Sarsa

Shusen Wang

http://wangshusen.github.io/

Derive TD Target

Discounted Return

Definition of discounted return:

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \gamma^4 \cdot R_{t+4} + \cdots$

Discounted Return

Definition of discounted return:

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \gamma^4 \cdot R_{t+4} + \cdots$

$$= \gamma \cdot (R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + \gamma^3 \cdot R_{t+4} + \cdots)$$

Discounted Return

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \gamma^4 \cdot R_{t+4} + \cdots$

$$= R_t + \gamma \cdot \underbrace{(R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + \gamma^3 \cdot R_{t+4} + \cdots)}_{= U_{t+1}}$$

Discounted Return

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \gamma^4 \cdot R_{t+4} + \cdots$

$$= R_t + \gamma \cdot \underbrace{(R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + \gamma^3 \cdot R_{t+4} + \cdots)}_{= U_{t+1}}$$

Identity: $U_t = R_t + \gamma \cdot U_{t+1}$.

Derive TD Target

Identity: $U_t = R_t + \gamma \cdot U_{t+1}$.

- Assume R_t depends on (S_t, A_t, S_{t+1}).
- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$

Derive TD Target

Identity: $U_t = R_t + \gamma \cdot U_{t+1}$.

- Assume R_t depends on (S_t, A_t, S_{t+1}).
- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$

$$= \mathbb{E}[R_t + \gamma \cdot U_{t+1} | s_t, a_t]$$

Derive TD Target

Identity: $U_t = R_t + \gamma \cdot U_{t+1}$.

- Assume R_t depends on (S_t, A_t, S_{t+1}).
- $$Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$$
$$= \mathbb{E}[R_t + \gamma \cdot U_{t+1} | s_t, a_t]$$
$$= \mathbb{E}[R_t | s_t, a_t] + \gamma \cdot \mathbb{E}[U_{t+1} | s_t, a_t]$$

Derive TD Target

Identity: $U_t = R_t + \gamma \cdot U_{t+1}$.

- Assume R_t depends on (S_t, A_t, S_{t+1}).
- $Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$

$$= \mathbb{E}[R_t + \gamma \cdot U_{t+1} | s_t, a_t]$$

$$= \mathbb{E}[R_t | s_t, a_t] + \gamma \cdot \mathbb{E}[U_{t+1} | s_t, a_t]$$

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

Derive TD Target

- Assume R_t depends on (S_t, A_t, S_{t+1}).
- $$Q_\pi(s_t, a_t) = \mathbb{E}[U_t | s_t, a_t]$$
$$= \mathbb{E}[R_t + \gamma \cdot U_{t+1} | s_t, a_t]$$
$$= \mathbb{E}[R_t | s_t, a_t] + \gamma \cdot \mathbb{E}[U_{t+1} | s_t, a_t]$$
$$= \mathbb{E}[R_t | s_t, a_t] + \gamma \cdot \mathbb{E}[Q_\pi(S_{t+1}, A_{t+1}) | s_t, a_t].$$

Derive TD Target

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

Derive TD Target

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

- We do not know the expectation.
- Approximate it using Monte Carlo (MC).

