# PART 5

## Reinforcement Learning

# MDP

* State space : $S$
* Action space : $A$
* Transition prob: $p(s' | s, a)$ — Action Model

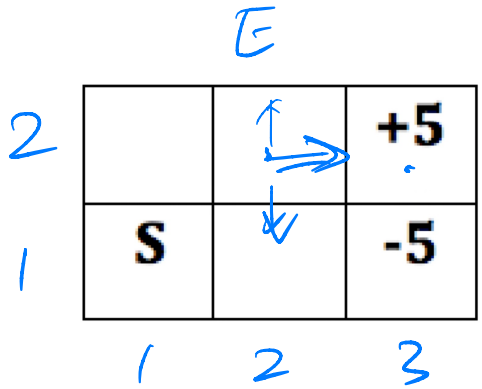
Next state

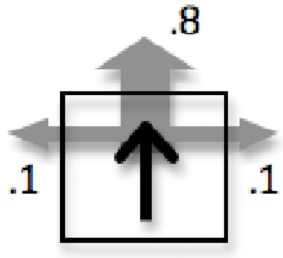
Current state

Current action

* Reward function: $r(s, a, s')$

$$r(s, a) = \sum_{s'} p(s' | s, a) \, r(s, a, s')$$



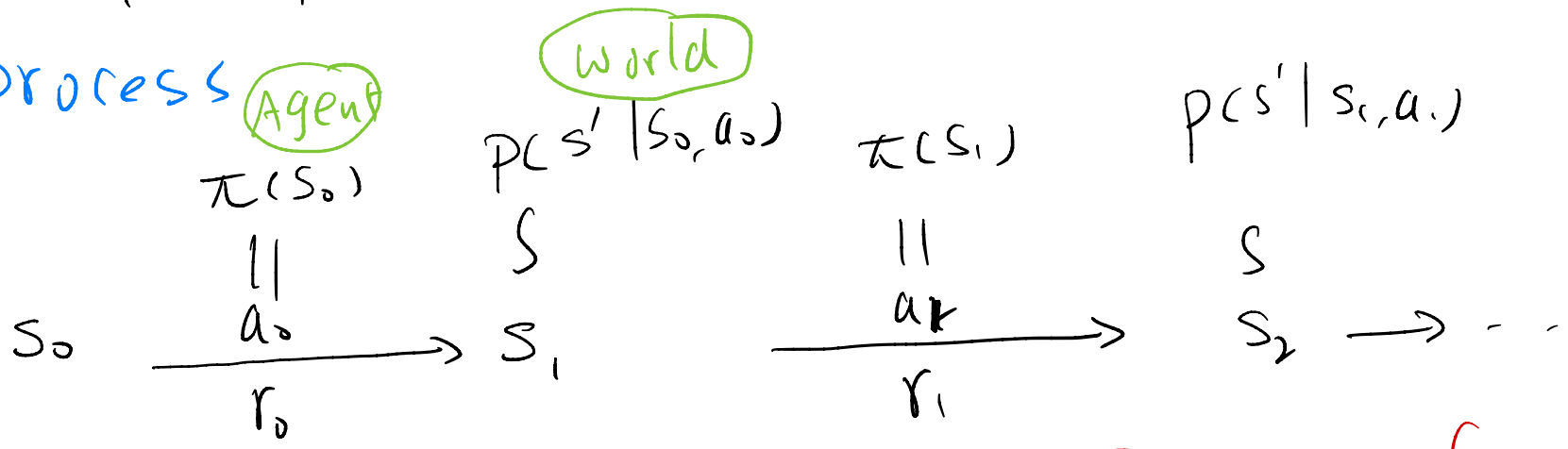

| $s$ | $a$ | $s'$ | $p(s'\|s,a)$ | $r(s,a,s')$ |
|---|---|---|---|---|
| $(2,2)$ | $E$ | $(3,2)$ | 0.8 | +5 |
| | | $(2,2)$ | 0.1 | 0 |
| | | $(2,1)$ | 0.1 | 0 |

