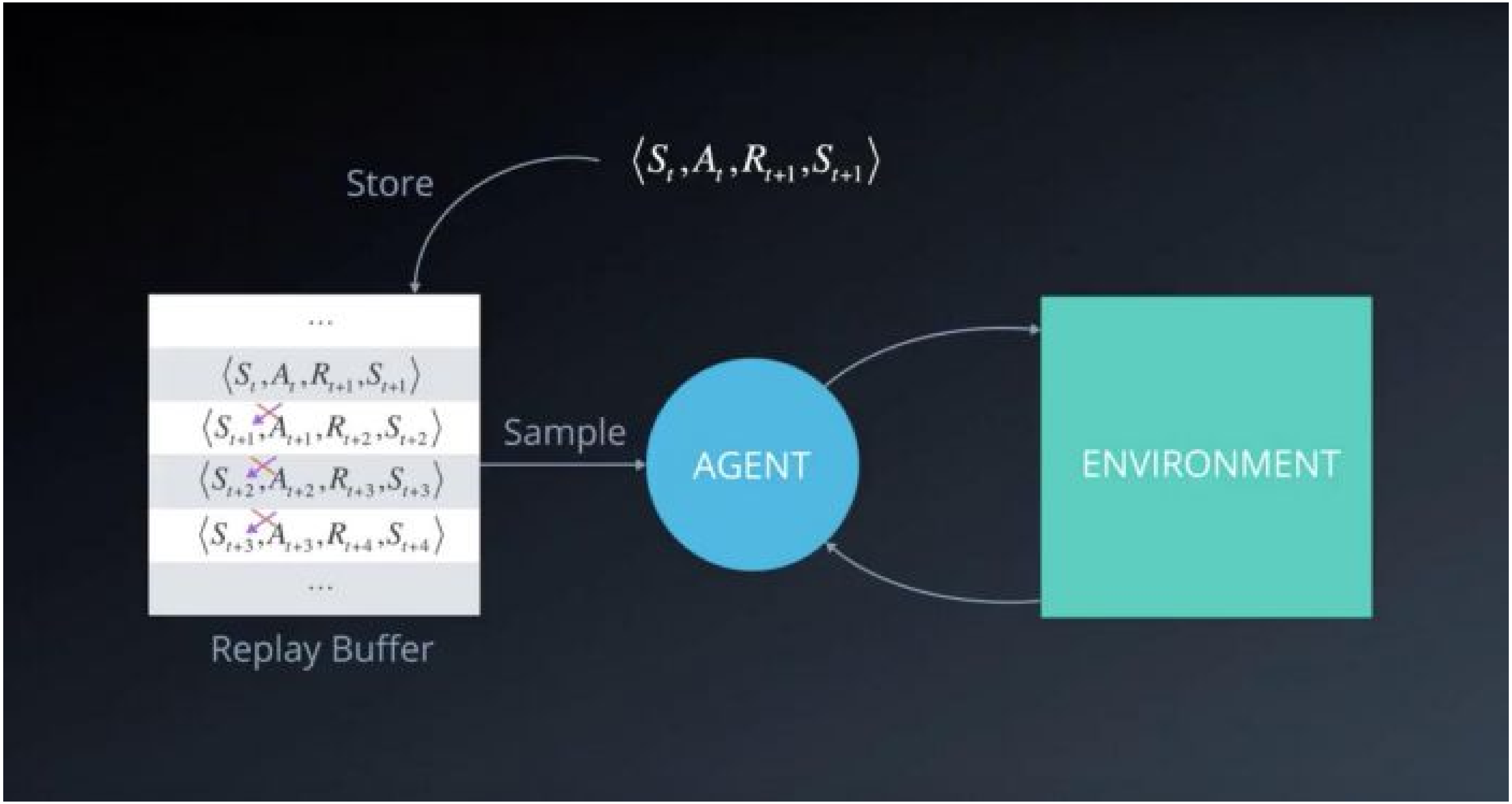
Experience Replay


Learn
AGENT
ENVIRONMENT

$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle$

