Q-Learning

Shusen Wang

http://wangshusen.github.io/

Sarsa VS Q-Learning

- Sarsa is for training action-value function, $Q_\pi(s, a)$.
- TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$.
- We used Sarsa for updating value network (critic).

Sarsa VS Q-Learning

- Q-learning is for training the optimal action-value function, $Q^{\star}(s, a)$.
- TD target: $y_t = r_t + \gamma \cdot \max_a Q^{\star}(s_{t+1}, a)$.
- We used Q-learning for updating DQN.

Derive TD Target

Derive TD Target

- We have proved that for all π,

$$Q_\pi(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_\pi(S_{t+1}, A_{t+1})].$$

Derive TD Target

- We have proved that for all π,

$$Q_{\pi}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_{\pi}(S_{t+1}, A_{t+1})].$$

- If π is the optimal policy $\pi^{\star}$, then

$$Q_{\pi^{\star}}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_{\pi^{\star}}(S_{t+1}, A_{t+1})].$$

Derive TD Target

- We have proved that for all π,

$$Q_{\pi}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_{\pi}(S_{t+1}, A_{t+1})].$$

- If π is the optimal policy $\pi^{\star}$, then

$$Q_{\pi^{\star}}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_{\pi^{\star}}(S_{t+1}, A_{t+1})].$$

- $Q_{\pi^{\star}}$ and $Q^{\star}$ both denote *the optimal action-value function*.

Derive TD Target

- We have proved that for all π,

$$Q_\pi(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_\pi(S_{t+1}, A_{t+1})].$$

- If π is the optimal policy $\pi^\star$, then

$$Q_{\pi^\star}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q_{\pi^\star}(S_{t+1}, A_{t+1})].$$

- $Q_{\pi^\star}$ and $Q^\star$ both denote *the optimal action-value function*.

Identity: $Q^\star(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q^\star(S_{t+1}, A_{t+1})]$.

Derive TD Target

Identity: $Q^{\star}(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q^{\star}(S_{t+1}, A_{t+1})]$.

- The action A_{t+1} is computed by

$$A_{t+1} = \underset{a}{\operatorname{argmax}}\, Q^{\star}(S_{t+1}, a).$$

- Thus $Q^{\star}(S_{t+1}, A_{t+1}) = \max_{a} Q^{\star}(S_{t+1}, a)$.

Derive TD Target

Identity: $Q^\star(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q^\star(S_{t+1}, A_{t+1})].$

- Thus $Q^\star(S_{t+1}, A_{t+1}) = \max_a Q^\star(S_{t+1}, a).$

Derive TD Target

Identity: $Q^\star(s_t, a_t) = \mathbb{E}[R_t + \gamma \cdot Q^\star(S_{t+1}, A_{t+1})].$

$$= \max_a Q^\star(S_{t+1}, a)$$

Identity: $Q^\star(s_t, a_t) = \mathbb{E}\left[R_t + \gamma \cdot \max_a Q^\star(S_{t+1}, a)\right].$

Derive TD Target

Identity: $Q^{\star}(s_t, a_t) = \mathbb{E}\left[R_t + \gamma \cdot \max_{a} Q^{\star}(S_{t+1}, a)\right].$

Derive TD Target

Identity: $Q^{\star}(s_t, a_t) = \mathbb{E}\left[\underline{R_t} + \gamma \cdot \max_{a} Q^{\star}\boxed{(S_{t+1},} a)\right].$

$\approx r_t$

$==> s_t$

Derive TD Target

Identity: $Q^{\star}(s_t, a_t) = \mathbb{E}\left[R_t + \gamma \cdot \max_{a} Q^{\star}(S_{t+1}, a) \right].$

$\approx r_t$

$\approx \max_{a} Q^{\star}(s_{t+1}, a)$

Derive TD Target

Identity: $Q^{\star}(s_t, a_t) = \underline{\mathbb{E}\left[R_t + \gamma \cdot \max_{a} Q^{\star}(S_{t+1}, a)\right]}.$

$$\approx \boxed{r_t + \gamma \cdot \max_{a} Q^{\star}(s_{t+1}, a)}$$

TD target y_t

Q-Learning: Tabular Version

Q-Learning (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \max_{a} Q^{\star}(s_{t+1}, a)$.