$$r((2,2),E) = 0.8 \times 5 + 0.1 \times 0 + 0.1 \times 0 = 0.4$$

One-step expected reward

Policy: $\pi : S \to A$

* Process




Discount factor

$0 < \gamma < 1$

* Math reason
* Intuitive

Discounted total rewards

$$R^{\pi}(s_0) = r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots$$

(state) Value Function: $V^{\pi}(s) = E[R^{\pi}(s)]$

Total rewards for following $\pi$ from $s$

random: Yes ~~No~~

Theorem: Exist $\pi^*$ s.t. $\forall \pi$

$$V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall s$$

optimal policy: $\pi^*$

Optimal Value function: $V^*(s) = V^{\pi^*}(s)$

Task 1: Planning

$p(s'|s,a), r(s,a) \Rightarrow \pi^*$

Task 2: Reinforcement Learning

$\{(s, a, r, s')\} \Rightarrow \pi^*$

Experience tuple

Task 1 : $p(s'|s,a)$ $r(s,a)$ $\Rightarrow \pi^*$

(1) Compute $V^*$

(2) Obtain $\pi^*$ from $V^*$

(2) Optimal **action-state value function**

In $s$, take $a$, then follow $\pi^*$

$$Q^*(s,a) = \underbrace{r(s,a)}_{\text{Current reward}} + \gamma \sum_{s'} p(s'|s,a) \underbrace{V^*(s')}_{\text{Future reward}}$$

$$\pi^*(s) = \arg\max_a Q^*(s,a)$$

Task 1, step 1: Compute $V^*$

Value Iteration

* pick $V_0$

Bellman Operator

* For $k = 0$ to termination

$$V_{k+1}(s) = \max_a \left\{ \underbrace{r(s,a)}_{\text{current}} + \gamma \underbrace{\sum_{s'} p(s'|s,a) V_k(s')}_{\text{Future}} \right\}$$

$\uparrow$ improve

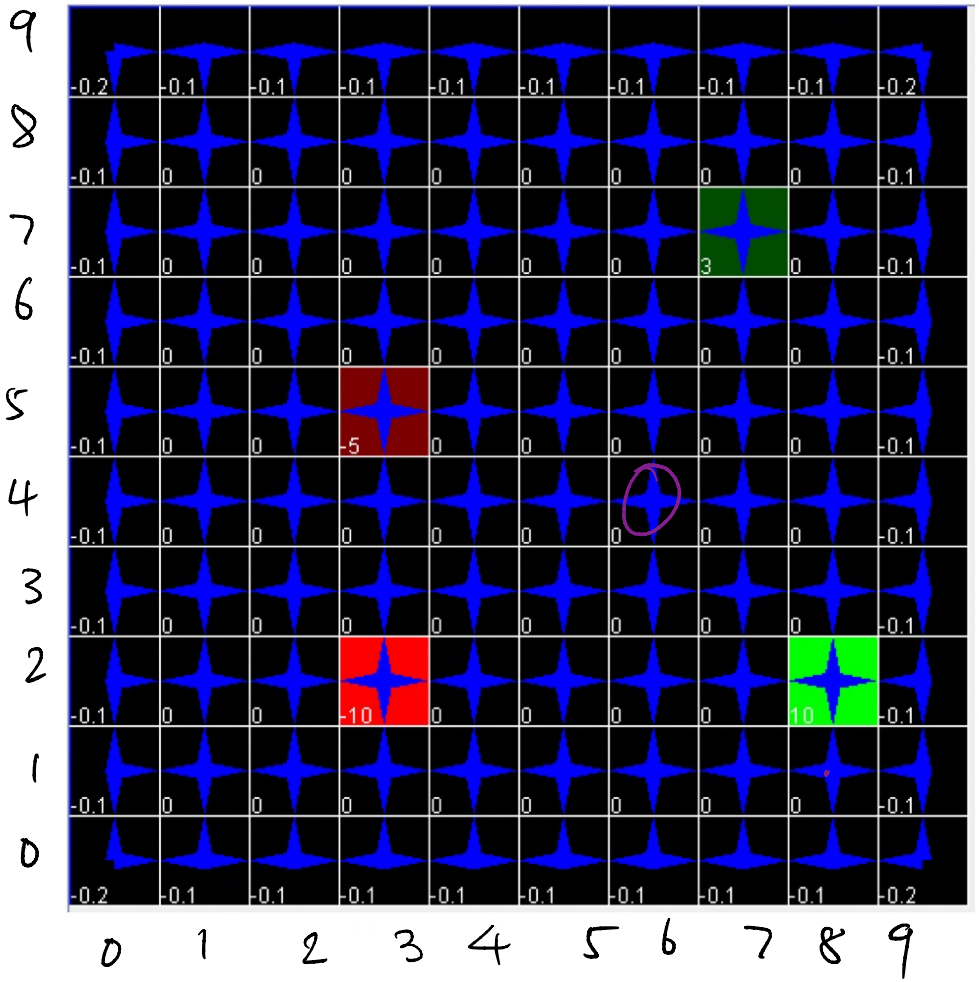
Theorem: $\lim_{k \to \infty} V_k(s) = V^*$