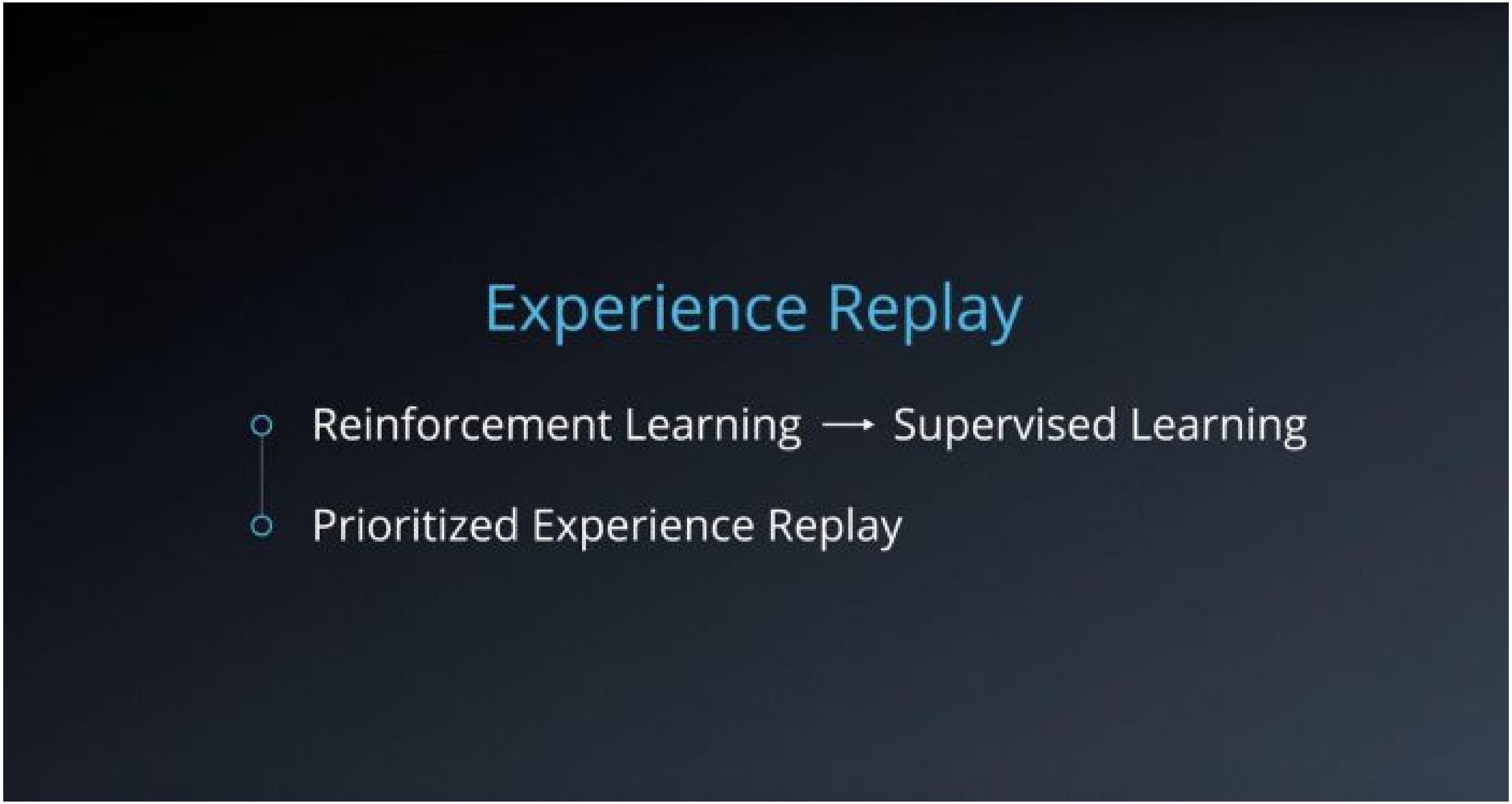
Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Experience Replay


