Soft Actor Critic (SAC) 강화학습 알고리즘 소개

Soohee Han

Pohang University of Science and Technology

soohee.han@postech.ac.kr

June 22, 2022

Overview

Recursive solving strategies

- Tower of Hanoi

$$\text{Hanoi}(N, \text{start}, \text{to}, \text{via}) = \begin{cases} \text{If N} == 1, \\ \quad \text{Move}(1, \text{start}, \text{to}) \\ \text{Else}, \\ \quad \text{Hanoi}(N-1, \text{start}, \text{via}, \text{to}) \\ \quad + \text{Move}(N, \text{start}, \text{to}) \\ \quad + \text{Hanoi}(N-1, \text{via}, \text{to}, \text{start}) \end{cases}$$

- Consider a function $f : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, n\}$. Compute the number of possible one to one correspondence (bijective, invertible) function such that $f(1) \neq 1$, $f(2) \neq 2$, $\cdots$, $f(n) \neq n$.

$$\text{num}(n) = (n-1)\{\text{num}(n-2) + \text{num}(n-1)\}$$

Recursive solving strategies

- Volume of a n-dimensional ball

$$V_n(R) = \int_{-R}^{R} V_{n-1}(\sqrt{R^2 - x^2})dx = \begin{cases} \text{n=2k ,} & \frac{\pi^k}{k!}R^{2k} \\ \text{n=2k+1,} & \frac{k!2^{2k+1}\pi^k}{(2k+1)!}R^{2k+1} \end{cases}$$

- Dynamic programming : https://wooder2050.medium.com

$$\begin{pmatrix} 3(S) & 7 & 9 & 2 & 7 \\ 9 & 8 & 3 & 5 & 5 \\ 1 & 7 & 9 & 8 & 5 \\ 3 & 8 & 6 & 4 & 10 \\ 6 & 3 & 9 & 7 & 8(G) \end{pmatrix}$$

$$\text{sum}(y, x) = \max(\text{sum}(y, x-1), \text{sum}(y-1, x)) + \text{value}(y, x)$$

Recursive solving strategies : contraction mapping

- Contraction : $f(\cdot)$ is a contraction, or a contraction mapping, if there is a real number k, $0 \leq k < 1$ such that

$$\|f(x) - f(y)\| \leq k\|x - y\|$$

  for all x and y.
- Fixed point theorem : If $f(\cdot)$ is a contraction, there is one and only one point x_0 such that $f(x_0) = x_0$.
- $x_2 = f(x_1)$, $x_3 = f(x_2)$, $\cdots$: $x_n \to x_0$ as n goes to ∞.
- Example :

$$\begin{aligned} R_{n+1} &= f(R_n) = \frac{R_n}{R_n + 1} + 2 \\ \longrightarrow \quad R_{n+1} - R_{m+1} &= f(R_n) - f(R_m) = \frac{R_n - R_m}{(R_n + 1)(R_m + 1)} \end{aligned}$$

Markov decision process

A Markov decisoin process is defined by 1 5-tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{R}$, γ)

- $\mathcal{S}$: A set of states
- $\mathcal{A}$: A set of actions
- $\mathcal{P}$: A state transition probability $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,\ 1]$
- $\mathcal{R}$: A reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Re$
- γ : A discount factor

Notations :

$s_t \to a_t \to r_t \to s_{t+1} \to a_{t+1}, \quad s \to a \to r \to s' \to a'$

$\tau = (s_0, a_0, s_1, a_1, \cdots, s_T, a_T)$

$P^a_{ss'} = P(S_{t+1} = s' | S_t = s, A_t = a)$

$P(s', r|s, a) = P(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a)$

$R^a_{ss'} = \mathbb{E}[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s']$

$R^a_s = \mathbb{E}[R_t | S_t = s, A_t = a] = \sum_{s' \in S} P^a_{ss'} R^a_{ss'}$

$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$: Return

Markov decision process

- State :

  The state of a system at time t_0 is the amount of information at t_0 that, together with input $u_{[t_0,\infty)}$ determines uniquely the behavior of the system for all $t \geq t_0$.

- Optimal control for $s_{t+1} = As_t + Ba_t$

  $a_t = -Ks_t$ (o)

  $a_t = -K_1 s_t - K_2 s_{t-1}$ (x)

  Cost function : $\sum_{t=t_0}^{\infty} [s_t^T Q s_t + u_t^T R u_t]$

- Optimal controls for $s_{t+1} = As_t + B_1 a_t + B_2 a_{t-1}$ and $s_{t+1} = A_1 s_t + A_2 s_{t-1} + Ba_t$

Value functions

- State value function : $\tau = (a_t, s_{t+1}, a_{t+1}, \cdots)$

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[G(t)|S_t = s_t] = \mathbb{E}_{\pi}\Big[\sum_{k=0} \gamma^k R_{t+k}|S_t = s_t\Big]$$

- State-action value function : $\tau = (s_{t+1}, a_{t+1}, s_{t+2}, \cdots)$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[\sum_{k=0} \gamma^k R_{t+k}|S_t = s_t, A_t = a_t]$$

