HSLU Lucerne University of Applied Sciences and Arts

Policy Gradient Methods

go right

Wait, I am still calculating Q-values.....

Reinforcement Learning
December 1, 2022

Learning Objectives

- Define policies as parametrized functions
- Understand the advantages (and disadvantages) of parametrized policies over action-value based methods
- Understand the objective function for policy gradient methods
- Understand the policy-gradient theorem
- Describe the actor-critic algorithm for control with function approximation

Policy Gradient Methods

- So far, almost all methods have been *action-value methods*
- Policies were only calculated from those action-value estimates (using GPI)

- We now turn to methods that directly learn a parametrized policy

$$\pi(a|s, \boldsymbol{\theta}) = \Pr\{A_t = a | S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$$

  where $\boldsymbol{\theta}$ are the parameters (weights)
- (If we also need a parametrized value function, we will use $\mathbf{w}$ for its parameters to distinguish between the two function approximations)

Constraints

The policy should be a probability over the different actions and must use exploration, therefor:

- The probability of any action should be greater than 0:

$$\pi(a|s, \boldsymbol{\theta}) > 0, \quad \text{for all } a \in \mathcal{A}, s \in \mathcal{S}$$

- The sum of all probabilities must be 1:

$$\sum_a \pi(a|s, \boldsymbol{\theta}) = 1, \quad \text{for all } s \in \mathcal{S}$$

Softmax for action preferences

- One common possibility to ensure those constraints is to use parameterized action-preferences:

$$h(s, a, \boldsymbol{\theta}) \in \mathbb{R}$$

- and then compute action probabilities using the softmax function:

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

Advantages of Policy Parametrization

A parametrized policy

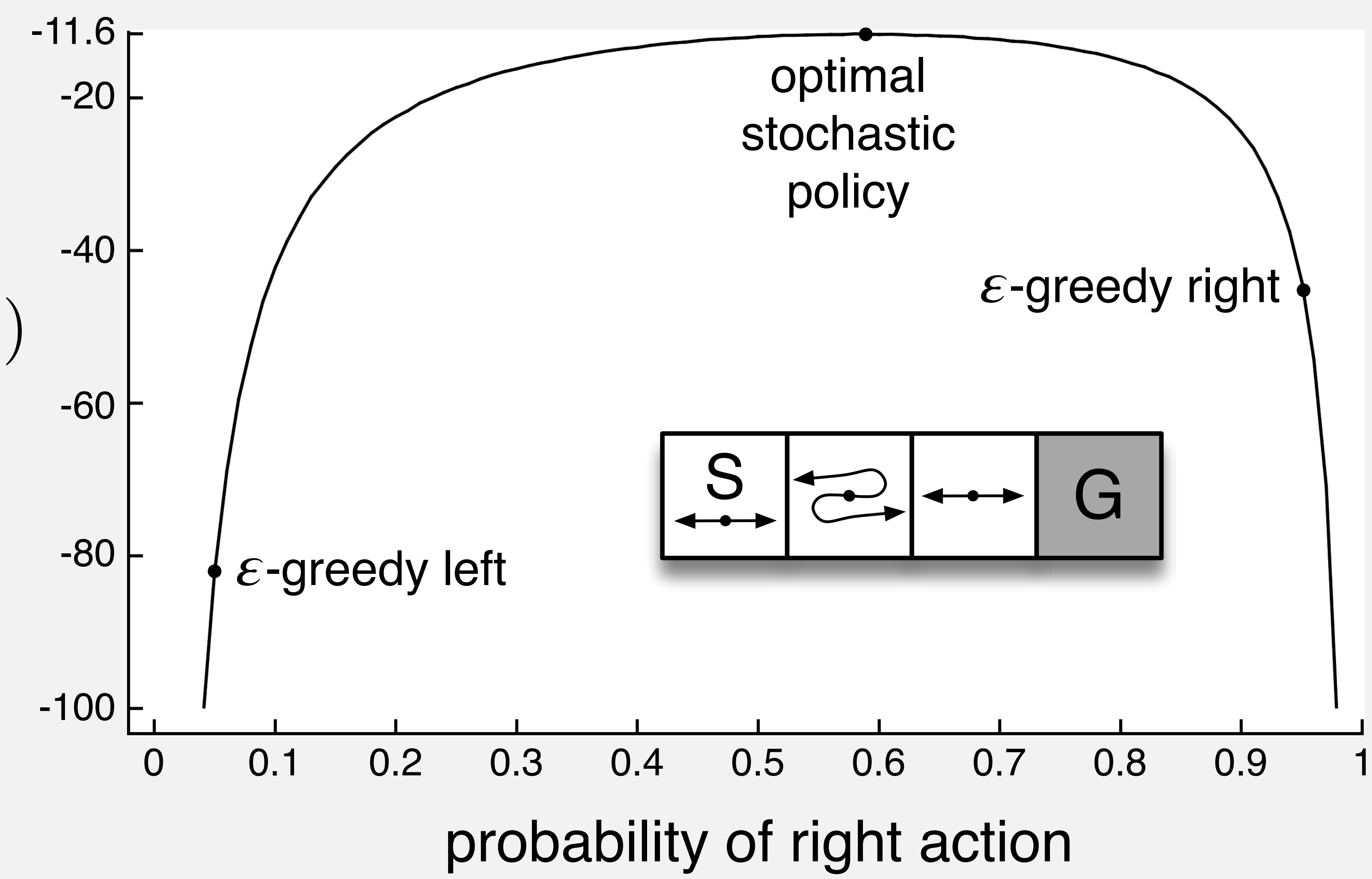
- can approach a deterministic policy over time (in comparison to epsilon-greedy, which will always explore)
- can model stochastic policies