Store
$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle$
...
$\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle$
$\langle S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2} \rangle$
$\langle S_{t+2}, A_{t+2}, R_{t+3}, S_{t+3} \rangle$
$\langle S_{t+3}, A_{t+3}, R_{t+4}, S_{t+4} \rangle$
...
Replay Buffer
Sample
AGENT
ENVIRONMENT

[11] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

DQN Experience Replay means SL approach and Prioritized Experience Replay

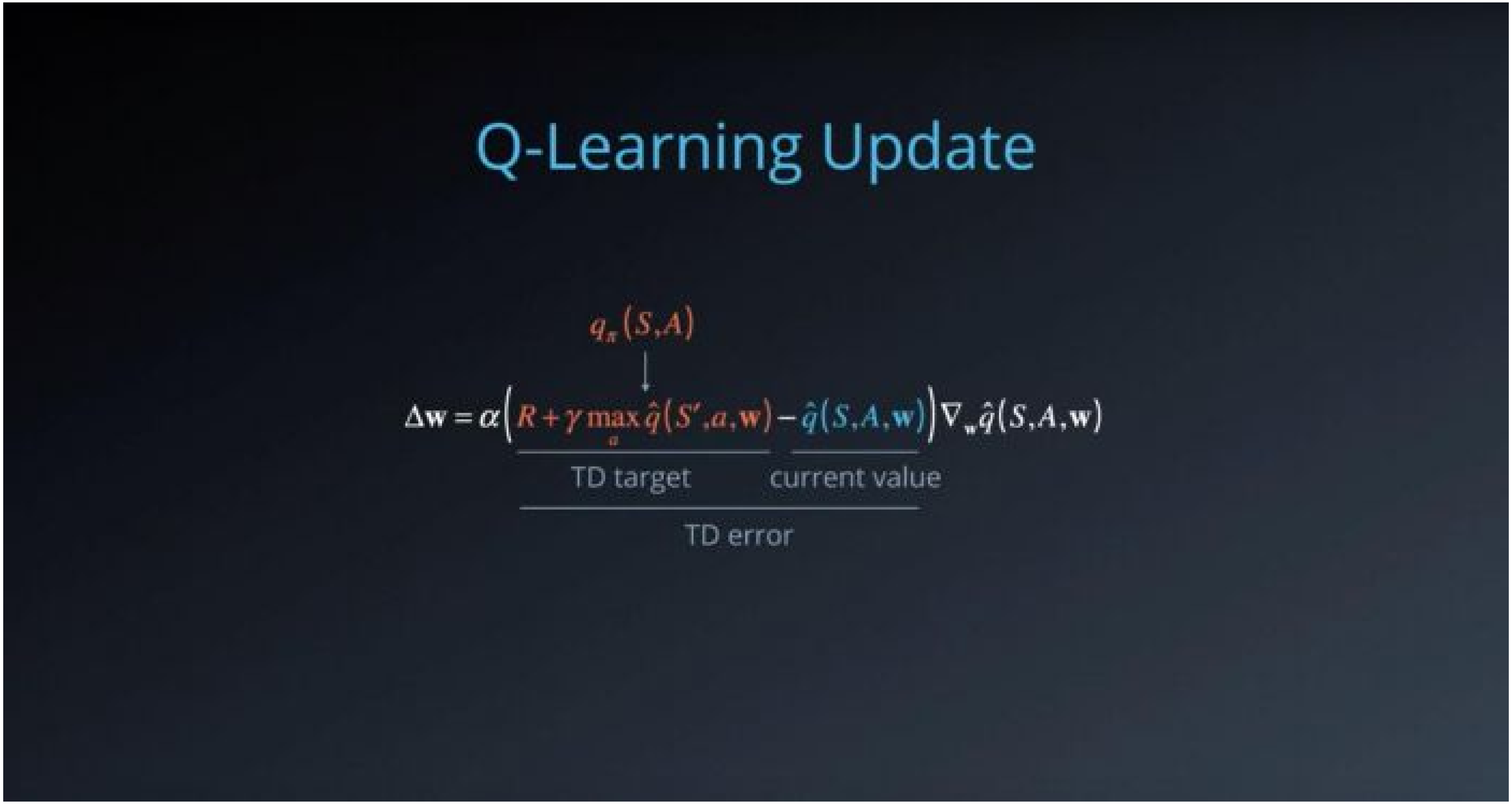


Experience Replay