- The relation between $V^{\pi}(s_t)$ and $Q^{\pi}(s_t, a_t)$

$$\begin{aligned} V^{\pi}(s_t) &= \int_{a_t \in \mathcal{A}} \pi(a_t|s_t) Q^{\pi}(s_t, a_t) da_t = \mathbb{E}_{a_t \sim \pi(a_t|s_t)}[Q^{\pi}(s_t, a_t)] \\ Q^{\pi}(s_t, a_t) &= R_{s_t}^{a_t} + \gamma \int_{s_{t+1} \in \mathcal{S}} P_{s_t s_{t+1}}^{a_t} V^{\pi}(s_{t+1}) ds_{t+1} \end{aligned}$$

$\longrightarrow$ Linear equations

Optimal value functions

- Optimal state value function

$$\begin{aligned} V^*(s) &= \max_\pi V^\pi(s) \\ \pi^* &= \arg\max_\pi V^\pi(s) \end{aligned}$$

  For all s and π, we have

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

- Optimal state-action value function

$$Q^*(s,a) = Q^{\pi^*}(s,a) \geq Q^\pi(s,a)$$

  for all s, a, and π.

- The relation between $V^*(s)$ and $Q^*(s,a)$

$$\begin{aligned} V^*(s) &= Q^*(s,\pi^*(s)) = \max_{a\in\mathcal{A}} Q^*(s,a) \\ Q^*(s,a) &= R_s^a + \gamma \int_{s'\in S} P_{ss'}^a V^*(s')ds' \end{aligned}$$

Bellman expectation equations

- Bellman expectation equation for $V^\pi(s)$

$$\begin{aligned} V^\pi(s) &= \int_{a\in\mathcal{A}} \pi(a|s) Q^\pi(s,a) da \\ &= \int_{a\in\mathcal{A}} \pi(a|s) \int_{s'\in\mathcal{S}} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi(s')] ds' da \\ &= \mathbb{E}_{a\sim\pi(a|s)}[R^a_s] + \mathbb{E}_{a\sim\pi(a|s),\ s'\sim P^a_{ss'}}[\gamma V^\pi(s') \mid s] \end{aligned}$$

- Bellman expectation equation for $Q^\pi(s,a)$

$$\begin{aligned} Q^\pi(s,a) &= \int_{s'\in\mathcal{S}} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi(s')] ds' \\ &= \int_{s'\in\mathcal{S}} P^a_{ss'}[R^a_{ss'} + \gamma \int_{a'\in\mathcal{A}} \pi(a'|s') Q^\pi(s',a') da'] ds' \\ &= R^a_s + \mathbb{E}_{s'\sim P^a_{ss'},\ a'\sim\pi(a'|s')}[\gamma Q^\pi(s',a') \mid s,a] \end{aligned}$$

Bellman optimality equations

- Bellman optimality equation for $V^*(s)$

$$\begin{aligned} V^*(s) &= \max_{a \in \mathcal{A}} Q^*(s,a) = \max_{a \in \mathcal{A}} \left[R_s^a + \gamma \int_{s' \in S} P_{ss'}^a V^*(s') ds' \right] \\ &= \max_{a \in \mathcal{A}} \left[R_s^a + \mathbb{E}_{s' \sim P_{ss'}^a} \left[\gamma V^*(s') \right] \right] \end{aligned}$$

Bellman optimality equations

- Bellman optimality equation for $Q^*(s)$

$$\begin{aligned} Q^*(s,a) &= R_s^a + \gamma \int_{s' \in S} P_{ss'}^a V^*(s') ds' \\ &= R_s^a + \gamma \int_{s' \in S} P_{ss'}^a \max_{a' \in \mathcal{A}} Q^*(s', a') ds' \\ &= R_s^a + \mathbb{E}_{s' \sim P_{ss'}^a} \left[\gamma \max_{a' \in \mathcal{A}} Q^*(s', a') \right] \end{aligned}$$

$\longrightarrow$ Nonlinear equations

$\longrightarrow$ Q is more tractable than V

- Optimal policy $\pi^*(s)$

$$\begin{aligned} \pi^*(s) &= \arg\max_{a \in \mathcal{A}} Q^*(s,a) = \arg\max_{a \in \mathcal{A}} \left[R_s^a + \gamma \int_{s' \in \mathcal{S}} P_{ss'}^a V^*(s') ds' \right] \\ &= \arg\max_{a \in \mathcal{A}} \left[R_s^a + \mathbb{E}_{s' \sim P_{ss'}^a} \left[\gamma V^*(s') \right] \right] \end{aligned}$$

Policy and value iterations

- Policy iteration = Policy evaluation + Policy improvement

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \cdots \xrightarrow{I} \pi_i \xrightarrow{E} V^{\pi_i} \xrightarrow{I} \pi_{i+1} \xrightarrow{E} V^{\pi_{i+1}} \cdots \pi^*$$

- Policy improvement :

$$\begin{aligned} \pi_{i+1}(s) &= \arg\max_{a \in \mathcal{A}} Q^{\pi_i}(s,a) \\ V^{\pi_{i+1}}(s) &\geq V^{\pi_i}(s) \end{aligned}$$

- Policy evaluation :

$$\begin{aligned} Q^{\pi}_{j+1}(s,a) &= R^a_s + \mathbb{E}_{s' \sim P^a_{ss'},\ a' \sim \pi(a'|s')}[\gamma Q^{\pi}_j(s',a') \mid s,a] \\ V^{\pi}_{j+1}(s) &= \mathbb{E}_{a \sim \pi(a|s)}[R^a_s] + \mathbb{E}_{a \sim \pi(a|s),\ s' \sim P^a_{ss'}}[\gamma V^{\pi}_j(s') \mid s] \end{aligned}$$

Note that j is used instead of i.