Example for stochastic policy:

Small stochastic corridor:

- Reward = -1 for all steps
- Actions are left / right
- Second state is reversed: if the action is left it will go right
- All states appear identical under function approximation

$$J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(\mathsf{S})$$



Episodic case

- Goal: optimize the total return from a (particular) state

$$J(\boldsymbol{\theta}) \doteq v_{\pi_{\boldsymbol{\theta}}}(s_0)$$

- Problem: v depends on state distribution
- Policy gradient theorem (see book for derivation):

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

Policy Gradient

REINFORCE: Monte Carlo Policy Gradient

$$
\begin{aligned}
\nabla J(\boldsymbol{\theta}) &\propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\boldsymbol{\theta}) \\
&= \mathbb{E}_\pi\left[\sum_a q_\pi(S_t,a) \nabla \pi(a|S_t,\boldsymbol{\theta})\right] \\
&= \mathbb{E}_\pi\left[\sum_a \pi(a|s,\boldsymbol{\theta}) q_\pi(S_t,a) \frac{\nabla \pi(a|S_t,\boldsymbol{\theta})}{\pi(a|s,\boldsymbol{\theta})}\right] \\
&= \mathbb{E}_\pi\left[q_\pi(S_t,A_t) \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|s,\boldsymbol{\theta})}\right], \quad \text{replacing } a \text{ by a sample } A_t \\
&= \mathbb{E}_\pi\left[G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|s,\boldsymbol{\theta})}\right], \quad \text{because } \mathbb{E}_\pi[G_t|S_t,A_t] = q_\pi(S_t,A_t)
\end{aligned}
$$

REINFORCE: Monte Carlo Policy Gradient

- Update the weights according to:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|s,\boldsymbol{\theta})} \\ &= \boldsymbol{\theta}_t + \alpha G_t \nabla \ln \pi(A_t|S_t,\boldsymbol{\theta})\end{aligned}$$