- Reinforcement Learning → Supervised Learning
- Prioritized Experience Replay

[12] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Fixed Q-Targets

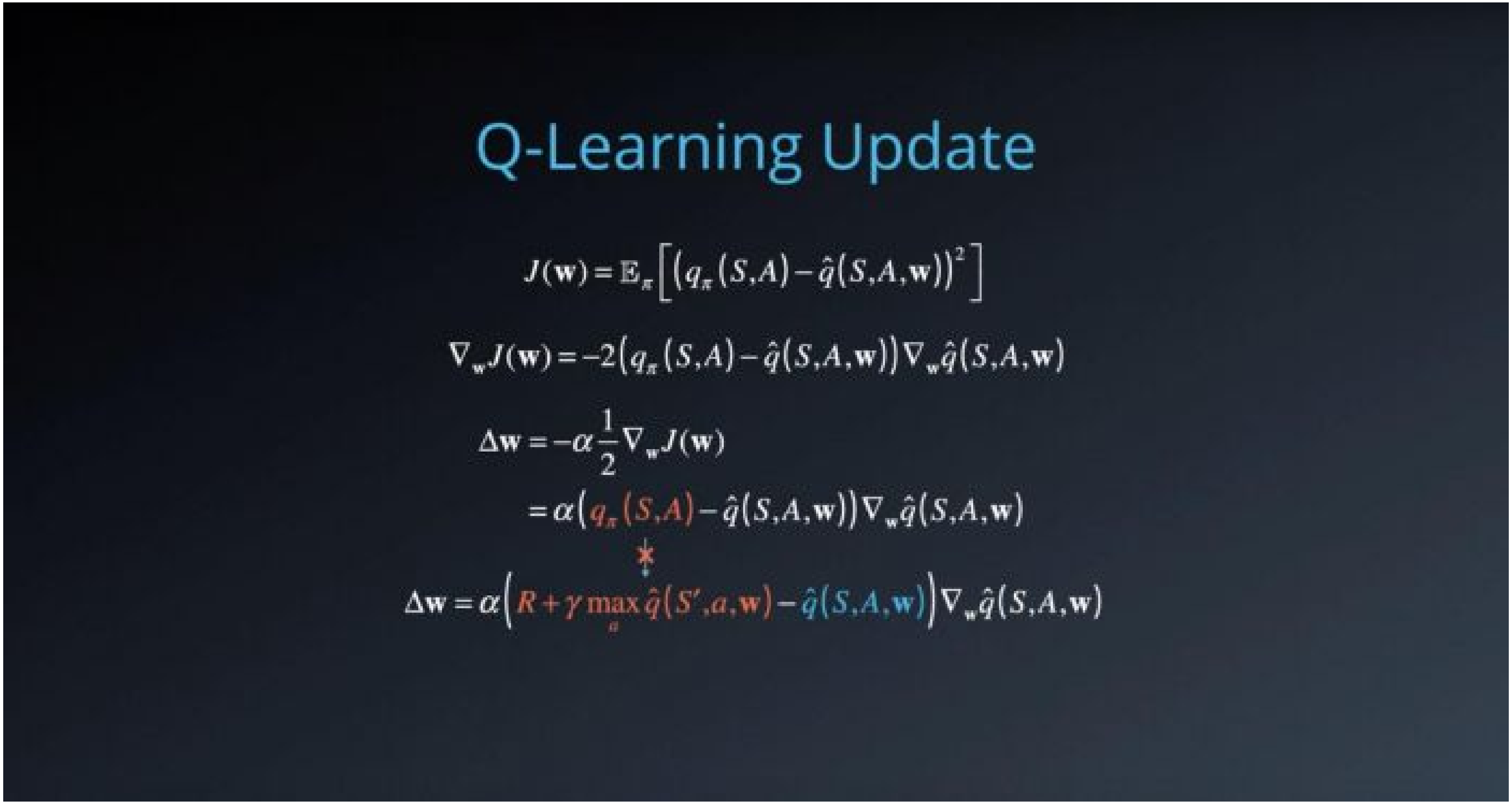


Q-Learning Update

$$\Delta \mathbf{w} = \alpha \Big(\underbrace{\underbrace{R + \gamma \max_a \overset{\overset{q_\pi(S,A)}{\downarrow}}{\hat{q}(S', a, \mathbf{w})}}_{\text{TD target}} - \underbrace{\hat{q}(S, A, \mathbf{w})}_{\text{current value}}}_{\text{TD error}} \Big) \nabla_{\mathbf{w}} \hat{q}(S, A, \mathbf{w})$$

Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Fixed Q-Targets



