Policy Gradient

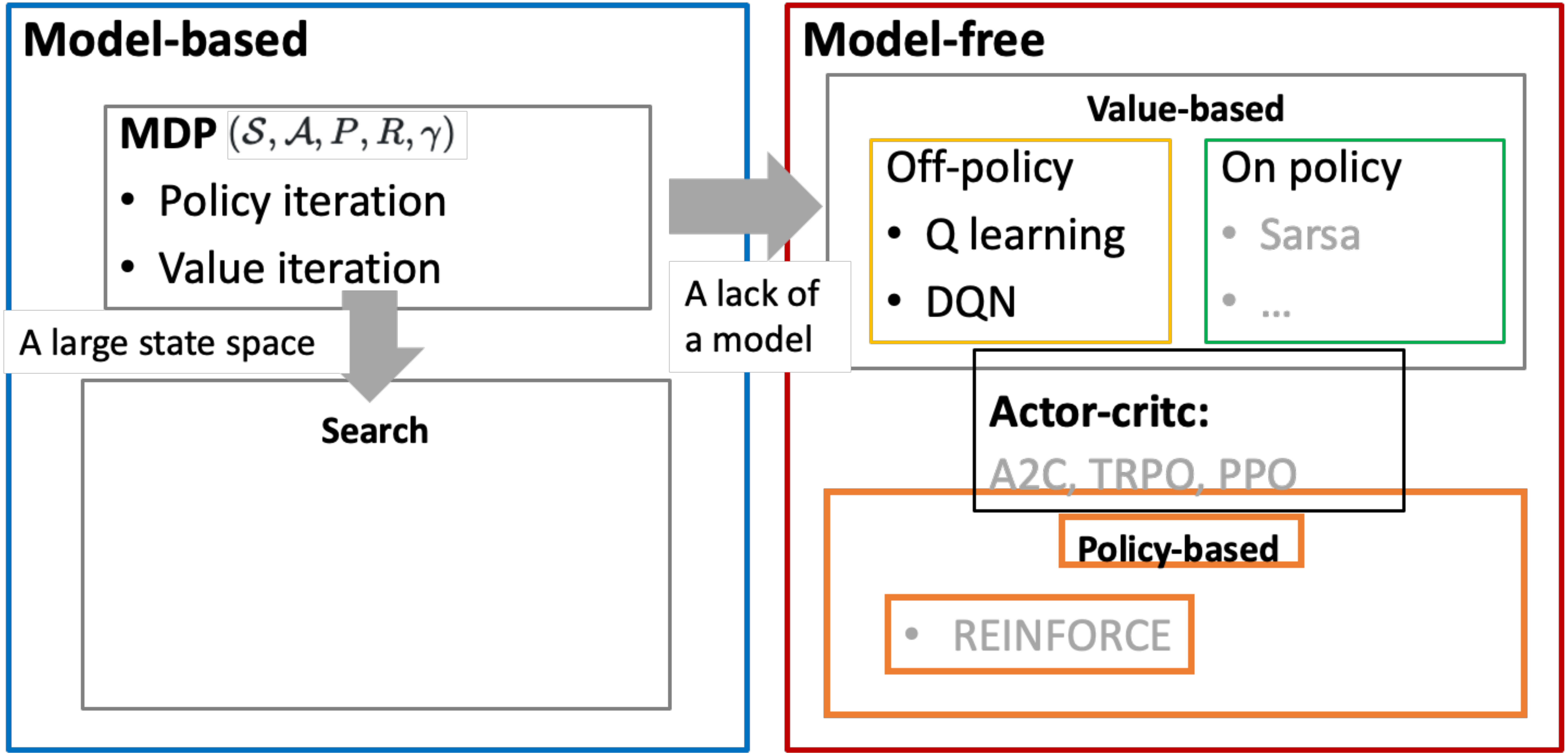
Reinforcement Learning 4

zjy, 2023/6/28

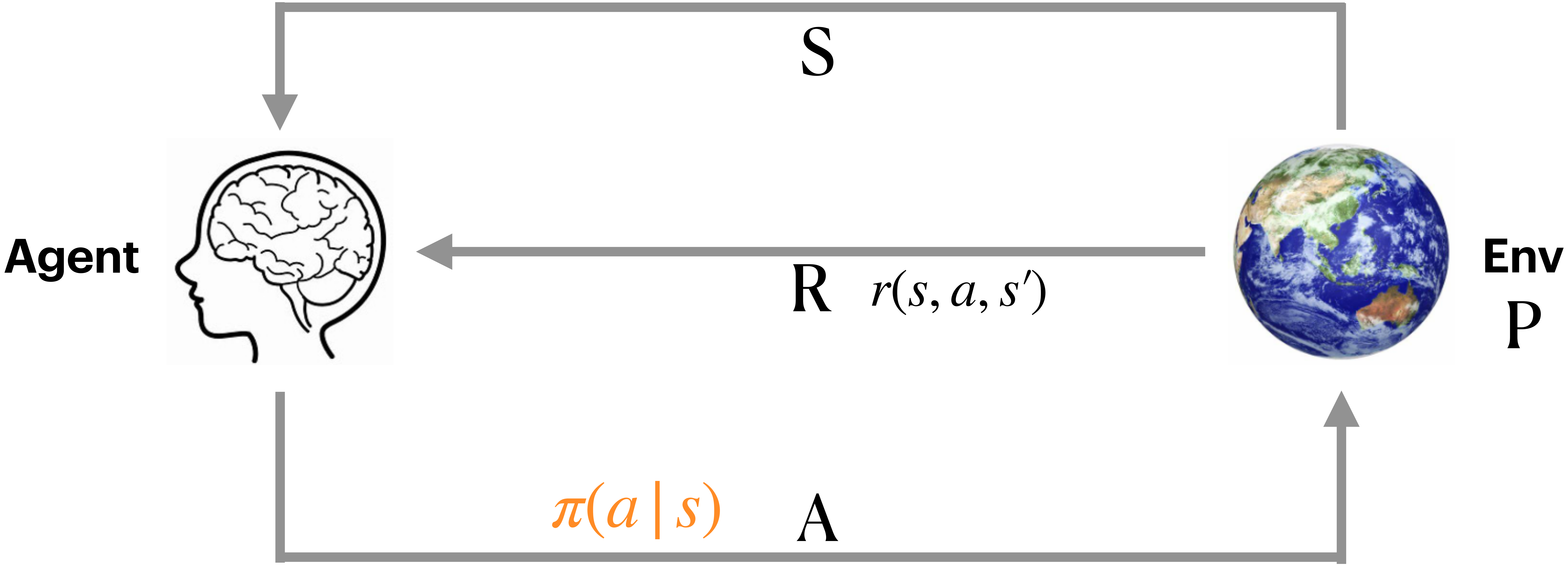
Syllabus

- Recap
- Policy-based
- Policy-Gradient Theorem
- *REINFORCE*

I. Recap


Model-based
MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$
• Policy iteration
• Value iteration
A large state space
Search
A lack of a model
Model-free
Value-based
Off-policy
• Q learning
• DQN
On policy
• Sarsa
• ...
Actor-critc:
A2C, TRPO, PPO
Policy-based
• REINFORCE