$$V_{k+1}(s) = \max_a \left\{ r(s,a) + \gamma \sum_{s'} p(s'|s,a) V_k(s') \right\}$$

$$\Downarrow \quad k \to \infty$$

$$V^*(s) = \max_a \left\{ r(s,a) + \gamma \sum_{s'} p(s'|s,a) V^*(s') \right\}$$

Bellman's Optimality Equation




$S = (8, 1)$ $V_1(S) = 0$

| | | Current | Future | max a |
|---|---|---|---|---|
| a | up | 0 | large + | ✓ |
| | down | 0 | small − | |
| | left | 0 | small | |
| | right | 0 | small | |

$V_2(S)$ ↗

---

$S = (3, 1)$ $V_1(S) = 0$

| | | Current | Future | max a |
|---|---|---|---|---|
| a | up | | large − | X |
| | down | | small − | |
| | left | | small − | |
| | right | | small − | |

$V_2(S)$ ↘

---

$S = (6, 4)$ ↗

$V_1(S) = 0$ $V_2(S) = 0$ ?

↘

Value iteration for $Q^*(s,a)$ $V^*(s)$

* pick $Q_0$

$Q(s,a)$ Table

| | $a_1$ | $a_2$ | ... |
|---|---|---|---|
| $s_1$ | 0.5 | 2 | ... |
| $s_2$ | 3.2 | 0.1 | ... |

* for $k=0$ to termination

$$Q_{k+1}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q_k(s', a')$$

Current — Future — improve

Theorem: $\lim_{k\to\infty} Q_k(s,a) = Q^*(s,a)$

$\pi_k(s) = \arg\max_a Q_k(s,a)$ Greedy policy based on $Q_k$

Task 2: $\{(s, a, r, s')\} \Rightarrow Q^*(s, a)$

Experience Tuples

* Initialize $Q(s, a)$ — Q-table

* Collect $(s, a, r, s')$ — Feed back

$$Q(s,a) \leftarrow Q(s,a) + \alpha \Delta$$

$\alpha$: learning rate

$$\Delta = \underbrace{r(s,a) + \gamma \max_{a'} Q(s',a')}_{\text{TD target}} - \underbrace{Q(s,a)}_{\text{current estimate}}$$

**Q-learning**: Converges to $Q^*$ if each $(s,a)$ visited infinitely often

$$Q_{k+1}(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q_k(s',a')$$

$$\approx r(s,a) + \gamma \frac{1}{m} \sum_{j=1}^{m} \max_{a'} Q_k(s_j', a') \qquad s_1', \ldots, s_m' \sim p(s'|s,a)$$

$$\Rightarrow \quad r(s,a) + \gamma \max_{a'} Q_k(s_1', a')$$

when $m=1$

* high variance
* unbiased

$(s, a, r, s_1')$

Improve $Q(s,a)$ using $(s,a,r,s')$

$$r + \gamma \max_{a'} Q(s', a')$$

TD target (TD: Time difference)

# Q-learning is off-policy

- Choose $a$ for the state $s$ ($\epsilon$-greedy with $\arg\max_a Q(s,a)$)
- Take action $a$, observe $r$ and $s'$
- Update:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$
$$s \leftarrow s'$$

$Q_{Next}$

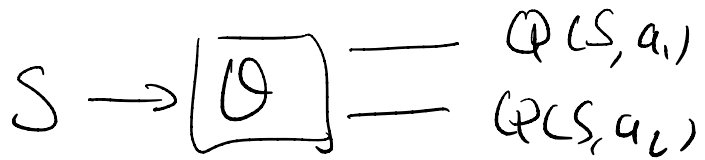
Action for the next time step ?

* For estimation: $\arg\max_{a'} Q(s',a')$

* Actually execute: $\arg\max_{a'} Q_{Next}(s',a')$

L13 Value-Based Deep RL

# Deep Q-Network (DQN)




* Initialize $\theta$

* Repeat: Get $(s, a, r, s')$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \left(Y_\theta - Q(s, a; \theta)\right)^2$$

* Target Network
* Replay Buffer
* Prioritized Replay

$$Y_\theta = \underbrace{r + \gamma \max_{a'} Q(s', a'; \theta)}_{\text{TD target}}$$

## Moving Target

Target

$\theta_0$

$\theta_1 \leftarrow \theta_0 - \alpha \nabla_\theta (Y_{\theta_0} - Q(s_1, a_1; \theta))^2$

$\theta_2 \leftarrow \theta_1 - \alpha \nabla_\theta (Y_{\theta_1} - Q(s_2, a_2; \theta))^2$

. .