Q-Learning Update

$$J(\mathbf{w}) = \mathbb{E}_{\pi}\left[\left(q_{\pi}(S,A) - \hat{q}(S,A,\mathbf{w})\right)^{2}\right]$$

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = -2\left(q_{\pi}(S,A) - \hat{q}(S,A,\mathbf{w})\right)\nabla_{\mathbf{w}}\hat{q}(S,A,\mathbf{w})$$

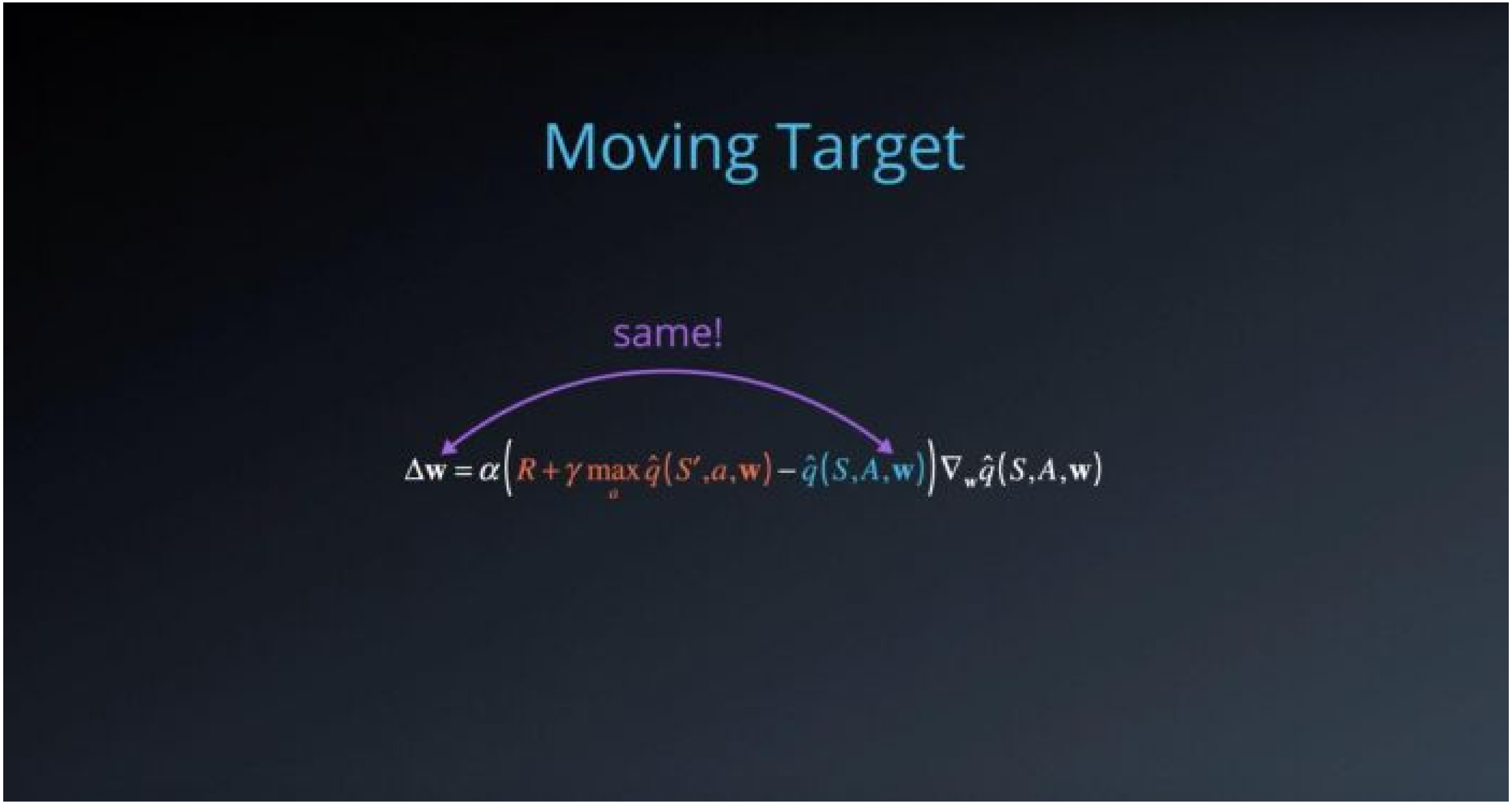
$$\begin{aligned}\Delta\mathbf{w} &= -\alpha\frac{1}{2}\nabla_{\mathbf{w}} J(\mathbf{w})\\ &= \alpha\left(q_{\pi}(S,A) - \hat{q}(S,A,\mathbf{w})\right)\nabla_{\mathbf{w}}\hat{q}(S,A,\mathbf{w})\end{aligned}$$