I. Recap



$\mathrm{MDP} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$

$q(s, a)$ / $v(s)$

Value / Policy Iteration

I. Recap, model-based, value / policy

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $\cancel{V(s) \leftarrow \sum_{s',r} p(s', r | s, \pi(s))[r + \gamma V(s')]}$
   $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)[R(s') + \gamma V(s')]$
   Is it the right value?
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\cancel{\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r | s, a)[r + \gamma V(s')]}$
   $\boxed{\pi(s) \leftarrow \arg\max_a \sum_{s'} p(s'|s,a)[R(s') + \gamma V(s')]}$
   Is it the right policy?
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $\cancel{V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)[r + \gamma V(s')]}$
| $V(s) \leftarrow \max_a \sum_{s'} p(s'|s,a)[R(s') + \gamma V(s')]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

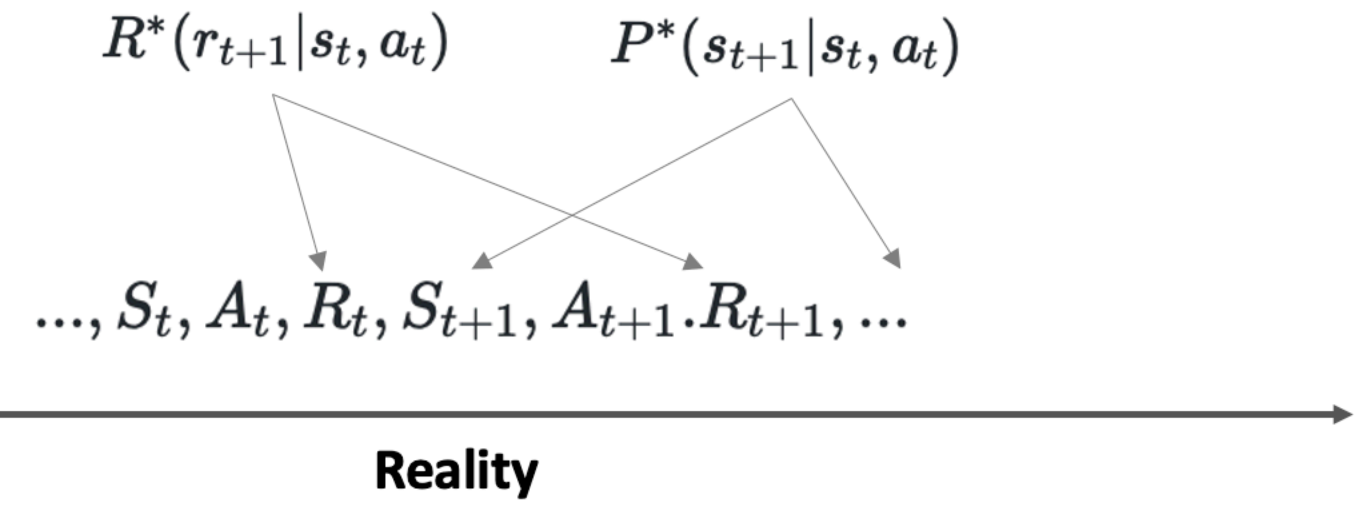
$\boxed{\text{Output a deterministic policy, } \pi \approx \pi_*, \text{ such that } \pi(s) = \arg\max_a \sum_{s',r} p(s', r | s, a)[r + \gamma V(s')]}$

- V / Q - Based on Bellman Equation, go over all states
- Diff: Policy $\pi(a \mid s)$