Solution: keep a slow-changing target network

DQN

# Review : P11, P13

Update rule for DQN:

TD target

$$\theta \leftarrow \theta - \alpha \nabla_\theta ([r(s,a) + \gamma \max_{a'} Q(s', a'; \theta^-)] - Q(s,a;\theta))^2$$

current  Future estimate

① pick next action : $a' = \arg\max_{a'} Q(s', a'; \theta^-)$

② Evaluate $a'$ : $Q(s', a'; \theta^-)$

$\Rightarrow$ over-estimate of future reward

Update rule for Double DQN:

$$\theta \leftarrow \theta - \alpha \nabla_\theta ([r(s,a) + \gamma Q(s', a^*; \theta^-)] - Q(s,a;\theta))^2$$

where $a^* = \arg\max_{a'} Q(s', a'; \theta)$.

Analogy: Estimate highest score of a class in final exam

① Teacher $\theta^-$ : (a) pick best student $a_1^*$

(b) Estimate score for $a_1^*$

② (a) Teacher $\theta$ pick best student $a_2^*$

(b) Teacher $\theta^-$ estimate score for $a_2^*$

Score 1 $>$ Score 2 ☆

which one trust more?

DQN: $s \rightarrow [\theta] \rightarrow Q(s, a_1)$, $Q(s, a_2)$

Discrete Action space

what about continuous action space?

DDPG

# L14: Policy-Based Deep RL

Value-Based Deep RL:

$S \rightarrow [\theta] \rightarrow Q$

$\{(s, a, r, s')\} \Longrightarrow Q(s, a; \theta)$

$\Rightarrow \pi(s) = \arg\max_a Q(s, a; \theta)$

Deterministic

policy-based Deep RL

stochastic

$\{(s, a, r, s')\} \Longrightarrow \pi_\theta(a|s)$

$S \rightarrow [\theta] \rightarrow \pi_\theta(a|s)$

$\downarrow$

loss

| A | $\pi$ |
|---|---|
| N | 0.8 |
| S | 0.1 |
| E | 0.05 |
| W | 0.05 |

$$\nabla_\theta J(\theta) = \nabla_\theta \int \pi_\theta(\tau)\, r(\tau)\, d\tau$$

$$= \int r(\tau)\, \nabla_\theta \pi_\theta(\tau)\, d\tau$$

$$= \int r(\tau)\, \pi_\theta(\tau)\, \nabla_\theta \log \pi_\theta(\tau)\, d\tau$$

$$= \int \pi_\theta(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)\, r(\tau)\, d\tau$$

log gradient trick

$$\nabla_\theta \log \pi_\theta(\tau)$$

$$= \frac{1}{\pi_\theta(\tau)} \nabla_\theta \pi_\theta(\tau)$$

$$\nabla_\theta \pi_\theta(\tau) = \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau)$$

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots s_t, a_t, r_t, \ldots)$$

$$\pi_\theta(\tau) = p(s_0) \prod_t \pi_\theta(a_t|s_t)\, p(s_{t+1}|s_t, a_t)$$

$$\nabla_\theta \log \pi_\theta(\tau) = \nabla_\theta \log p(s_0) + \nabla_\theta \sum_t \log \pi_\theta(a_t|s_t) + \nabla_\theta \sum_t \log p(s_{t+1}|s_t, a_t)$$

$\nabla_\theta \log p(s_0) = 0$; $\nabla_\theta \sum_t \log p(s_{t+1}|s_t, a_t) = 0$

$$\nabla_\theta J = \int \pi_\theta(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)\, r(\tau)\, d\tau$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \sum_t \nabla_\theta \log \pi_\theta(a_t^i | s_t^i)\, r(\tau^i)$$

$$\tau^1, \ldots, \tau^N \sim \pi_\theta(\tau)$$

$$\tau^i = (s_0^i, a_0^i, r_0^i, \ldots s_t^i, a_t^i, r_t^i, \ldots)$$

Supervised Learning: $\{x_i, y_i\}_{i=1}^{N} \Rightarrow p(y|x;\theta)$

$$\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log p(y_i|x_i;\theta)$$

Maximize prob of data ($y_i$)