- The second line can be derived from

$$\nabla \ln(f(x)) = \frac{\nabla f(x)}{f(x)}$$

- This is gradient **ascent**, as we want to maximize the return

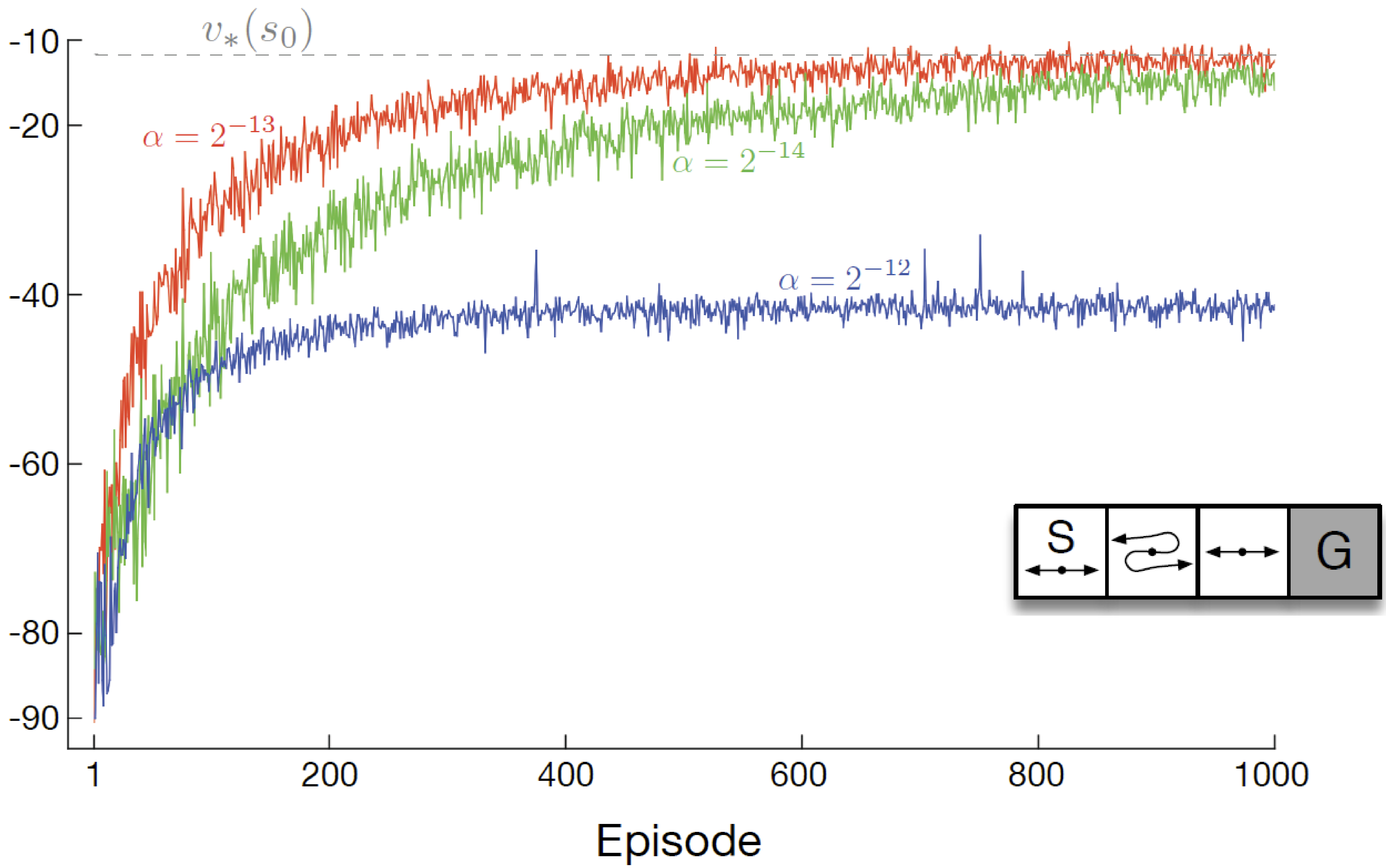
REINFORCE

REINFORCE: MC Policy-Gradient Control (episodic)