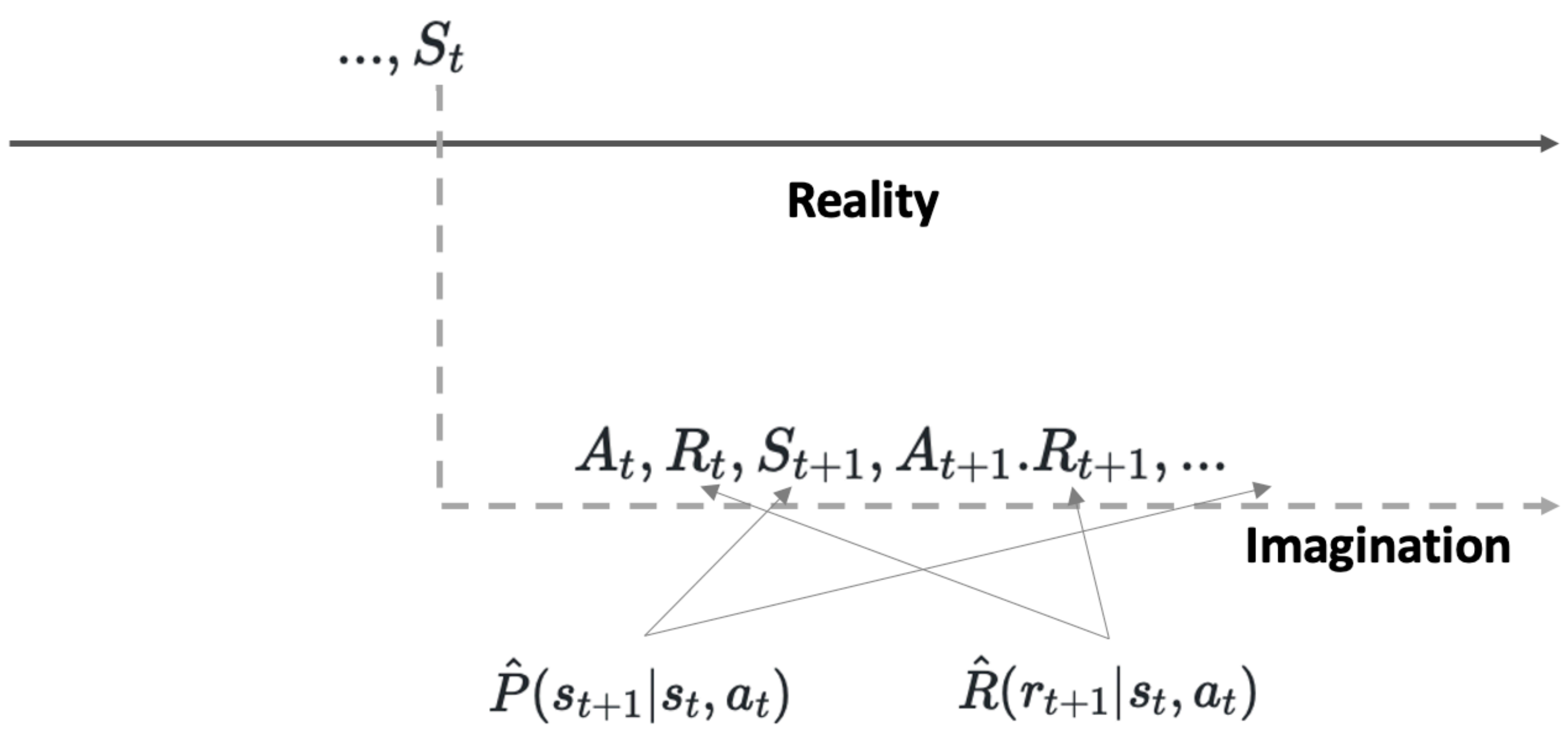
Planning & Searching

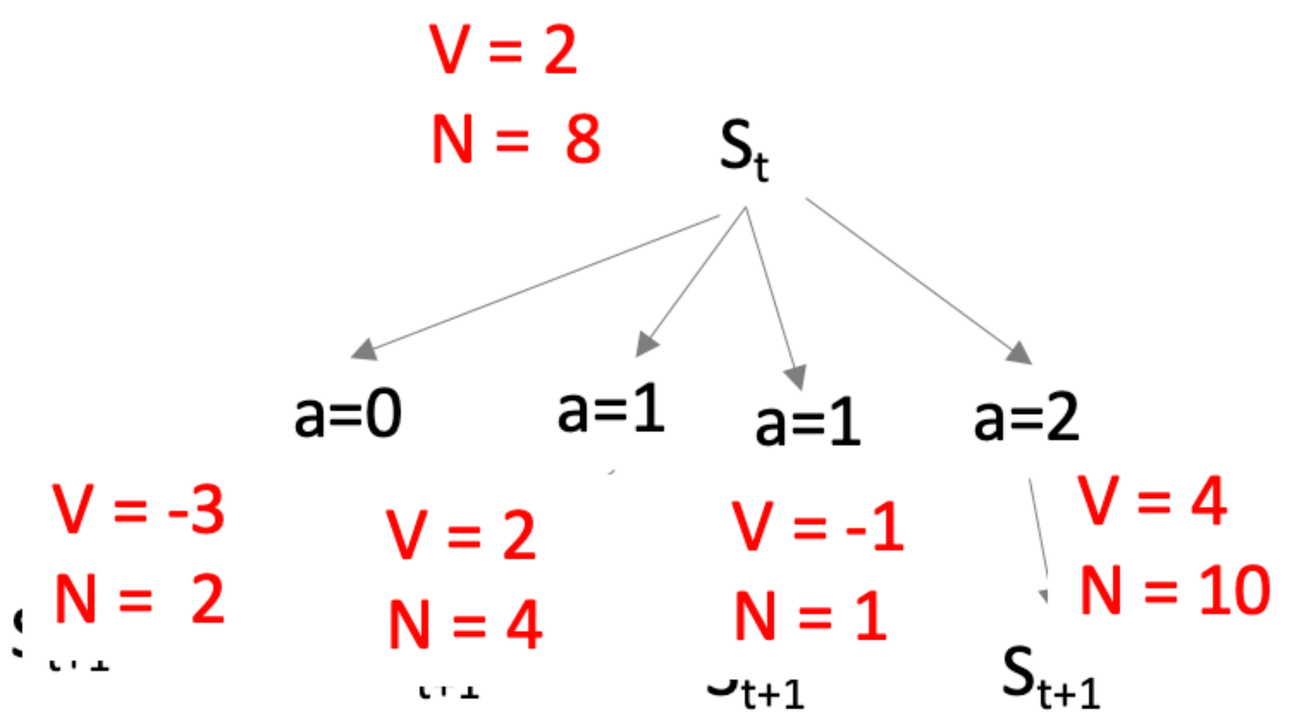
I. Recap, Model-based

Learning



Planning





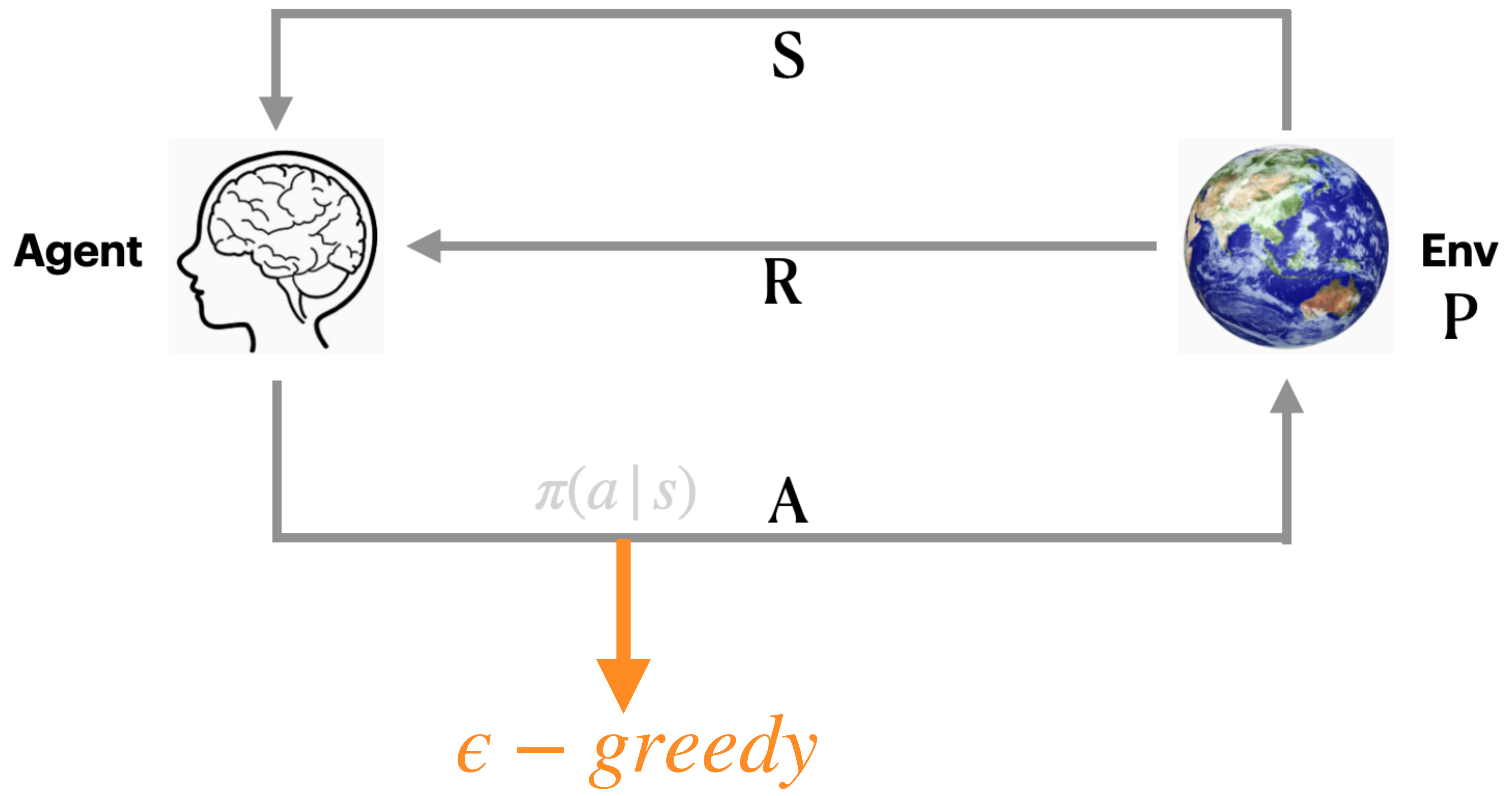
Choose the best action using UCT policy

$$\pi(a|s) = \frac{V(a)}{N(a)} + c\sqrt{\frac{2\ln N(s)}{N(a)}}$$

- Monte-Carlo Tree Search
  get V —> get a

Q-learning (DQN)

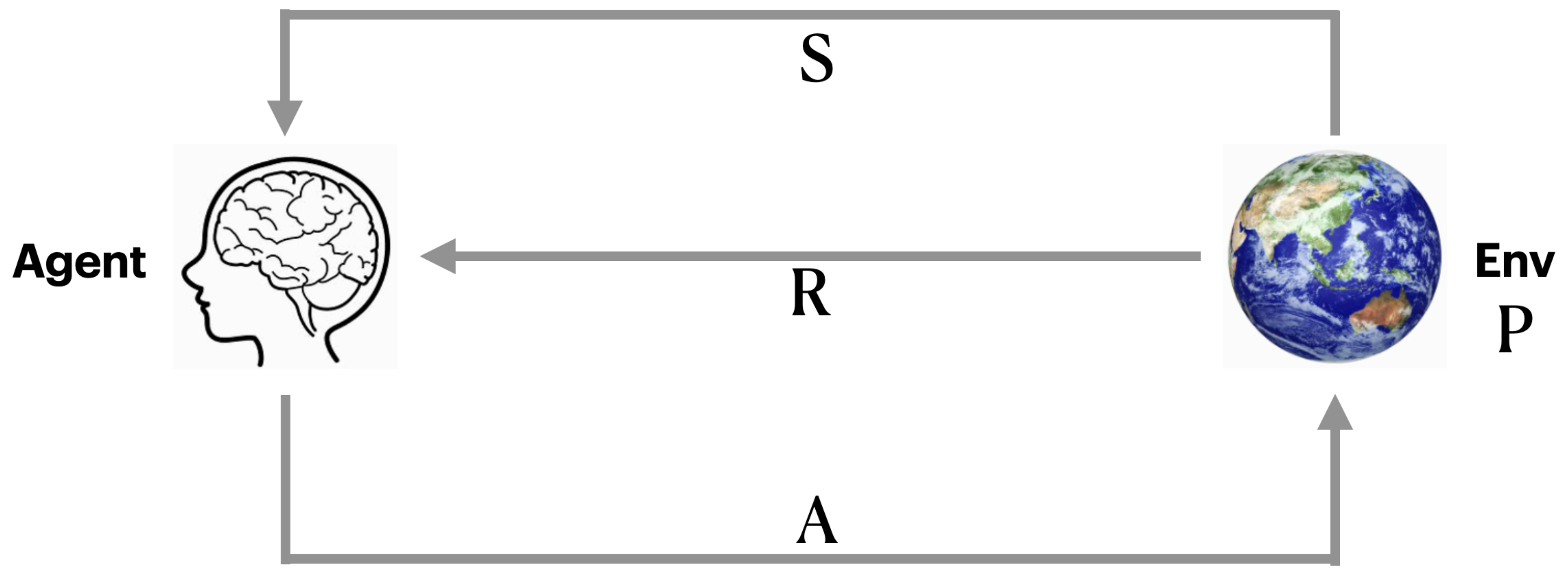
I. Recap, Model-free, Value-based



1. Experience Replay (get data)