Policy and value iterations

- Bellman operator for policy evaluation

$$\mathcal{B}^{\pi}(Q) = R_s^a + \gamma \int_{s' \in \mathcal{S}} \int_{a' \in \mathcal{A}} \pi(a'|s') P_{ss'}^a Q(s', a') da' ds'$$

- Contraction mapping

$$\|\mathcal{B}^{\pi}(Q_1) - \mathcal{B}^{\pi}(Q_2)\|_{\infty} \leq \gamma \|Q_1 - Q_2\|_{\infty}$$

$$\|Q_1(s,a) - Q_2(s,a)\|_{\infty} = \max_{s,a} |Q_1(s) - Q_2(s)|$$

Policy and value iterations

- Value iteration
  $V^*(s)$ is available from the iterative Bellman optimality equation.

$$\begin{aligned} Q^*(s,a) &= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V^*(s') \\ \pi^*(s) &= \arg\max_{a \in \mathcal{A}} Q^*(s,a) \end{aligned}$$

- To obtain $Q^*(s,a)$

$$\begin{aligned} Q_{j+1}^*(s,a) &= R_s^a + \mathbb{E}_{s' \sim P_{ss'}^a} \left[\gamma \max_{a \in \mathcal{A}} Q_j^*(s,a) \right] \\ V_{j+1}^*(s) &= \max_{a \in \mathcal{A}} \left[R_s^a + \mathbb{E}_{s' \sim P_{ss'}^a} \left[\gamma V_j^*(s') \right] \right] \end{aligned}$$

  Note that j is used instead of i.

Policy and value iterations

- A contraction property of the Bellman optimality operator :

$$\begin{aligned}\mathcal{B}^*(V(s)) &= \max_{a\in\mathcal{A}}\left[R_s^a + \mathbb{E}_{s'\sim P_{ss'}^a}\left[\gamma V(s')\right]\right] \\ &= \max_{a\in\mathcal{A}}\left[R_s^a + \gamma\int_{s'\in\mathcal{S}} P_{ss'}^a V(s')ds'\right]\end{aligned}$$

Proof : Let $d = \|V_1(s) - V_2(s)\|_\infty = \max_s |V_1(s) - V_2(s)|$. Then we have

$$\begin{aligned}V_1(s) - d \le V_2(s) \le V_1(s) + d \\ \mathcal{B}^*(V_1(s)) - \gamma d \le \mathcal{B}^*(V_2(s)) \le \mathcal{B}^*(V_1(s)) + \gamma d\end{aligned}$$

which implies

$$\|\mathcal{B}^*(V_1(s)) - \mathcal{B}^*(V_2(s))\|_\infty \le \gamma d$$

A recursive equation $V_{j+1}(s) = \mathcal{B}^*(V_j(s))$ converges to a unique fixed point

Soft state value function

- Standard state value function : $\tau = (a_t, s_{t+1}, a_{t+1}, \cdots)$

$$V^{\pi}(s_t) = \mathbb{E}_{\pi}[G(t)|S_t = s_t] = \mathbb{E}_{\pi}\Big[\sum_{k=0} \gamma^k R_{t+k}|S_t = s_t\Big]$$

- Soft state value function : α is a temperature parameter.

$$V^{\pi}_{\text{soft}}(s_t) = \mathbb{E}_{\pi}\left[\sum_{k=0} \gamma^k \left(R_{t+k} - \alpha \log \pi(a_{t+k}|s_{t+k})\right) \middle| S_t = s_t\right]$$

It is reduced to the standard state value function if α is set to be zero.

- Entropy

$$-\mathbb{E}_{\pi}\left[\sum_{k=0} \gamma^k \alpha \log \pi(a_{t+k}|s_{t+k}) \middle| S_t = s_t\right] = \sum_{k=0} \gamma^k \mathbb{E}_{\pi}\left[\alpha \mathcal{H}(\pi(a_{t+k}|s_{t+k}))\right.$$

Soft state-action value function

- Standard state-action value function : $\tau = (s_{t+1}, a_{t+1}, s_{t+2}, \cdots)$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[\sum_{k=0} \gamma^k R_{t+k} | S_t = s_t, A_t = a_t]$$

- Soft state-action value function : α is a temperature parameter.

$$Q^{\pi}_{\text{soft}}(s_t, a_t) = \mathbb{E}_{\pi}\Bigg[\sum_{k=0} \gamma^k \left(R_{t+k} - \gamma\alpha \log \pi(a_{t+k+1}|s_{t+k+1})\right) \Bigg| S_t = s_t, A_t = a_t\Bigg]$$

