ECE 457A TUTORIAL 10: REINFORCEMENT LEARNING

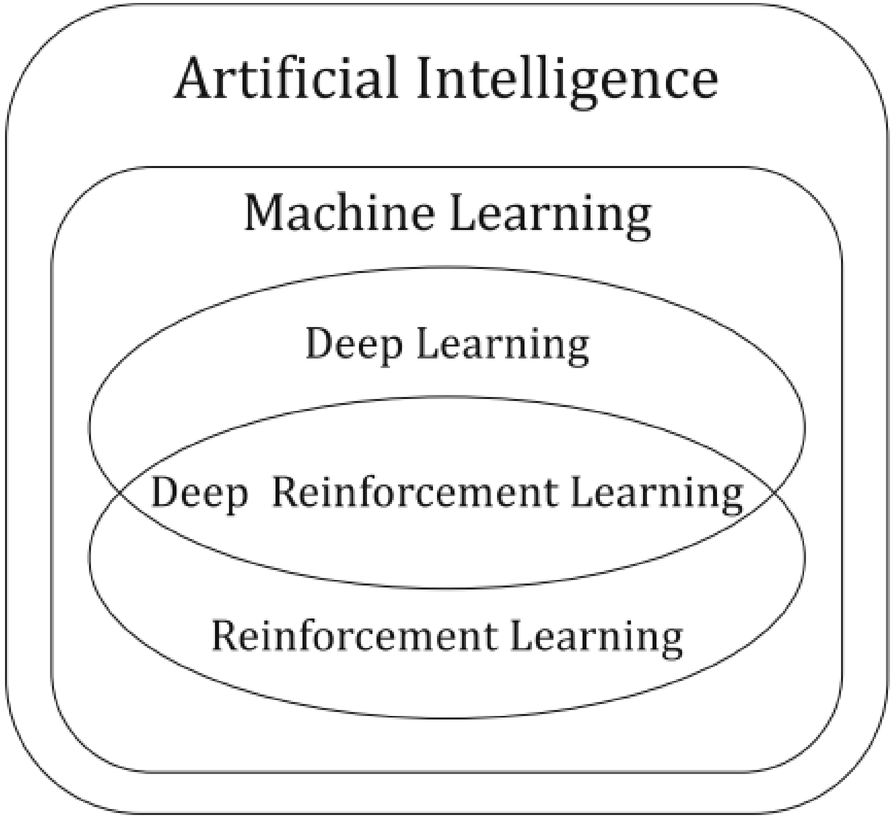
20-Nov-2023

Danial Sadrian Zadeh,
Department of Electrical and Computer Engineering

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

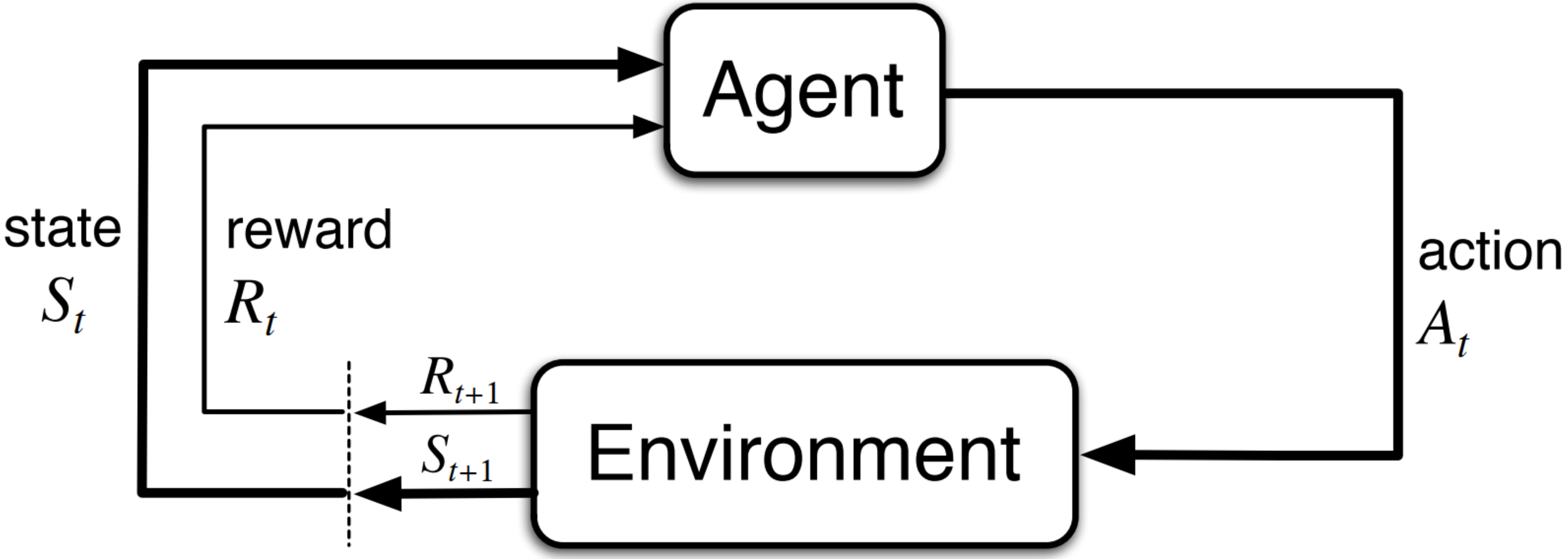
Introduction to Reinforcement Learning (RL)