   - Sample a_t from $Q(s_t, a; \theta)$, with ϵ-greedy
   - Store experience $e_t = (s_t, a_t, r_t, s_{t+1})$ in D
2. Update **Q-network**
   - [Input] state
   - [Output] q(s,a)

Policy Gradient

II. Model-free, Policy-based

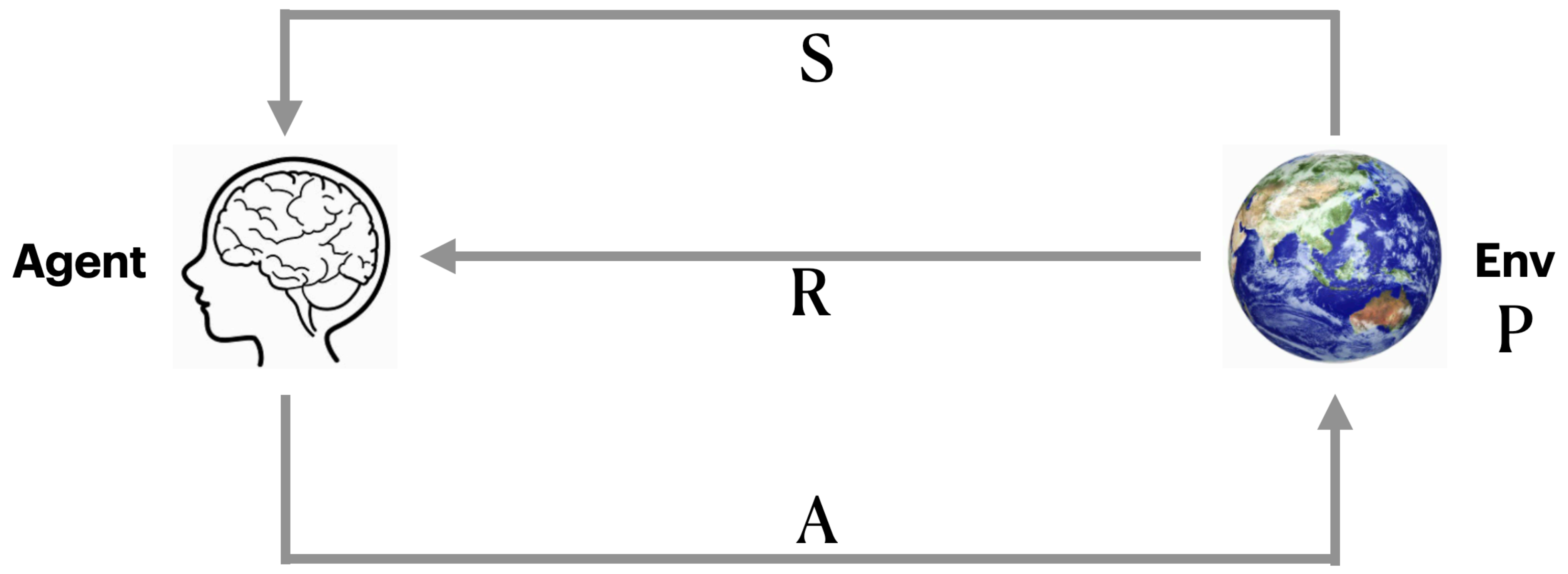


$$\pi(a \mid s, \theta)$$

- 之前的： *action-value* methods, learn V/Q
- 今天的： learn a *parameterized policy*,
  $\pi(a \mid s, \theta) = Pr\{A_t = a, S_t = s, \theta_t = \theta\}$
- If using learnt value func as well -> *actor-critic*