$$\ast$$

$$\Delta\mathbf{w} = \alpha\left(R + \gamma\max_{a}\hat{q}(S',a,\mathbf{w}) - \hat{q}(S,A,\mathbf{w})\right)\nabla_{\mathbf{w}}\hat{q}(S,A,\mathbf{w})$$

Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Fixed Q-Targets

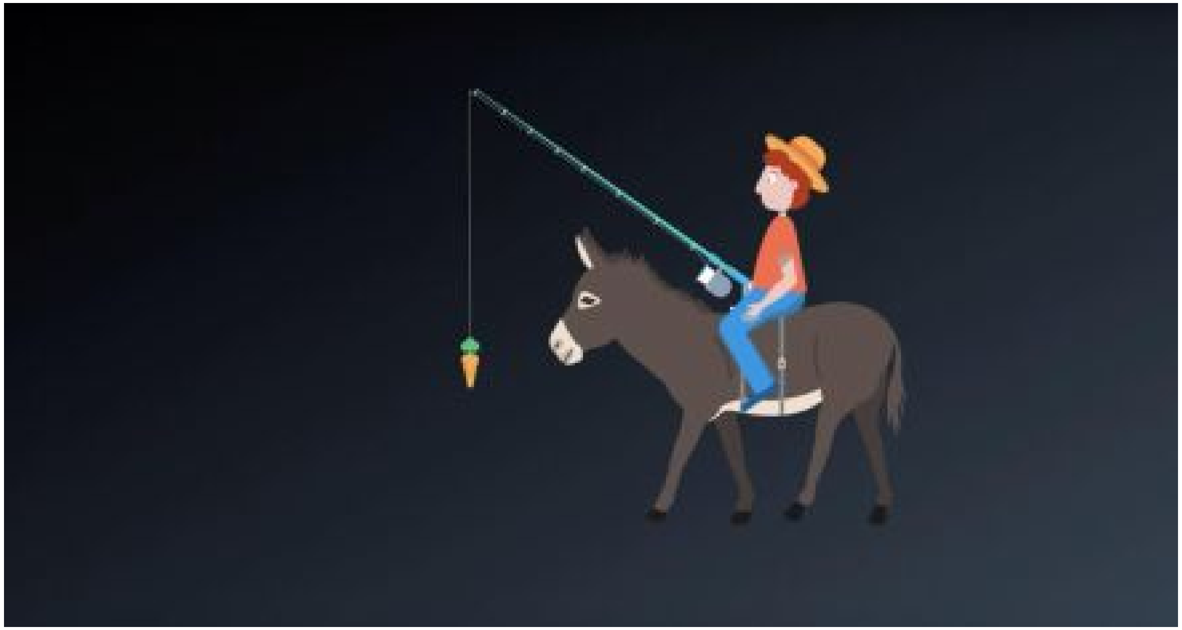


Moving Target

$$\Delta \mathbf{w} = \alpha \left(R + \gamma \max_{a} \hat{q}(S', a, \mathbf{w}) - \hat{q}(S, A, \mathbf{w}) \right) \nabla_{\mathbf{w}} \hat{q}(S, A, \mathbf{w})$$