- The relationship between $V^{\pi}_{\text{soft}}(s_t)$ and $Q^{\pi}_{\text{soft}}(s_t, a_t)$:
  $\tau = (a_t, s_{t+1}, a_{t+1}, \cdots)$

$$V^{\pi}_{\text{soft}}(s_t) = \mathbb{E}_{a_t \sim \pi(a_t|s_t)}\left[-\alpha \log \pi(a_t|s_t) + Q^{\pi}_{\text{soft}}(s_t, a_t)\right]$$

$$Q^{\pi}_{\text{soft}}(s_t, a_t) = R^a_s + \gamma\, \mathbb{E}_{s_{t+1} \sim P^{a_t}_{s_t s_{t+1}}}\left[V^{\pi}_{\text{soft}}(s_{t+1})\right]$$

Soft Bellman equations

- Soft Bellman equation for $V^{\pi}_{\text{soft}}(s_t)$

$$\begin{aligned} V^{\pi}_{\text{soft}}(s_t) &= \mathbb{E}_{a_t \sim \pi(a_t|s_t)}[R^{a_t}_{s_t} + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim P^{a_t}_{s_t s_{t+1}}}[V^{\pi}_{soft}(\mathbf{x}_{t+1})] \\ &- \alpha \log \pi(a_t|s_t)] \end{aligned}$$

- Soft Bellman equation for $Q^{\pi}_{\text{soft}}(s_t, a_t)$

$$Q^{\pi}_{\text{soft}}(s_t, a_t) = R^{a}_{s} + \gamma \ \mathbb{E}_{s' \sim P^{a}_{ss'}, a' \sim \pi(a'|s')}[Q^{\pi}_{\text{soft}}(s', a') - \alpha \log \pi(a'|s')]$$

Optimal policy

- Cost function

$$
\begin{aligned}
V^{\pi}_{\text{soft}}(s_t) &= \mathbb{E}_{a_t \sim \pi(a_t|s_t)} \left[-\alpha \log \pi(a_t|s_t) + Q^{\pi}_{\text{soft}}(s_t, a_t) \right] \\
&= \int_a \left[Q^{\pi}_{\text{soft}}(s,a) - \alpha \log \pi(a|s) \right] \pi(a|s) da \\
&= -\alpha D_{KL} \left(\pi(a|s) \parallel \frac{\exp\left(\frac{1}{\alpha} Q^{\pi}_{\text{soft}}(s,a)\right)}{Z(s)} \right) + \alpha \log Z(s)
\end{aligned}
$$

- Optimal policy $\pi(a|s)$

$$
\pi(a|s) = \frac{\exp\left(\frac{1}{\alpha} Q^{\pi}_{\text{soft}}(s,a)\right)}{\int_{\bar{a}} \exp\left(\frac{1}{\alpha} Q^{\pi}_{\text{soft}}(s,\bar{a})\right) d\bar{a}}
$$

Soft policy iteration

- Policy improvement

$$\pi_{i+1}(a|s) = \frac{\exp(\frac{1}{\alpha} Q^{\pi}_{i,\text{soft}}(s,a))}{\int_{\bar{a}} \exp(\frac{1}{\alpha} Q^{\pi}_{i,\text{soft}}(s,\bar{a}))d\bar{a}}$$

$$Q^{\pi}_{i+1,\text{soft}}(s,a) \geq Q^{\pi}_{i,\text{soft}}(s,a)$$

- Policy evaluation

$$V^{\pi}_{\text{soft}}(s) = \alpha \log \int_{a'} \exp\left(\frac{1}{\alpha} Q^{\pi}_{\text{soft}}(s,a')\right) da'$$

$$Q^{\pi}_{\text{soft}}(s,a) = R^{a}_{s} + \gamma \, \mathbb{E}_{s' \sim P^{a}_{ss'}}[V^{\pi}_{\text{soft}}(s')]$$

Soft policy iteration

- A contraction property of the soft Bellman operator

$$\mathcal{T}Q^{\pi}_{\text{soft}}(s,a) = R^{a}_{s} + \gamma\ \mathbb{E}_{s' \sim P^{a}_{ss'}}[\alpha \log \int_{a'} \exp\left(\frac{1}{\alpha} Q^{\pi}_{\text{soft}}(s,a')\right) da']$$

$$\|\mathcal{T}Q^{\pi}_{1,\text{soft}} - \mathcal{T}Q^{\pi}_{2,\text{soft}}\|_{\infty} \leq \gamma \|Q^{\pi}_{1,\text{soft}} - Q^{\pi}_{2,\text{soft}}\|_{\infty}$$

Soft Actor-Critic(SAC)

- Current State-of-the-art algorithm
- The entropy measure of the policy for encouring exploration
- An off-policy actor-critic model following the maximum entropy reinforcement learning framework
- A policy that acts as randomly as possible while still able to succeed at the task
- Three important functions of SAC
  1. The policy with parameter θ : π_θ
  2. The soft state-value function parameterized by w : $Q^\pi_{w,\text{soft}}(s_t, a_t)$

$$Q^\pi_{w,\text{soft}}(s_t, a_t) = R^{a_t}_{s_t} + \gamma \, \mathbb{E}_{s_{t+1} \sim P^{a_t}_{s_t s_{t+1}}} [V^\pi_{\text{soft}}(s_{t+1})]$$