Policy Gradient

II. Model-free, Policy-based



$\pi(a \mid s, \theta)$

- 之前的： *action-value* methods, learn V/Q
- 今天的： learn a *parameterized policy*,
  $\pi(a \mid s, \theta) = Pr\{A_t = a, S_t = s, \theta_t = \theta\}$
- If using learnt value func as well -> *actor-critic*

- $\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$
- $J(\theta_t)$ — performance measure (just like loss function)

How to get $J(\theta_t)$?

Policy Approximation

II. Policy-based

How to parameterize policy $\pi(a \mid s, \theta)$?

- Soft-max in action preference: $$\pi(a \mid s, \theta) \doteq \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}}$$
- $h(s, a, \theta)$ - parameterized numerical preference
  - $h(s, a, \theta) = \theta^\top x(s, a)$, linear;
    $x(s, a)$, feature vectors
  - $h(s, a, \theta)$, ANN

Advantages

II. Policy-based

1. Deterministic Policy, according to soft-max distribution
   (ϵ-greedy is sometimes random)
2. selection of actions with arbitrary probabilities
   (A best approximate policy may be stochastic, if with imperfect information)
3. Policy may be a simpler function to approximate
4. Choice of policy parameterization, a good way of injecting prior knowledge

III. Policy Gradient Theorem

How to get $J(\theta_t)$?

- Define performance measure,
  $J(\theta) \doteq v_{\pi_\theta}(s_0)$, value of the start state of the episode

Proof of the Policy Gradient Theorem (episodic case)

With just elementary calculus and re-arranging of terms, we can prove the policy gradient theorem from first principles. To keep the notation simple, we leave it implicit in all cases that π is a function of $\boldsymbol{\theta}$, and all gradients are also implicitly with respect to $\boldsymbol{\theta}$. First note that the gradient of the state-value function can be written in terms of the action-value function as

$$
\begin{aligned}
\boxed{\nabla v_\pi(s) = \nabla \left[\sum_a \pi(a|s) q_\pi(s,a)\right],} \quad & \text{for all } s \in \mathcal{S} \qquad \text{(Exercise 3.18)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a)\Big] \quad \text{(product rule of calculus)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla \sum_{s',r} p(s',r|s,a)\big(r + v_\pi(s')\big)\Big] \\
& \qquad \text{(Exercise 3.19 and Equation 3.2)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \nabla v_\pi(s')\Big] \quad \text{(Eq. 3.4)} \\
&= \sum_a \Big[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \sum_{s'} p(s'|s,a) \quad \text{(unrolling)} \\
& \qquad \sum_{a'} \big[\nabla \pi(a'|s') q_\pi(s',a') + \pi(a'|s') \sum_{s''} p(s''|s',a') \nabla v_\pi(s'')\big]\Big] \\
&\boxed{= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \Pr(s \to x, k, \pi) \sum_a \nabla \pi(a|x) q_\pi(x,a),}
\end{aligned}
$$

after repeated unrolling, where $\Pr(s \to x, k, \pi)$ is the probability of transitioning from state s to state x in k steps under policy π. It is then immediate that

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \Pr(s_0 \to s, k, \pi)\right) \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_s \eta(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) \qquad \text{(box page 199)} \\
&= \sum_{s'} \eta(s') \sum_s \frac{\eta(s)}{\sum_{s'} \eta(s')} \sum_a \nabla \pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta(s') \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a) \qquad \text{(Eq. 9.3)} \\
&\boxed{\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)} \qquad \text{(Q.E.D.)}
\end{aligned}
$$

The on-policy distribution in episodic tasks

In an episodic task, the on-policy distribution is a little different in that it depends on how the initial states of episodes are chosen. Let $h(s)$ denote the probability that an episode begins in each state s, and let $\eta(s)$ denote the number of time steps spent, on average, in state s in a single episode. Time is spent in a state s if episodes start in s, or if transitions are made into s from a preceding state $\bar{s}$ in which time is spent:

$$\eta(s) = h(s) + \sum_{\bar{s}} \eta(\bar{s}) \sum_{a} \pi(a|\bar{s}) p(s|\bar{s}, a), \quad \text{for all } s \in \mathcal{S}. \tag{9.2}$$

This system of equations can be solved for the expected number of visits $\eta(s)$. The on-policy distribution is then the fraction of time spent in each state normalized to sum to one:

$$\boxed{\mu(s) = \frac{\eta(s)}{\sum_{s'} \eta(s')},} \quad \text{for all } s \in \mathcal{S}. \tag{9.3}$$

This is the natural choice without discounting. If there is discounting ($\gamma < 1$) it should be treated as a form of termination, which can be done simply by including a factor of γ in the second term of (9.2).

III. Policy Gradient Theorem

How to get $J(\theta_t)$?

- Define *<u>performance measure</u>*,

  $J(\theta) \doteq v_{\pi_\theta}(s_0)$, value of the start state of the episode

$$\boxed{\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a \mid s, \theta)}$$

- $\mu(s)$ — on-policy distribution under π *(prob of staying in state S in 1 episode)*

IV. REINFORCE

Monte Carlo Policy Gradient; 变换 -> putting samples in

Step 1. Introducing S_t

All-action method

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &\propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\boldsymbol{\theta}) \\
&= \mathbb{E}_\pi\left[\sum_a q_\pi(S_t,a) \nabla \pi(a|S_t,\boldsymbol{\theta})\right].
\end{aligned}
$$

$$
\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \sum_a \hat{q}(S_t,a,\mathbf{w}) \nabla \pi(a|S_t,\boldsymbol{\theta}),
$$

IV. *REINFORCE*

Monte Carlo Policy Gradient; 变换 -> putting samples in

Step 1. Introducing S_t

All-action method

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\boldsymbol{\theta})$$
$$= \mathbb{E}_\pi \left[\sum_a q_\pi(S_t,a) \nabla \pi(a|S_t,\boldsymbol{\theta}) \right].$$