[15] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

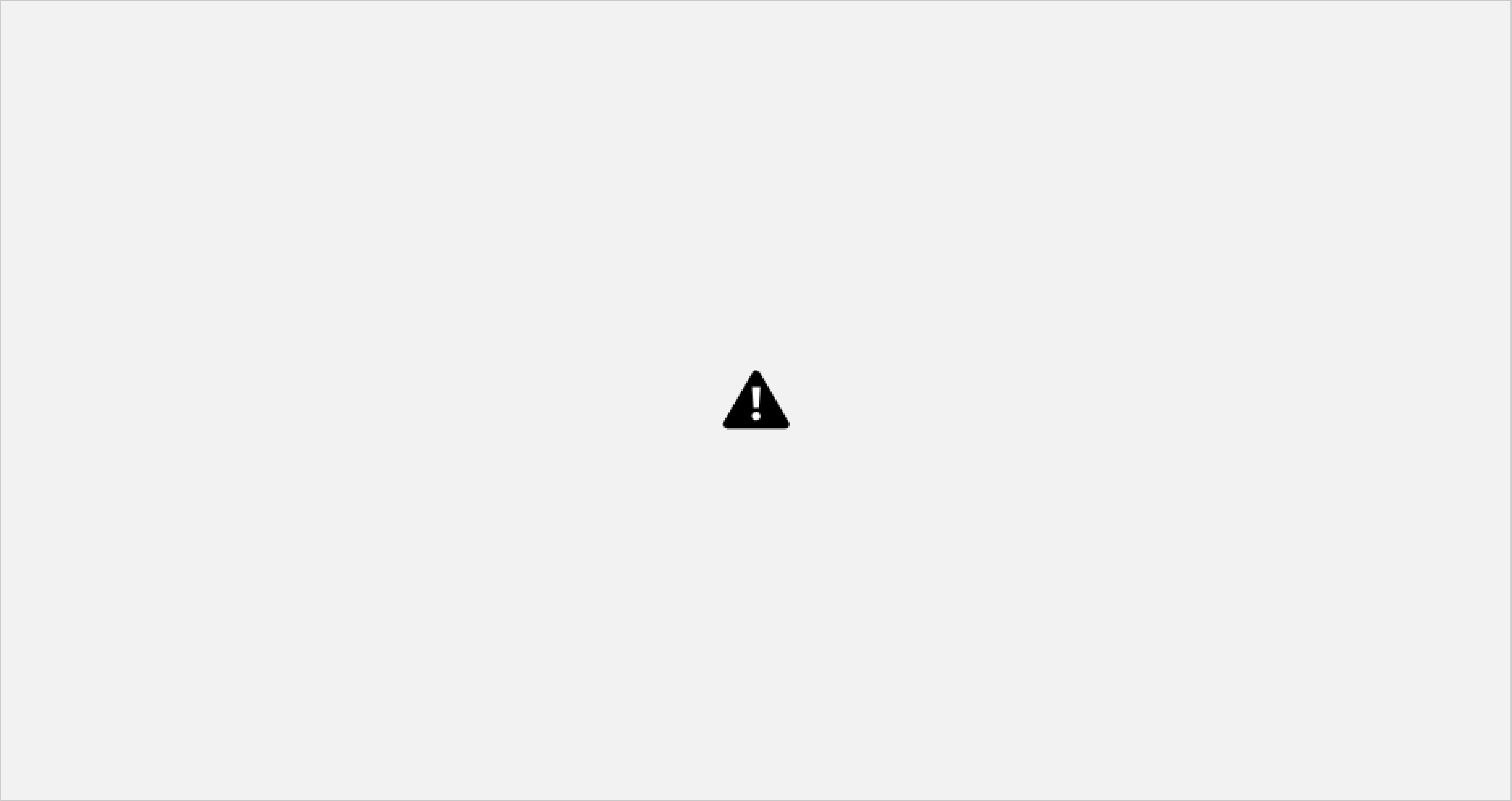
Fixed Q-Targets



Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] {

https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Fixed Q-Targets



[17] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Deep Q-Learning Algorithm

Double DQN

Deep Q-Learning [tends to overestimate (refer to "Issues in Using Function Approximation for Reinforcement Learning"

[18] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

paper)](https://www.ri.cmu.edu/pub_files/pub1/thrun_sebastian_1993_1/thrun_sebastian_1993_1.pdf) action values. Double Q-Learnin
refer to "Dee Reinforcement Learnin with Double Q-learnin" aerhtts:arxiv.orabs1509.06461

Double DQN

Deep Q-Learning [tends to overestimate (refer to "Issues in Using Function Approximation for Reinforcement Learning" paper)](https://www.ri.cmu.edu/pub_files/pub1/thrun_sebastian_1993_1/thrun_sebastian_1993_1.pdf) action values. Double Q-Learnin refer to "Dee Reinforcement Learnin with Double Q-learnin" aerhtts:arxiv.orabs1509.06461

Prioritized Experience Replay

Deep Q-Learning samples experience transitions uniformly from a replay memory. [Prioritized experienced replay (refer to "Prioritized experienced replay"

paper)](https://arxiv.org/abs/1511.05952) is based on the idea that the agent can learn more effectively from some transitions than from others, and the more imortant transitions should be samled with hiher robabilit.

Prioritized Experience Replay

Deep Q-Learning samples experience transitions uniformly from a replay memory. [Prioritized experienced replay (refer to "Prioritized experienced replay"

paper)](https://arxiv.org/abs/1511.05952) is based on the idea that the agent can learn more effectively from some transitions than from others, and the more imortant transitions should be samled with hiher robabilit.