Derive TD Target

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

$\approx r_t$

$\approx Q_\pi(s_{t+1}, a_{t+1})$

Derive TD Target

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

$$\approx r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$$

TD target y_t

Derive TD Target

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

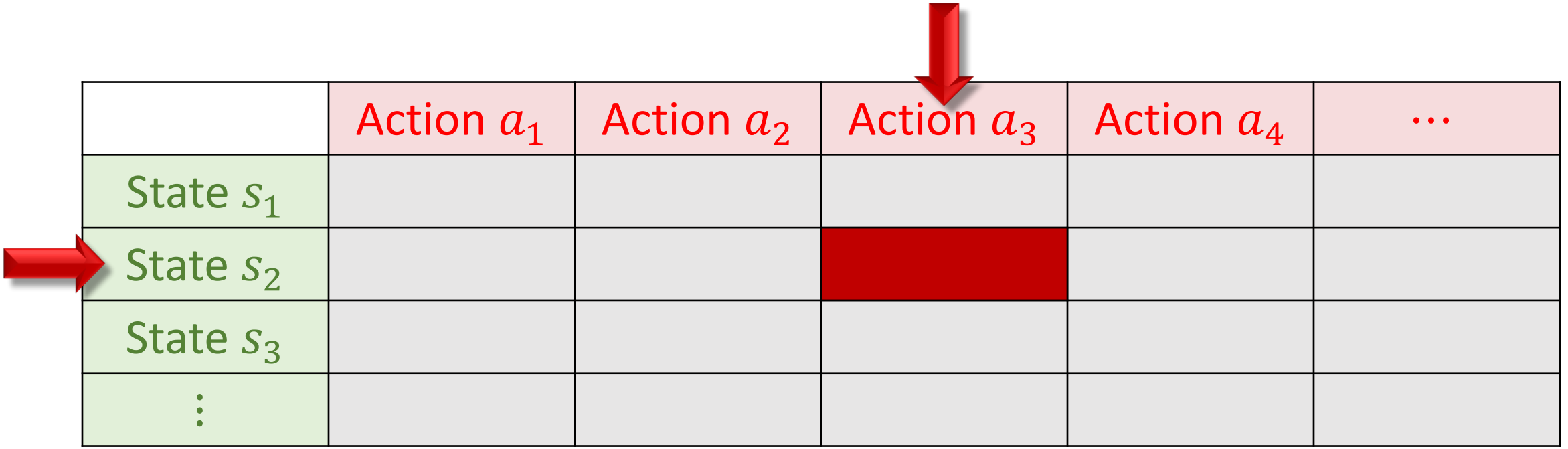
$\approx y_t$

TD learning: Encourage $Q_\pi(s_t, a_t)$ to approach y_t.

Sarsa: Tabular Version

Tabular Version

- We want to learn $Q_\pi(s, a)$.
- Suppose the numbers of states and actions are finite.
- Draw a table and learn the entries.



	Action a_1	Action a_2	Action a_3	Action a_4	⋯
State s_1					
State s_2					
State s_3					
⋮					

Sarsa (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- Sample $a_{t+1} \sim \pi(\cdot\,|s_{t+1})$, where π is the policy function.
- TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$.

Sarsa (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- Sample $a_{t+1} \sim \pi(\cdot\,|s_{t+1})$, where π is the policy function.
- TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$.

	Action a_1	Action a_2	Action a_3	Action a_4	⋯
State s_1					
State s_2					
State s_3					
⋮					

Sarsa (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- Sample $a_{t+1} \sim \pi(\cdot\,|s_{t+1})$, where π is the policy function.
- TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$.
- TD error: $\delta_t = Q_\pi(s_t, a_t) - y_t$.

Sarsa (tabular version)

- Observe a transition (s_t, a_t, r_t, s_{t+1}).
- Sample $a_{t+1} \sim \pi(\cdot\,|s_{t+1})$, where π is the policy function.
- TD target: $y_t = r_t + \gamma \cdot Q_\pi(s_{t+1}, a_{t+1})$.
- TD error: $\delta_t = Q_\pi(s_t, a_t) - y_t$.
- Update: $Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) - \alpha \cdot \delta_t$.