Input:
  a differentiable policy parameterization $\pi(a|w, \boldsymbol{\theta})$
  step size $\alpha > 0$
Initialize:
  policy parameters $\boldsymbol{\theta}$

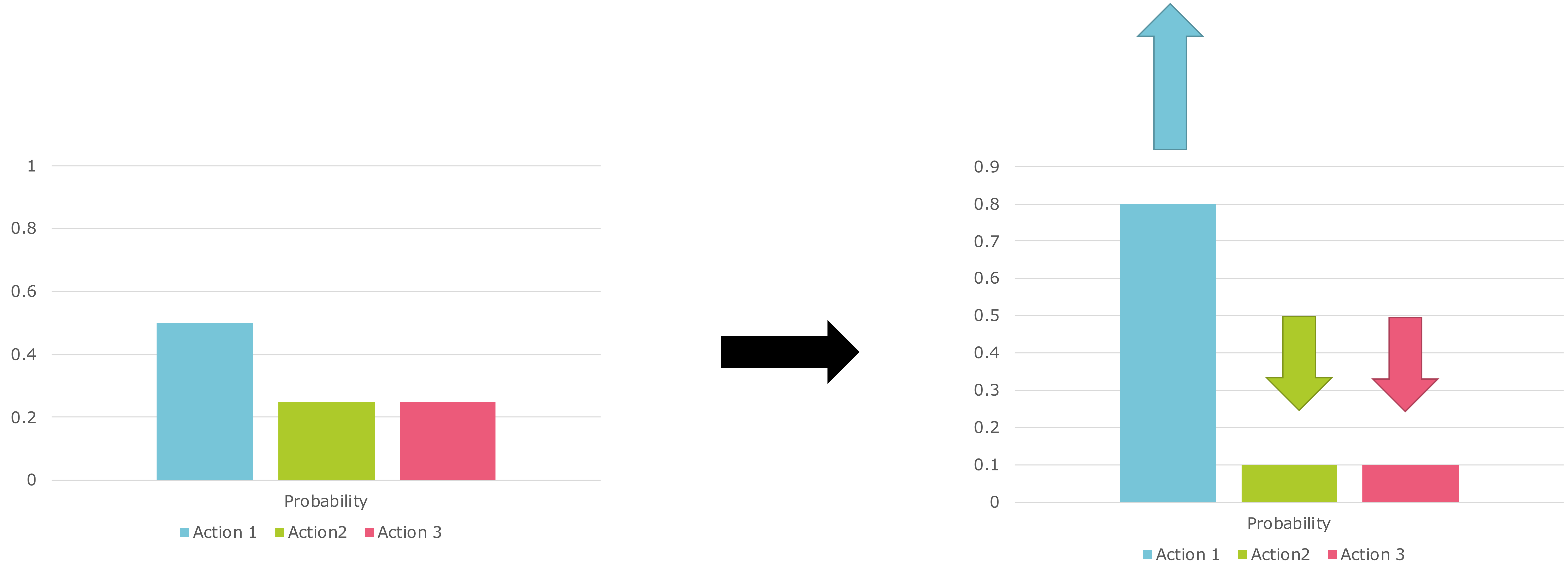
Loop for each episode:
  Generate an episode following π: $S_0, A_0, R_0, S_1, ..., R_T$
  For every step $t = 0, T$ in the episode:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Example


$v_*(s_0)$
-10
-20
-40
-60
-80
-90
$\alpha = 2^{-13}$
$\alpha = 2^{-14}$
$\alpha = 2^{-12}$
S
G
1
200
400
600
800
1000
Episode

How does Policy Gradient work?