- Relationship of
  - artificial intelligent (AI),
  - machine learning (ML),
  - deep learning (DL),
  - reinforcement learning (RL),
  - and deep reinforcement learning (DRL)

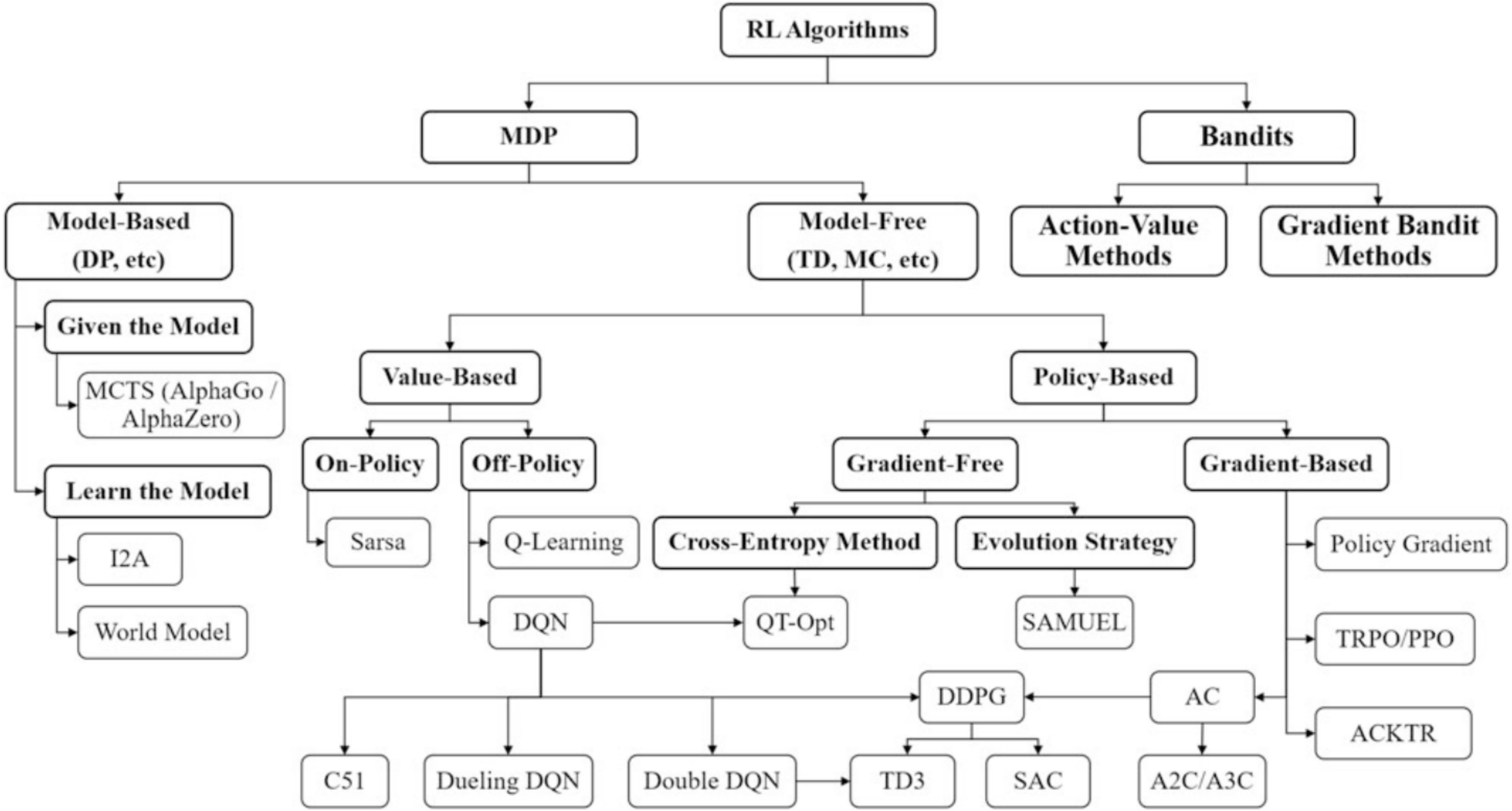


Introduction to Reinforcement Learning (RL)

- The agent-environment interaction in a Markov decision process (MDP)



Summary of RL Algorithms



UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Q-Learning: Off-Policy TD Control

- In this tutorial, we only focus of Q-learning algorithm.
  - It allows agents to learn optimal actions through trial and error.
  - It does not rely on an explicit model of the environment; hence, model-free.
  - It focuses on learning the values (Q-values) associated with different state-action pairs; hence, value-based.
  - It is called off-policy because the updated policy is different from the behavior policy.
  - It is a temporal-difference (TD) learning method because it updates its value estimates at each time step based on the observed rewards and the estimates of future rewards.
  - It tries to find the optimal policy; hence, it is a control problem.

Q-Learning Algorithm

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

Q-Learning Algorithm

- What is α?
  - It is the step-size parameter (or learning rate); it can either be constant or change over time.
- What is γ?
  - It is the discount rate ($0 \leq \gamma \leq 1$); it determines the present value of future rewards.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

- If $\gamma = 0$, the agent is myopic/shortsighted (maximizes immediate rewards).
- If $\gamma = 1$, the agent is farsighted (takes future rewards into account more strongly).