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \sum_a \hat{q}(S_t,a,\mathbf{w}) \nabla \pi(a|S_t,\boldsymbol{\theta}),$$

Step 2. Introducing A_t

REINFORCE

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_\pi \left[\sum_a \pi(a|S_t,\boldsymbol{\theta}) q_\pi(S_t,a) \frac{\nabla \pi(a|S_t,\boldsymbol{\theta})}{\pi(a|S_t,\boldsymbol{\theta})} \right]$$
$$= \mathbb{E}_\pi \left[q_\pi(S_t,A_t) \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right] \qquad \text{(replacing } a \text{ by the sample } A_t \sim \pi)$$
$$= \mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right], \qquad \text{(because } \mathbb{E}_\pi[G_t|S_t,A_t] = q_\pi(S_t,A_t))$$

$$\boxed{\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}.}$$

IV. *REINFORCE*

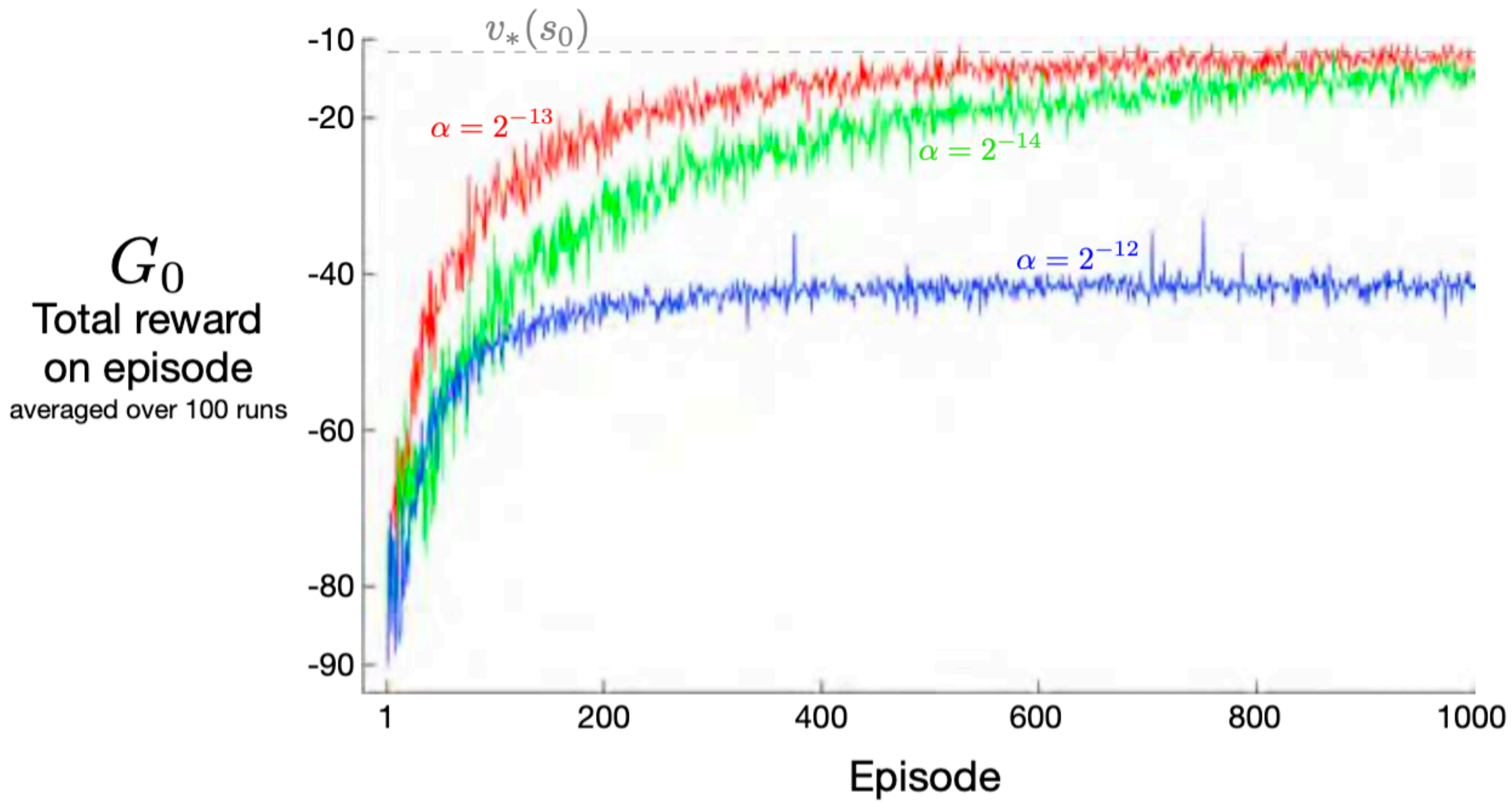
About $J(\theta)$

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}.$$

- $\nabla \pi(A_t | S_t, \theta_t)$ — ***direction*** most increases prob of repeating A_t
- G_t — Proportional to the ***return***
  favor actions that yield highest return
- $\pi(A_t | S_t, \theta_t)$ — Inversely proportional to the action probability
  otherwise, ***frequently-selected*** actions are at an advantage