- The gradient will push the selected action to have a higher probability (at the expense of the others)
- The amount that it will be pushed depends on the return G



Baseline

A "trick" for faster convergence is to subtract a baseline from the q values, where the baseline can be any function that does not depend on the action

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a (q_\pi(s,a) - b(s)) \nabla \pi(a|s,\boldsymbol{\theta})$$

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

Baseline

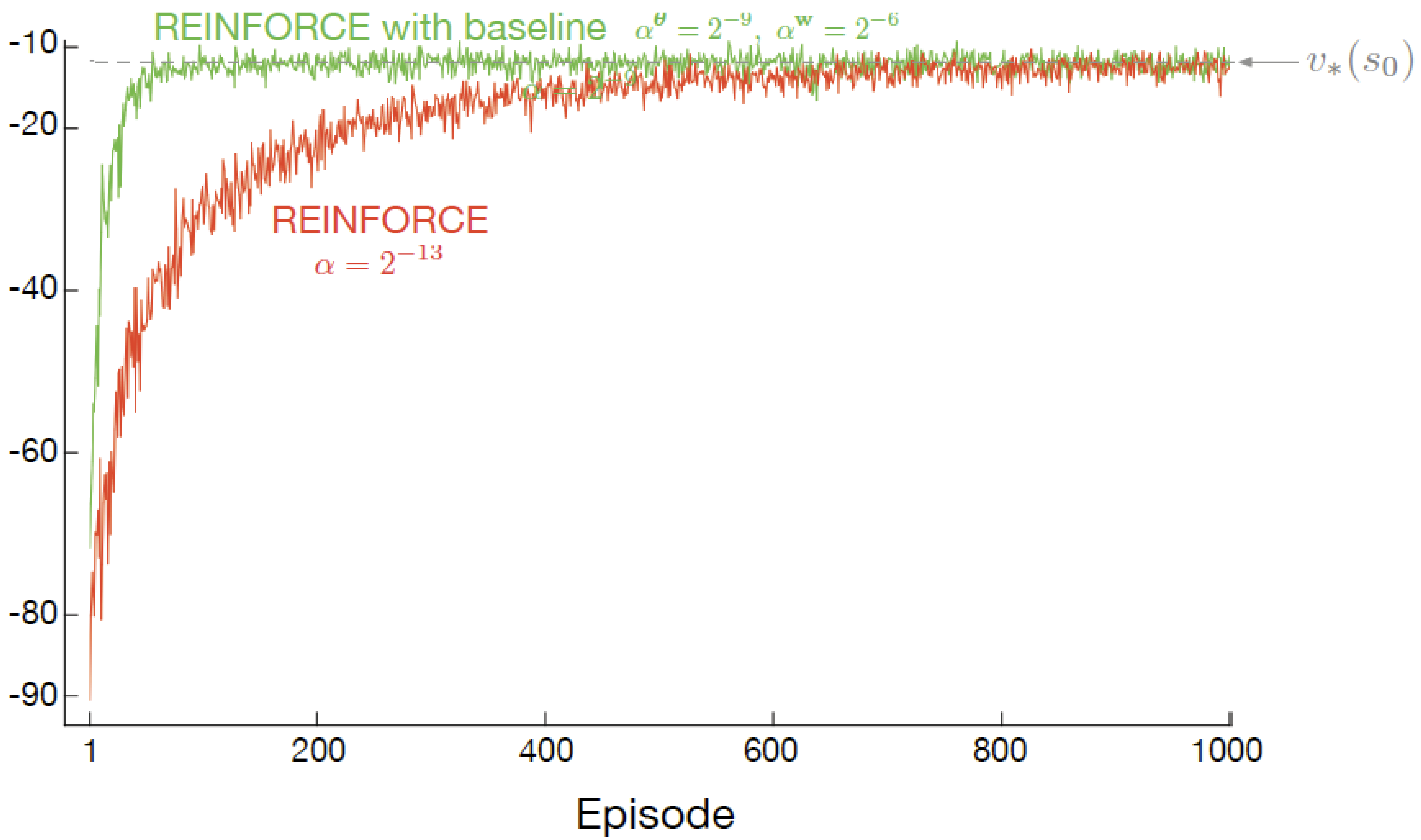
- A natural choice for b, is to estimate a state value

$$\hat{v}(S_t, \mathbf{w})$$

- where the weights would also be learned using the Monte-Carlo method
- The function is not dependent on the policy parametrization, so the gradient remains the same
- The difference between q and v is also called the *advantage* function:

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

REINFORCE with Baseline



Actor Critic: Policy Gradient Method with Critic

- We would like to implement a 1-step (or n-step) method like TD(0) for the policy
- However, we then need a **value** function, so we can replace the full return in REINFORCE with the 1-step return:

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha \left(R_{t+1} + \gamma \hat{v}(S_{t+1}, \mathbf{w}) - \hat{v}(S_t, \mathbf{w})\right) \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t) \\
&= \boldsymbol{\theta}_t + \alpha \, \delta_t \, \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta}_t)
\end{aligned}
$$

- The policy is called the *actor*, the value function the *critic*

One step Actor-Critic (episodic) for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input:

a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$

a differentiable state-value function parametrization $\hat{v}(s, \mathbf{w})$

step sizes $\alpha^{\boldsymbol{\theta}} > 0, \alpha^{\mathbf{w}} > 0$

Initialize:

policy parameters $\boldsymbol{\theta}$ and state-value weights $\mathbf{w}$

Loop for each episode:

Initialize S, first state of episode

$I \leftarrow 1$

For each time step of the episode:

Choose $A \sim \pi(\cdot|S, \boldsymbol{\theta})$

Take action A, observe S', R

$\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$

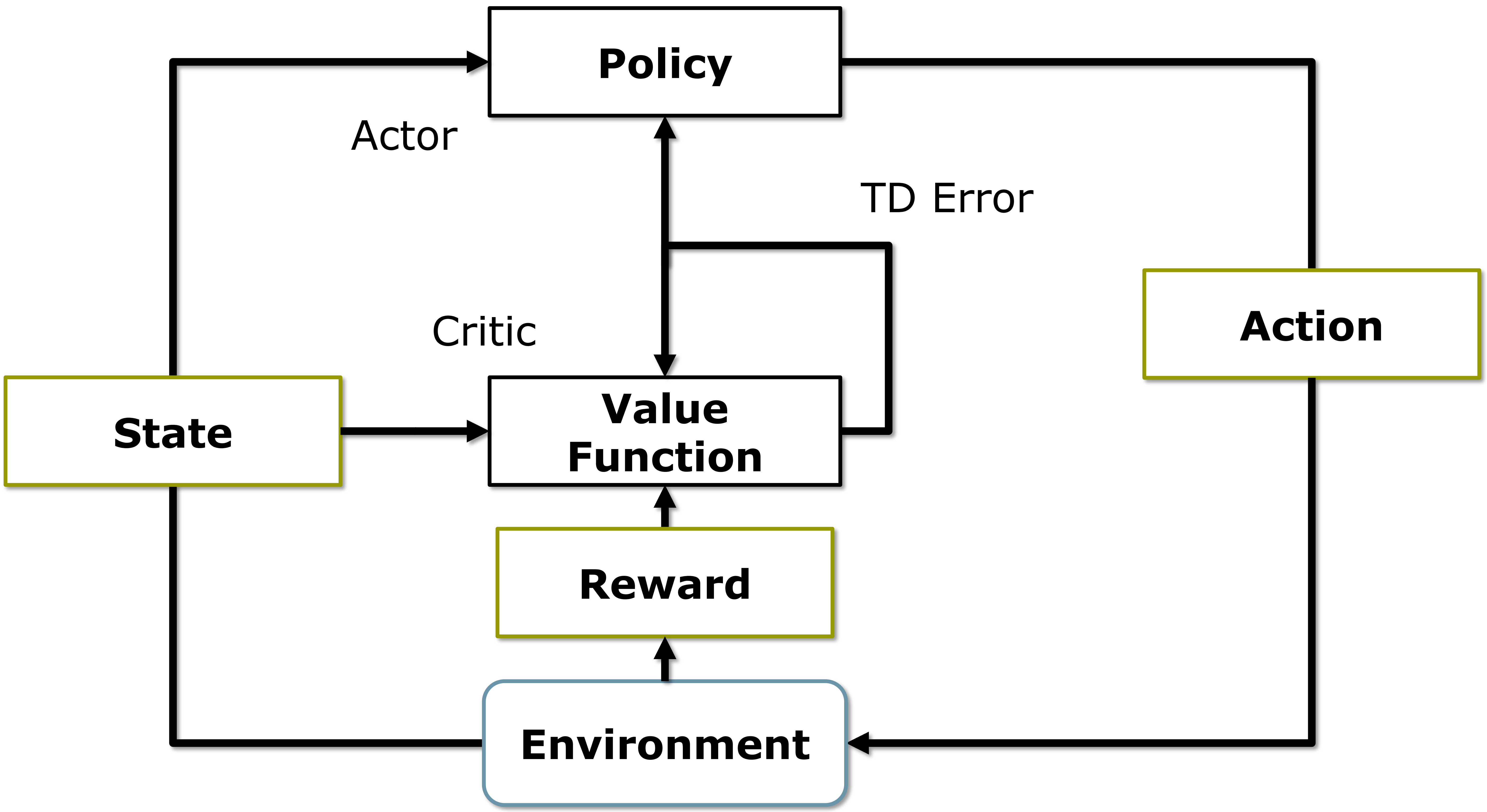
$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S, \mathbf{w})$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S, \boldsymbol{\theta})$