Current: $\{(s_t^i, a_t^i) \mid i = 1, \dots N, \; t = 1, \dots T\}$

($s_t^i$: $x$, $a_t^i$: $y$)

$$\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_i \sum_t \nabla_\theta \log p(a_t^i | s_t^i; \theta)$$

maximize prob of the $a_t^i$'s

Imitation Learning: Good if trajectories are expert demonstration

# Policy Gradient (REINFORCE)

$$\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_i \sum_t \nabla_\theta \log p(a_t^i | s_t^i; \theta) \, r(\tau^i)$$

$r(\tau^i) > 0$ : positive experience

Increase prob of actions

$r(\tau^i) < 0$ : Negative experience

Decrease prob of action

# Actor-Critic Algo

$V^*(s)$: optimal total reward starting from $s$

$Q^*(s,a)$: total reward for taking $a$ in $s$ and then act optimally

$$V^*(s) = \max_a Q^*(s,a)$$

$\pi$: Stochastic $\pi(a|s)$

$V^\pi(s)$: total reward for following $\pi$ starting from $s$

$Q^\pi(s,a)$: total reward for taking $a$ in $s$ and then following $\pi$

$$V^\pi(s) = \sum_a Q^\pi(s,a)\,\pi(a|s)$$ : Average of $Q$ over actions

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

Advantage function of $a$ in $a$

Policy Gradient (REINFORCE)

$$\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_i \sum_t \nabla_\theta \log p(a_t^i | s_t^i; \theta) \, r(\tau^i)$$

Actor – Critic (AC) Algo

$$\theta \leftarrow \theta + \alpha \frac{1}{N} \sum_i \sum_t \nabla_\theta \log p(a_t^i | s_t^i; \theta) \, A^\pi(s_t^i, a_t^i)$$

$A^\pi(s_t^i, a_t^i) > 0$ : $a_t^i$ better than average
increase its prob

$A^\pi(s_t^i, a_t^i) < 0$ : $a_t^i$ worse than average
decrease it prob

$\pi(a|s)$

How to estimate:

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

$$= r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

$(s_t, a_t, r_t, s_{t+1})$

$$Q^{\pi}(s_t, a_t) \approx \underbrace{r_t}_{\text{current}} + \underbrace{\gamma V^{\pi}(s_{t+1})}_{\text{future}}$$

TD target

$\Rightarrow$ P28, L14

Actor-Critic Algs

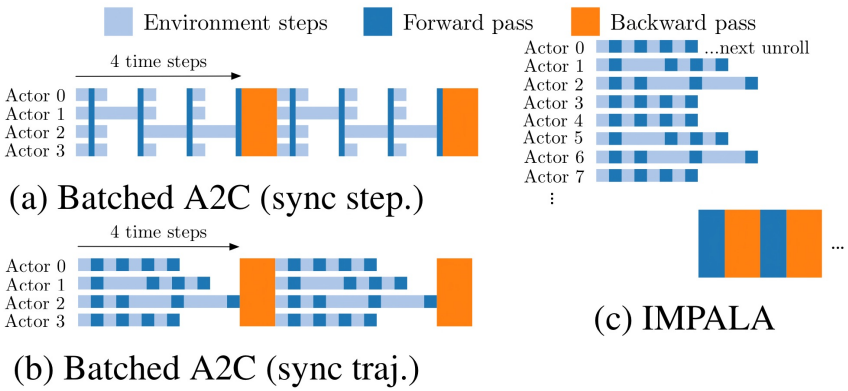
* On-policy, low sample efficiency

A3C

* Multiple actors updates the same parameters

IMPALA:

* One learner learning from experiences from multiple actors



PPO: Makes training more stable

Recall Actor-Critic Algorithm:

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}[\gamma^t \nabla_\theta \log \pi_\theta(a_t^i|s_t^i) A^\pi(s_t^i, a_t^i)]$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}[\gamma^t \frac{\nabla_\theta \pi_\theta(a_t^i|s_t^i)}{\pi_\theta(a_t^i|s_t^i)} A^\pi(s_t^i, a_t^i)]$$

$\frac{d \log y}{dx} = \frac{1}{y}\frac{dy}{dx}$

PPO

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}[\gamma^t \nabla_\theta \frac{\pi_\theta(a_t^i|s_t^i)}{\pi_{old}(a_t^i|s_t^i)} A^\pi(s_t^i, a_t^i)]$$

ratio $\rightarrow$ $r_t = \frac{\text{New prob of taking } a_t^i}{\text{old prob of taking } a_t^i}$

$1-\varepsilon \leq r_t \leq 1+\varepsilon$

Soft Actor - Critic

* Better exploration

Hide and Seek

* Multiagent RL