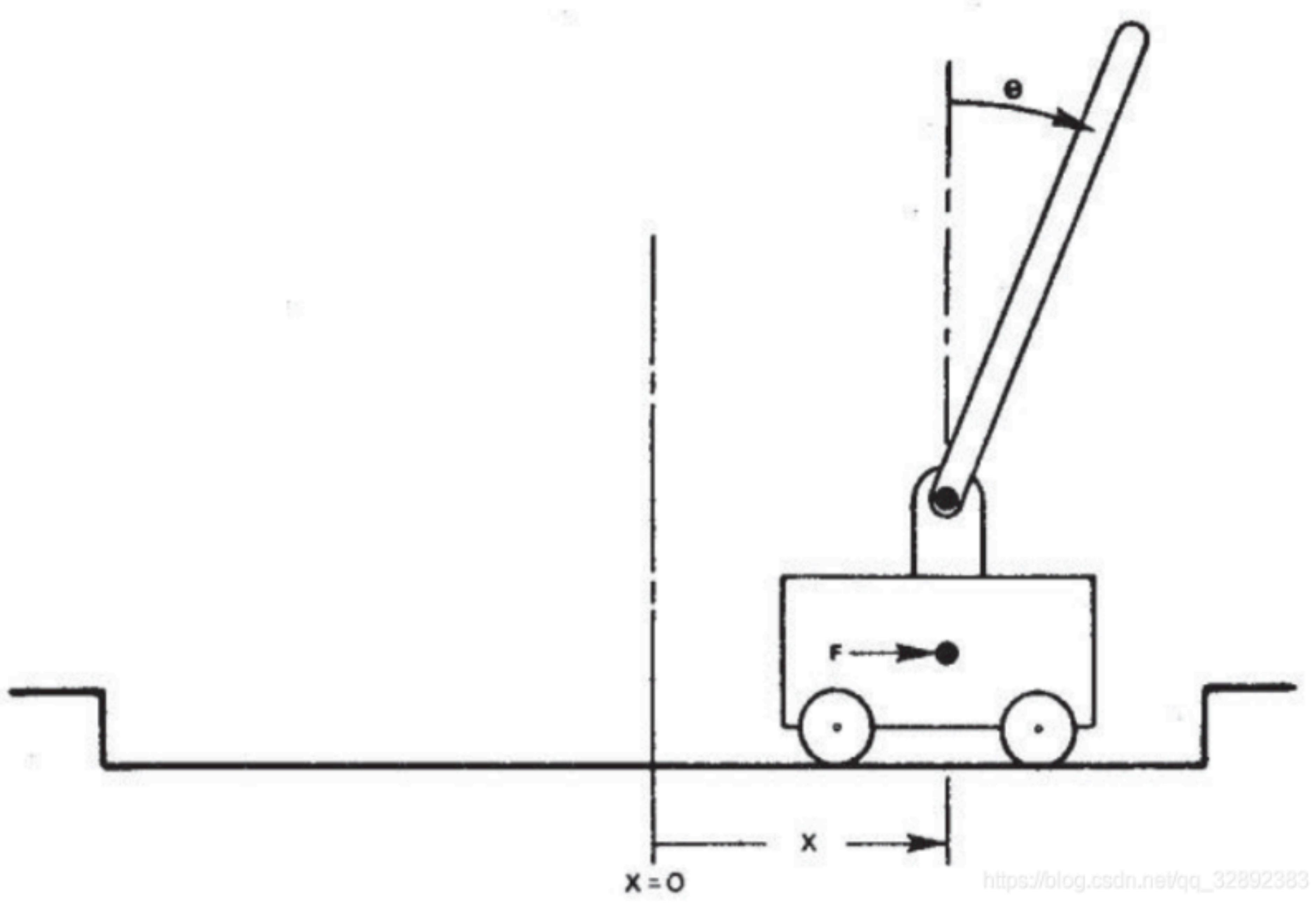
IV. *REINFORCE*

Highly depend on hyperparameters...



Coding - Task: Cartpole

IV. *REINFORCE*



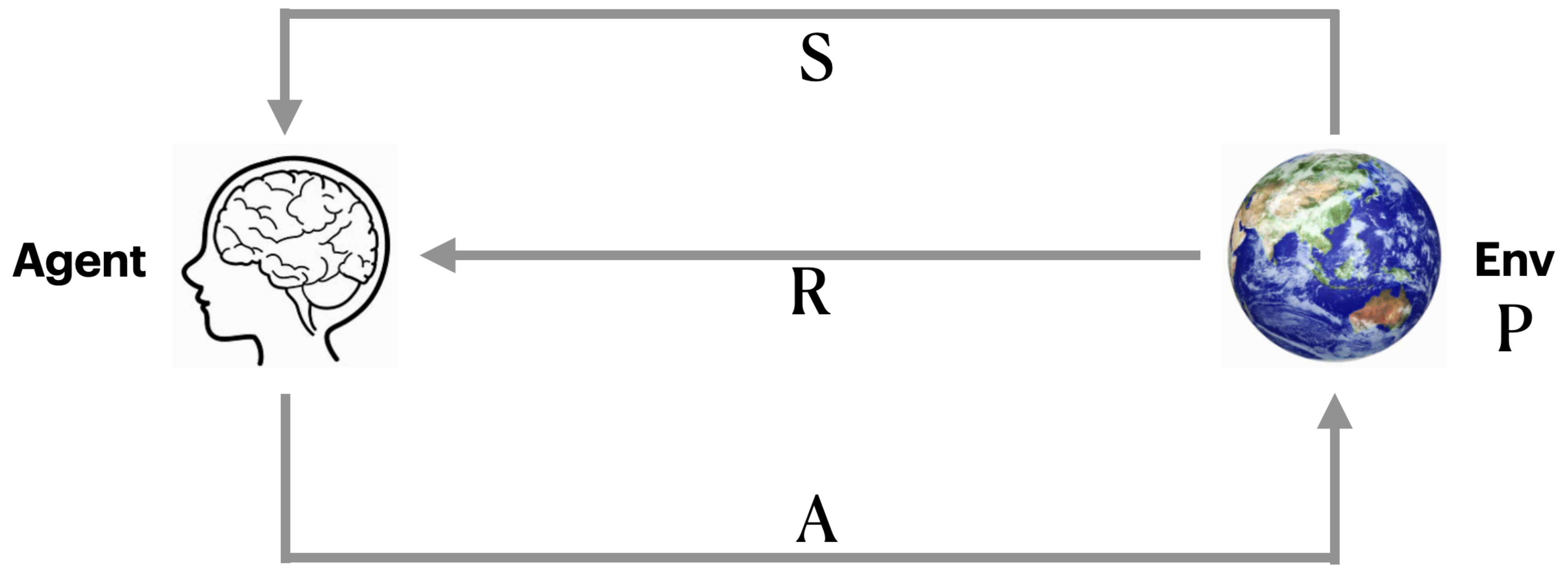
- State
  $(x, \theta, \dot{x}, \dot{\theta})$
- Action
  (0,1) left / right

Coding - Policy Network

IV. *REINFORCE, Cartpole*

$$\pi(a \mid s, \theta) \doteq \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}}$$

- [INPUT] State
- [OUTPUT]