Q-Learning (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \boxed{\max_{a} Q^{\star}(s_{t+1}, a)}$.

	Action a_1	Action a_2	Action a_3	Action a_4	$\cdots$
State s_1					
State s_2					
State s_3					
$\vdots$					

Q-Learning (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \max_{a} Q^{\star}(s_{t+1}, a)$.
- TD error: $\delta_t = Q^{\star}(s_t, a_t) - y_t$.

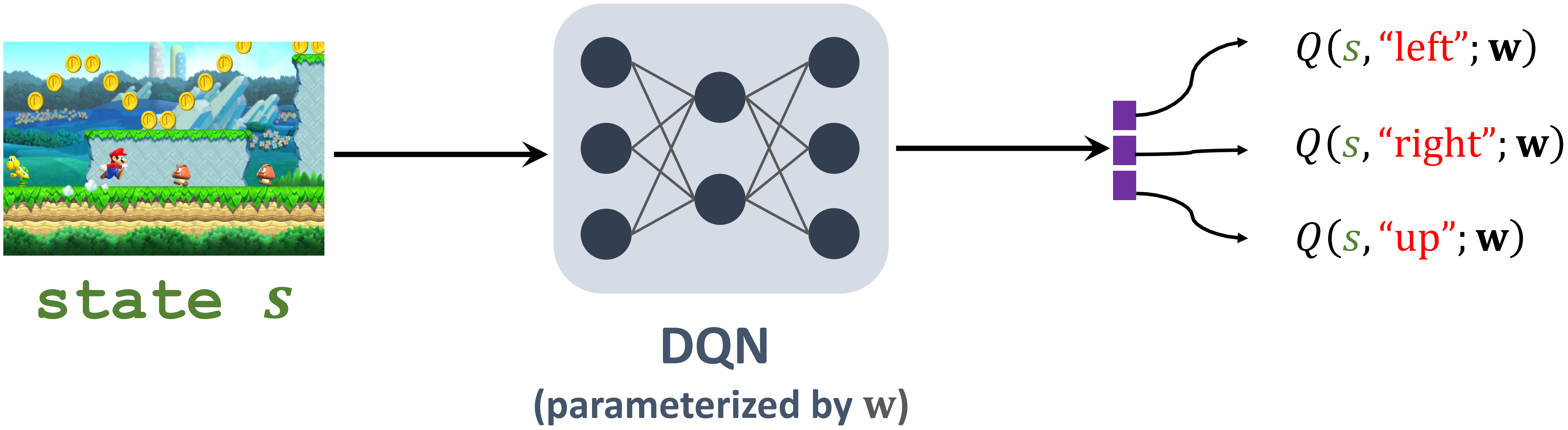
Q-Learning (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \max_{a} Q^{\star}(s_{t+1}, a)$.
- TD error: $\delta_t = Q^{\star}(s_t, a_t) - y_t$.
- Update: $Q^{\star}(s_t, a_t) \leftarrow Q^{\star}(s_t, a_t) - \alpha \cdot \delta_t$.

Q-Learning: DQN Version

DQN Version

- Approximate $Q^{\star}(s, a)$ by DQN, $Q(s, a; \mathbf{w})$.



DQN Version

- Approximate $Q^\star(s, a)$ by DQN, $Q(s, a; \mathbf{w})$.
- DQN controls the agent by: $a_t = \underset{a}{\operatorname{argmax}}\, Q(s_t, a; \mathbf{w})$.
- We seek to learn the parameter, $\mathbf{w}$.

Q-Learning (DQN Version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \max_{a} Q(s_{t+1}, a; \mathbf{w})$.

Q-Learning (DQN Version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- TD target: $y_t = r_t + \gamma \cdot \max_{a} Q(s_{t+1}, a; \mathbf{w})$.
- TD error: $\delta_t = Q(s_t, a_t; \mathbf{w}) - y_t$.
- Update: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial\, Q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}}$.

Summary

- **Goal:** Learn the optimal action-value function $Q^{\star}$.
- **Tabular version** (directly learn $Q^{\star}$).
  - There are finite states and actions.
  - Draw a table, and update the table by Q-learning.
- **DQN version** (function approximation).
  - Approximate $Q^{\star}$ by the DQN, $Q(s, a; \mathbf{w})$.
  - Update the parameter, $\mathbf{w}$, by Q-learning.

Thank you!