  3. The soft state value function parameterized by ψ : $V^\pi_{\psi,\text{soft}}(s_t)$

$$V^\pi_{\psi,\text{soft}}(s_t) = \mathbb{E}_{a_t \sim \pi(a_t|s_t)} \left[-\alpha \log \pi(a_t|s_t) + Q^\pi_{w,\text{soft}}(s_t, a_t) \right]$$

- The soft state-value function is trained to minimize the mean squared error:

$$\begin{aligned} J_Q(w) &= \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\Big[\frac{1}{2}\Big(Q^{\pi}_{w,\text{soft}}(s_t,a_t) - \big(R^{a_t}_{s_t} \\ &+ \gamma\mathbb{E}_{s_{t+1}\sim P^{a_t}_{s_t s_{t+1}}}\big[V_{\bar{\psi}}(s_{t+1})\big]\big)\Big)^2\Big] \\ \nabla_w J_Q(w) &= \nabla_w Q^{\pi}_{w,\text{soft}}(s_t,a_t)\Big(Q^{\pi}_{w,\text{soft}}(s_t,a_t) \\ &- R^{a_t}_{s_t} - \gamma V_{\bar{\psi}}(s_{t+1})\Big) \end{aligned}$$

  where $\mathcal{D}$ is a dataset stored in a replay buffer and $\bar{\psi}$ is the target value function which is the exponential moving average or only gets updated periodically.
- The soft state value function is trained to minimize the mean squared error:

$$\begin{aligned} J_V(\psi) &= \mathbb{E}_{s_t\sim\mathcal{D}}\Big[\frac{1}{2}\Big(V^{\pi}_{\psi,\text{soft}}(s_t) - \mathbb{E}_{a_t\sim\pi_\theta}\big[Q^{\pi}_{w,\text{soft}}(s_t,a_t) \\ &- \log\pi_\theta(a_t|s_t)\big]\Big)^2\Big] \\ \nabla_\psi J_V(\psi) &= \nabla_\psi V^{\pi}_{\psi,\text{soft}}(s_t)\Big(V^{\pi}_{\psi,\text{soft}}(s_t) - Q^{\pi}_{w,\text{soft}}(s_t,a_t) \\ &+ \log\pi_\theta(a_t|s_t)\Big) \end{aligned}$$

- The policy is updated to minimize the following KL-divergence:

$$\begin{aligned}\pi_{i+1} &= \arg\min_{\pi' \in \Pi} D_{\mathsf{KL}}\left(\pi'(a|s) \| \frac{\exp(Q^{\pi}_{i,\mathsf{soft}}(s,a))}{Z^{\pi}_{i}(s)}\right) \\ &= \arg\min_{\pi' \in \Pi} D_{\mathsf{KL}}\left(\pi'(a|s) \| \exp(Q^{\pi}_{i,\mathsf{soft}}(s,a) - \log Z^{\pi}_{i}(s))\right)\end{aligned}$$

- Gradient for updating the policy

$$J_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}, a_t \sim \pi_\theta}\Big[-\log\Big(\frac{\exp(Q^\pi_{w,\text{soft}}(s_t, a_t) - \log Z_w(s_t))}{\pi_\theta(a_t|s_t)}\Big)\Big]$$

$$= \mathbb{E}_{s_t \sim \mathcal{D}, a_t \sim \pi_\theta}\big[\log \pi_\theta(a_t|s_t) - Q^\pi_{w,\text{soft}}(s_t, a_t) + \log Z_w(s_t)\big]$$

Thus,

$$\nabla_\theta J_\pi(\theta) = \nabla_\theta \log \pi_\theta(a_t|s_t) + \big[\nabla_{a_t} \log \pi_\theta(a_t|s_t) - \nabla_{a_t} Q^\pi_{w,\text{soft}}(s_t, a_t)\big]\nabla_\theta f_\theta(\epsilon_t; s_t),$$

where $a_t = f_\theta(\epsilon_t; s_t)$ with a non-parametric noise $\epsilon_t \sim \mathcal{N}$.

SAC with 5 networks

- 1. Initialize actor network θ, Q-network w, and V-network ϕ arbitarily
- 2. Set target V network $\bar{\phi} \leftarrow \phi$
- 3. Make replay buffer $\mathcal{D}$

Repeat (for each episode)
Receive initial state S_0
Repeat (for each step t of episode)
Select action $A_t \sim \pi_\theta(\cdot|S_t)$
Execute action A_t and observe reward R_t and new state S_{t+1}
Store transition (S_t, A_t, R_t, S_{t+1}) in $\mathcal{D}$
if S_t is terminal:
Repeat (for each gradient step)

$$\phi \leftarrow \phi - \lambda_V \nabla_\phi J_V(\phi)$$
$$w \leftarrow w - \lambda_Q \nabla_w J_Q(w)$$
$$\theta \leftarrow \theta - \lambda_\pi \nabla_\theta J_\pi(\theta)$$
$$\bar{\phi} \leftarrow \tau \cdot \bar{\phi} + (1-\tau) \cdot \phi$$

Practical difficulties for using DRL in industrial areas

- A high computational cost
- Large memory storage
- Cost-effective and lightweight-embedded devices

Approaches for more efficient computation of DL

- Bayesian pruning : A widely employed method using a Bayesian regularization technique.
- Quantization : 3-4 bits are popular.
- Weight-sharing

$\rightarrow$ Existing studies on Bayesian pruning are focused on supervised learning settings.