$I \leftarrow \gamma I$

$S \leftarrow S'$

Actor-Critic



Example: Flight Simulator Landing

Average Reward for Continuing Tasks

- In the function approximation approaches, the discounted returns can be problematic
- We can instead use average-rewards

$$
\begin{aligned}
r(\pi) &\doteq \lim_{h \to \infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi] \\
&= \lim_{h \to \infty} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi] \\
&= \sum_{s} \mu_\pi(s) \sum_{a} \pi(a|s) \sum_{s',r} p(s', r|s, a) r
\end{aligned}
$$

Average Rewards: Differential Settings

If we are using average rewards, the returns are defined in terms of the difference between the obtained rewards and the average reward:

$$G_t = \sum_{t}^{\infty} R_t - r(\pi)$$

Similarly, we can define the other value functions, for example for the *differential* state-value function, we get:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r - r(\pi) + v_\pi(s')]$$

One step Actor-Critic (continuing) for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input:

a differentiable policy parameterization $\pi(a|w, \boldsymbol{\theta})$

a differentiable state-value function parametrization $\hat{v}(s, \mathbf{w})$

Parameters: $\alpha^{\boldsymbol{\theta}} > 0, \alpha^{\mathbf{w}} > 0, \alpha^{\bar{R}} > 0$

Initialize:

$\bar{R} \in \mathbb{R}$, for example to 0

policy parameters $\boldsymbol{\theta}$ and state-value weights $\mathbf{w}$

Initialize$S \in \mathcal{S}$

Loop forever (for each time step):

Choose $A \sim \pi(\cdot|S, \boldsymbol{\theta})$

Take action A, observe S', R

$\delta \leftarrow R - \bar{R} + \gamma\hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$

$\bar{R} \leftarrow \bar{R} + \alpha^{\bar{R}}\delta$

$\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S, \mathbf{w})$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}}\delta\nabla \ln \pi(A|S, \boldsymbol{\theta})$

$S \leftarrow S'$