Dueling DQN

Currently, in order to determine which states are (or are not) valuable, we have to estimate the corresponding action values for each action. However, by

replacing the traditional Deep Q-Network (DQN) architecture with a [dueling architecture (refer to "Dueling Network Architectures for Deep Reinforcement Learnin" aerhtts:arxiv.orabs1511.06581 we can assess the value of each state without havin to learn the effect of each action.

Rainbow

So far, you've learned about three extensions to the Deep Q-Networks (DQN) algorithm:

- Double DQN (DDQN)
- Prioritized experience replay
- Dueling DQN

But these aren't the only extensions to the DQN algorithm! Many more extensions have been proposed, including:

- Learning from [multi-step bootstrap targets](https://arxiv.org/abs/1602.01783) (as in A3C - you'll learn about this in Policy-based Method)
- [Distributional DQN](https://arxiv.org/abs/1707.06887)
- [Noisy DQN](https://arxiv.org/abs/1706.10295)

[24] Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Rainbo

w

- Each of the six extensions address a different issue with the original DQN algorithm.

- Researchers at Google DeepMind recently tested the performance of an agent that incorporated all six of these modifications. The corresponding algorithm was termed [Rainbow](https://arxiv.org/abs/1710.02298).

- It outperforms each of the individual modifications and achieves state-of-the art performance on Atari 2600 games!

Refer to [scientific article] { https://www.cs.swarthmore.edu/~meeden/cs63/s15/nature15a.pdf } that describes Deep Q-Networks. Refer to [research paper] { https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf } that first introduced the Deep Q-Learning algorithm.

Thank you