DRL + Bayesian pruning

- Policy optimization problme reformulated as a Bayesian optimization problem
- Off-policy DRL
- An adaptive strategy to automate the scaling of Bayesian regularization in reinforcement learning

$\rightarrow$ Sparse variational deterministic policy gradient (SVDPG)

Reinforcement learning

- Optimal policy

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho^\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$$

- State-action value function

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\mathbf{a}_{i>t} \sim \pi} \left[\sum_{i=t}^{\infty} \gamma^{i-t} r_i \,\middle|\, \mathbf{s}_t, \mathbf{a}_t \right]$$

- Bellman equation

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = r_t + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim P, \mathbf{a}_{t+1} \sim \pi} \left[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \right]$$

Bayesian pruning for neural network compression

- An effective neural network compression
- The dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$, where N is the number of data samples, $\mathbf{x}_n$ is the input, and $\mathbf{y}_n$ is the corresponding label.
- A Bayesian neural network (BNN) that consists of stochastic weights w with a prior distribution $p(w)$
- Bayesian deep learning to estimate the posterior distribution of the weights $p(w|\mathcal{D})$

$$p(w|\mathcal{D}) = p(\mathcal{D}|w)p(w)/p(\mathcal{D})$$

- A probability distribution $q_\phi(w)$ parameterized by ϕ is used to estimate the true posterior distribution $p(w|\mathcal{D})$

Bayesian pruning for neural network compression

- The parameter ϕ is optimized by

$$\mathcal{L}(\phi) = \mathcal{L}_{\mathcal{D}}(\phi) - D_{KL}\left(q_{\phi}(w) \parallel p(w)\right)$$

  where $\mathcal{L}_{\mathcal{D}}(\phi)$ is given by:

$$\mathcal{L}_{\mathcal{D}}(\phi) = \mathbb{E}_{w \sim q_{\phi}}\left[\log p(\mathcal{D}|w)\right]$$

- The expected log-likelihood in the first term $\mathcal{L}_{\mathcal{D}}(\phi)$ serves to minimize the model prediction error over the given dataset, and the KL-divergence in the second term is for the regularization purpose.

SVDPG

- Actor-critic architecture :
  Actor : Policy estimate ($\pi(\mathbf{s}_t) \approx \pi_w(\mathbf{s}_t)$)
  Critic : Value function estimate ($Q(\mathbf{s}_t, \mathbf{a}_t) \approx Q_\psi(\mathbf{s}_t, \mathbf{a}_t)$)
- Data set :
  $\mathcal{D} = \{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})_n\}_{n=1}^N$
- Training objective function of the critic (Q_ψ)

$$\mathcal{J}_Q(\psi) =$$

$$\mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\left(r_t + \gamma \mathbb{E}_{w \sim q_\phi} \left[Q_\psi \left(\mathbf{s}_{t+1}, \pi_w(\mathbf{s}_{t+1}) \right) \right] - Q_\psi(\mathbf{s}_t, \mathbf{a}_t) \right)^2 \right]$$

- Boltzmann distribution

$$B(\mathbf{s}_t, \mathbf{a}_t) = \frac{\exp\left(Q_\psi\left(\mathbf{s}_t, \mathbf{a}_t\right)/\kappa\right)}{\int_{\mathcal{A}} \exp\left(Q_\psi\left(\mathbf{s}_t, \mathbf{a}\right)/\kappa\right) d\mathbf{a}}$$

- Likelihood $p(\mathcal{D}|w)$

$$\begin{aligned} p(\mathcal{D}|w) &= \prod_{\mathbf{s}_t \in \mathcal{D}} B(\mathbf{s}_t, \pi_w(\mathbf{s}_t)) \\ &= \prod_{\mathbf{s}_t \in \mathcal{D}} \frac{\exp\left(Q_\psi\left(\mathbf{s}_t, \pi_w(\mathbf{s}_t)\right)/\kappa\right)}{Z_\psi(\mathbf{s}_t)} \end{aligned}$$

where $Z_\psi(\mathbf{s}_t) = \int_{\mathcal{A}} \exp\left(Q_\psi\left(\mathbf{s}_t, \mathbf{a}\right)/\kappa\right) d\mathbf{a}$.

- ELBO (Evidence lower bound)

$$\mathcal{L}(\phi) = \mathcal{L}_{\mathcal{D}}(\phi) - D_{KL}\left(q_\phi(w) \parallel p(w)\right)$$

- Training objective function of the actor (q_ϕ)

$$\mathcal{J}_\pi(\phi) = \frac{1}{N} \sum_{\mathbf{s}_t \in \mathcal{D}} \mathbb{E}_{w \sim q_\phi} \quad [Q_\psi(\mathbf{s}_t, \pi_w(\mathbf{s}_t))] - \lambda D_{KL}(q_\phi(w) \parallel p(w))$$

$$\lambda(t) = \begin{cases} (G_{\text{max}} - G_{\text{min}})/10^7 & \text{if } t > 0.5 \text{ million} \\ 0 & \text{Otherwise} \end{cases}$$

where t denotes the current time steps, G_{max} represents the maximum number of episodic total rewards that the agent has received during learning, and G_{min} indicates the minimum number of episodic total rewards.