Q-Learning Algorithm

- What is an episode?
  - The agent-environment interaction breaks naturally into subsequences, which we call episodes, such as plays of a game, trips through a maze, or any sort of repeated interaction. Each episode ends in a special state called the terminal state, followed by a reset to a standard starting state or to a sample from a standard distribution of starting states.

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        Choose A from S using policy derived from Q (e.g., ε-greedy)
        Take action A, observe R, S'
        $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
        $S \leftarrow S'$
    until S is terminal

Q-Learning Algorithm

- What is ϵ?
  - It is the probability of taking a random action in an ϵ-greedy policy (trade-off between exploration and exploitation).

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$
Initialize $Q(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
    $S \leftarrow S'$
  until S is terminal

Q-Learning Algorithm

- A Q-table for N states and M actions looks like this:

Actions

States

	A_1	A_2	...	A_M
S_1	$Q(S_1, A_1)$	$Q(S_1, A_2)$		$Q(S_1, A_M)$
S_2	$Q(S_2, A_1)$	$Q(S_2, A_2)$		$Q(S_2, A_M)$
$\vdots$			$\ddots$	$\vdots$
S_N	$Q(S_N, A_1)$	$Q(S_N, A_2)$	...	$Q(S_N, A_M)$

References

1. H. Dong, Z. Ding, and S. Zhang, Deep Reinforcement Learning. Springer Nature, 2020.

2. R. S. Sutton and A. G. Barto, Reinforcement Learning, second edition. MIT Press, 2018.

3. "Epsilon-Greedy Q-learning | Baeldung on Computer Science," Baeldung on Computer Science, Mar. 24, 2023. [Online]. Available: https://www.baeldung.com/cs/epsilon-greedy-q-learning