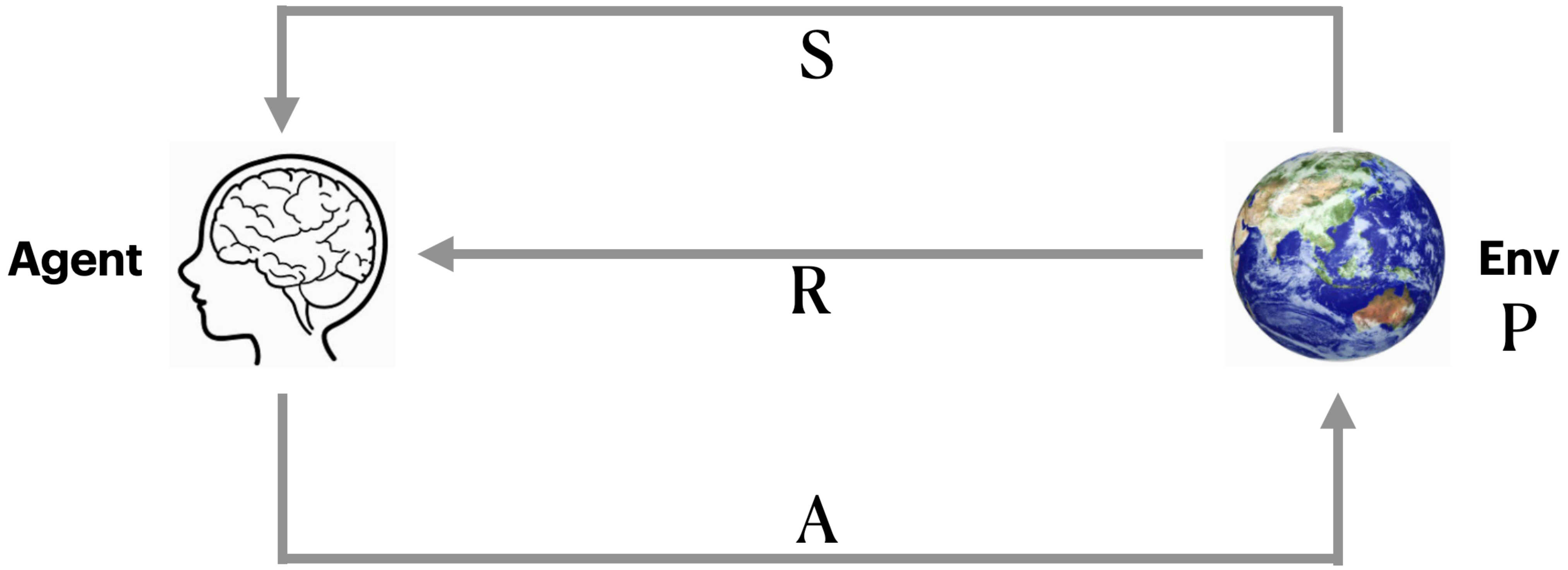
$PolicyNetwork,\ \pi(a \mid s, \theta)$

Coding - Policy Network

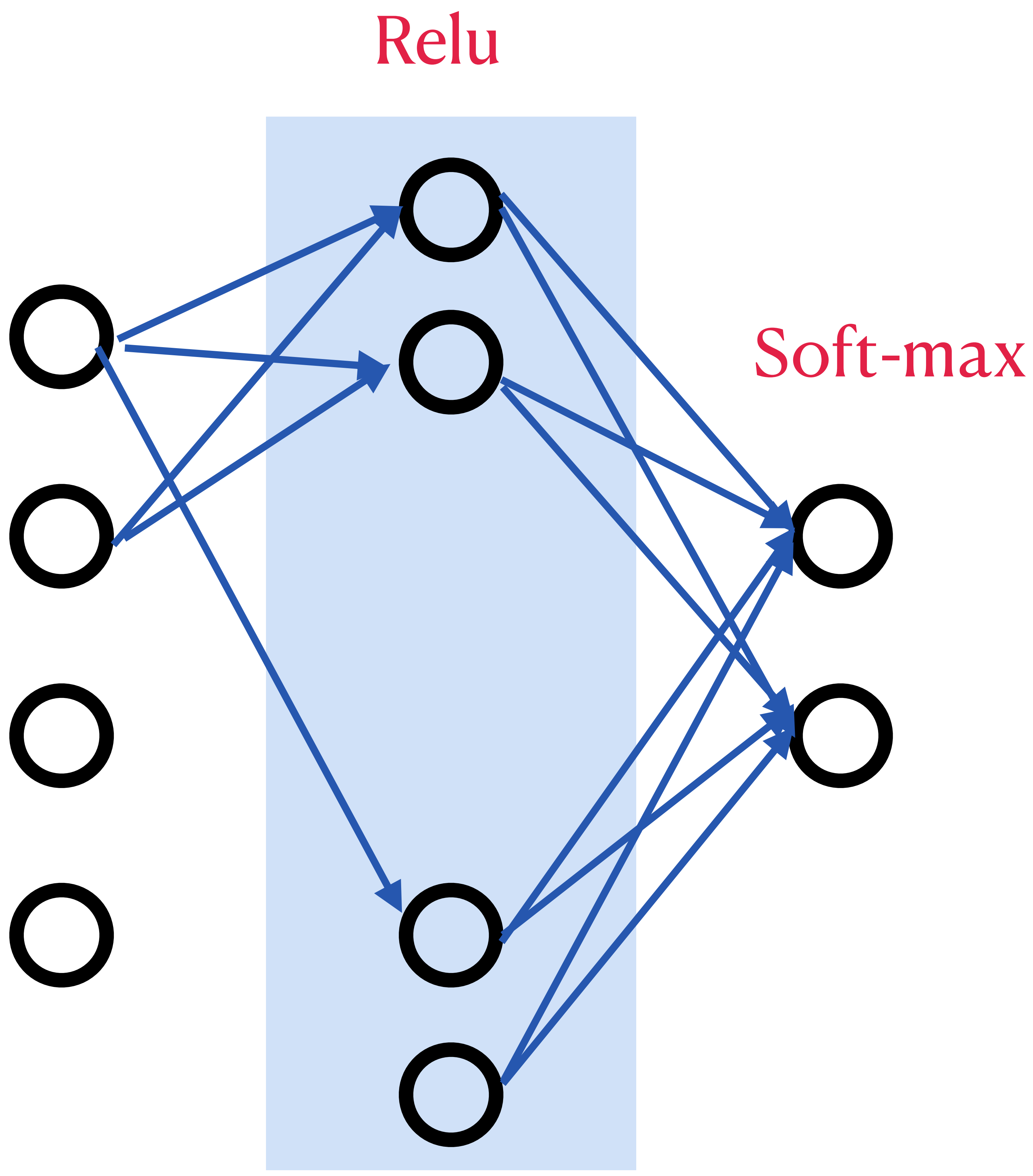
IV. *REINFORCE, Cartpole*

$$\pi(a \mid s, \theta) \doteq \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}}$$

- [INPUT] State, $x, \theta, \dot{x}, \dot{\theta}$
- [OUTPUT] $\pi(s, a)$



PolicyNetwork, $\pi(a \mid s, \theta)$



Coding

IV. *REINFORCE*

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot,\boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\gamma^t G \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})$

$$\pi(a\,|\,s,\theta) \doteq \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,a,\theta)}}$$

$h(a,s,\theta)$ — ANN

I. Define soft-max in action preference (Policy)

II. Sample

III. While updating θ

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}.$$

$$\nabla \ln \pi(A_t\,|\,S_t,\theta_t) = \frac{\nabla \pi(A_t\,|\,S_t,\theta_t)}{\pi(A_t\,|\,S_t,\theta_t)}$$

Other Stuff...

Actor-critic

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha \boxed{\Big(G_{t:t+1} - \hat{v}(S_t,\mathbf{w})\Big)} \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha \Big(R_{t+1} + \gamma \hat{v}(S_{t+1},\mathbf{w}) - \hat{v}(S_t,\mathbf{w})\Big) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha \delta_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta}_t)}{\pi(A_t|S_t, \boldsymbol{\theta}_t)}.
\end{aligned}$$