- Exploration policy

$$\pi'_w(s_t) = \pi_w(s_t) + \epsilon \ , \ \ \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Reparameterization trick

$$\mathbb{E}_{w \sim q_\phi} \left[Q_\psi \left(\mathbf{s}, \pi_w(\mathbf{s}) \right) \right] \simeq Q_\psi(\mathbf{s}, \pi_{\hat{w}}(\mathbf{s}))$$

$$\int Q_\psi \left(\mathbf{s}, \pi_w(\mathbf{s}) \right) q_\phi(w) dw \simeq \int Q_\psi \left(\mathbf{s}, \pi_{\hat{w}}(\mathbf{s}) \right) g(\hat{\epsilon}) d\hat{\epsilon}$$

where $\hat{w} = f(\phi, \hat{\epsilon})$ with a sample noise $\hat{\epsilon} \sim g(\epsilon)$.
The reparameterization trick is employed to build a differentiable relation between the parameters ϕ and the expectation.

- The M-sized minibatch sampled from the replay buffer $\mathcal{D}$ is denoted as $\mathcal{M} = \{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})_m\}_{m=1}^{M}$.
- Update rule of the critic (ψ)

$$\begin{aligned}\hat{\nabla}_\psi \mathcal{J}_Q(\psi) &= \frac{1}{M}\sum_{\mathcal{M}} \nabla_\psi Q_\psi(\mathbf{s}_t, \mathbf{a}_t)(r_t + \gamma Q_{\bar{\psi}}(\mathbf{s}_{t+1}, \pi_{\hat{w}}(\mathbf{s}_{t+1})) \\ &- Q_\psi(\mathbf{s}_t, \mathbf{a}_t))\end{aligned}$$

where $\hat{w} = f(\bar{\phi})$, and $\bar{\psi}$ and $\bar{\phi}$ are exponentially moving the average of ψ and ϕ.

- Update rule of the actor (ϕ)

$$\hat{\nabla}_\phi \mathcal{J}_\pi(\phi) = \frac{1}{M} \sum_{\mathcal{M}} \nabla_\phi Q_\psi(\mathbf{s}_t, \pi_{\hat{w}}(\mathbf{s}_t)) - \lambda \nabla_\phi D_{KL}(q_\phi(w) \parallel p(w))$$

- Less-frequent updates of the policy and target networks can ensure less volatile estimates of the critic and then lead to more stabilizing and efficient learning.

$$\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$$
$$\bar{\phi} \leftarrow \tau\phi + (1-\tau)\bar{\phi}$$

- Sparcification of the policy network

$$w_i^{\mathsf{SP}} = \mathbb{E}[w_i] \cdot \mathsf{I}_{[c,\infty)}(\mathsf{SNR}(w_i))$$

where

$$\mathsf{SNR}(w_i) = \frac{\mathbb{E}\left[w_i\right]^2}{\mathsf{Var}\left[w_i\right]}, \qquad w_i \sim q_{\phi_i}(w_i)$$

Implementation issues

- Parameterized posteriori distribution :

$$q_\phi(w) = \mathcal{N}(w|\theta, \sigma^2)$$

- Prior distribution :

$$p(\log|w|) = \text{constant} \Leftrightarrow p(|w|) \propto 1/|w|$$

- Reparameterization trick

$$w_i = f(\phi_i, \epsilon_i) = \theta_i + \sigma_i \epsilon_i, \quad \text{with} \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

- Approximation

$$D_{KL}\left(q_\phi(w) \parallel p(w)\right) = \sum_i D_{KL}\left(q_{\phi_i}(w_i) \parallel p(w_i)\right)$$

$$\simeq \sum_i \left(k_1 \delta(k_2 + k_3 \log \alpha_i) + 0.5 \log(1 + \alpha_i^{-1}) - k_1\right) \quad , \alpha_i = \sigma_i^2/\theta_i^2$$