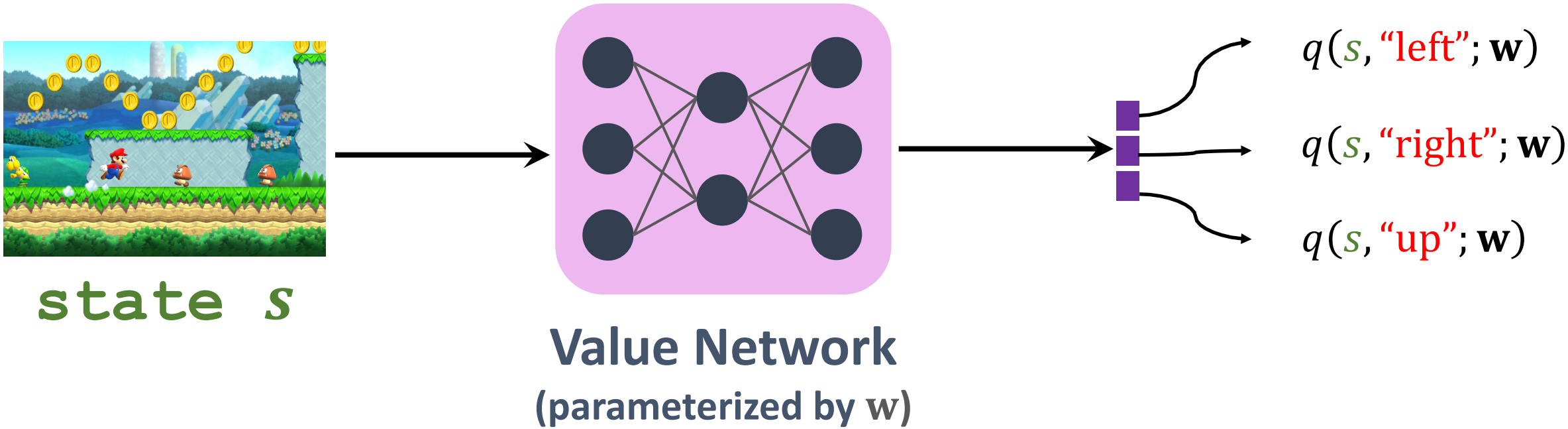
Sarsa's Name

- Use $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ for updating Q_π.
- State-Action-Reward-State-Action (SARSA).

Sarsa: Neural Network Version

Value Network Version

- Approximate $Q_\pi(s, a)$ by the value network, $q(s, a; \mathbf{w})$.



Value Network Version

- Approximate $Q_\pi(s, a)$ by the value network, $q(s, a; \mathbf{w})$.
- q is used as the critic who evaluates the actor. (Actor-Critic Method.)
- We want to learn the parameter, $\mathbf{w}$.

TD Error & Gradient

- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w})$.
- TD error: $\delta_t = q(s_t, a_t; \mathbf{w}) - y_t$.

TD Error & Gradient

- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w})$.
- TD error: $\delta_t = q(s_t, a_t; \mathbf{w}) - y_t$.
- Loss: $\delta_t^2/2$.
- Gradient: $\frac{\partial\, \delta_t^2/2}{\partial\, \mathbf{w}} = \delta_t \cdot \frac{\partial\, q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}}$.

TD Error & Gradient

- TD target: $y_t = r_t + \gamma \cdot q(s_{t+1}, a_{t+1}; \mathbf{w})$.
- TD error: $\delta_t = q(s_t, a_t; \mathbf{w}) - y_t$.
- Loss: $\delta_t^2/2$.
- Gradient: $\frac{\partial\, \delta_t^2/2}{\partial\, \mathbf{w}} = \delta_t \cdot \frac{\partial\, q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}}$.
- Gradient descent: $\mathbf{w} \leftarrow \mathbf{w} - \alpha \cdot \delta_t \cdot \frac{\partial\, q(s_t, a_t; \mathbf{w})}{\partial\, \mathbf{w}}$.

Summary

- **Goal:** Learn the action-value function Q_π.
- **Tabular version** (directly learn Q_π).
  - There are finite states and actions.
  - Draw a table, and update the table using Sarsa.
- **Value network version** (function approximation).
  - Approximate Q_π by the value network $q(s, a; \mathbf{w})$.
  - Update the parameter, $\mathbf{w}$, using Sarsa.
  - Application: actor-critic method.

Thank you!

http://wangshusen.github.io/