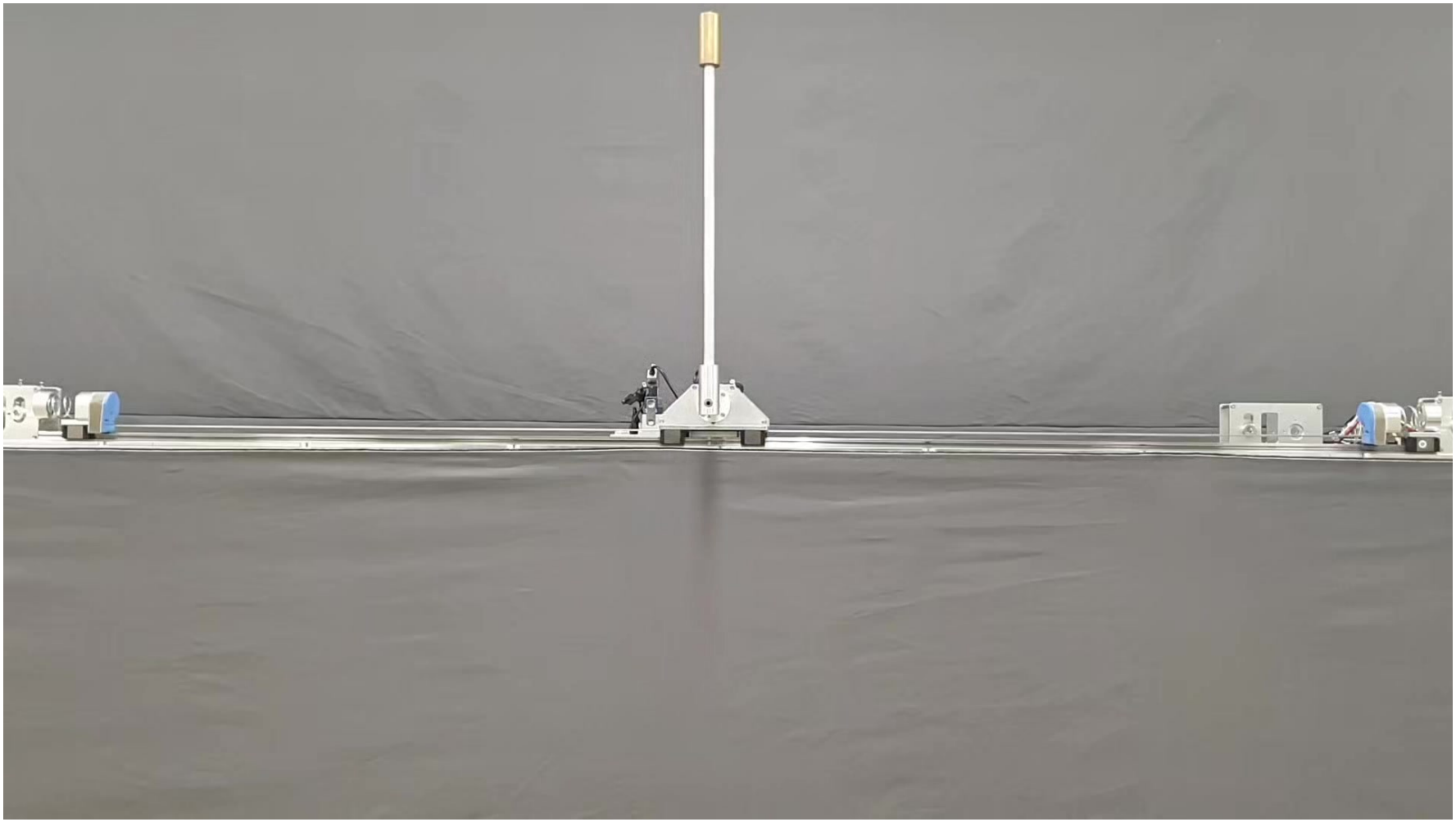
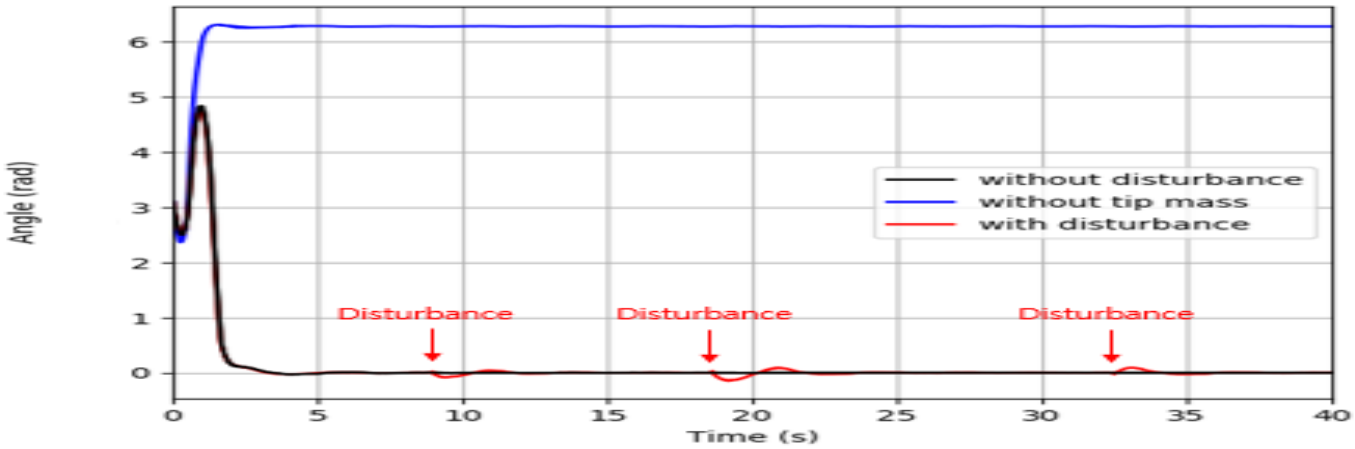
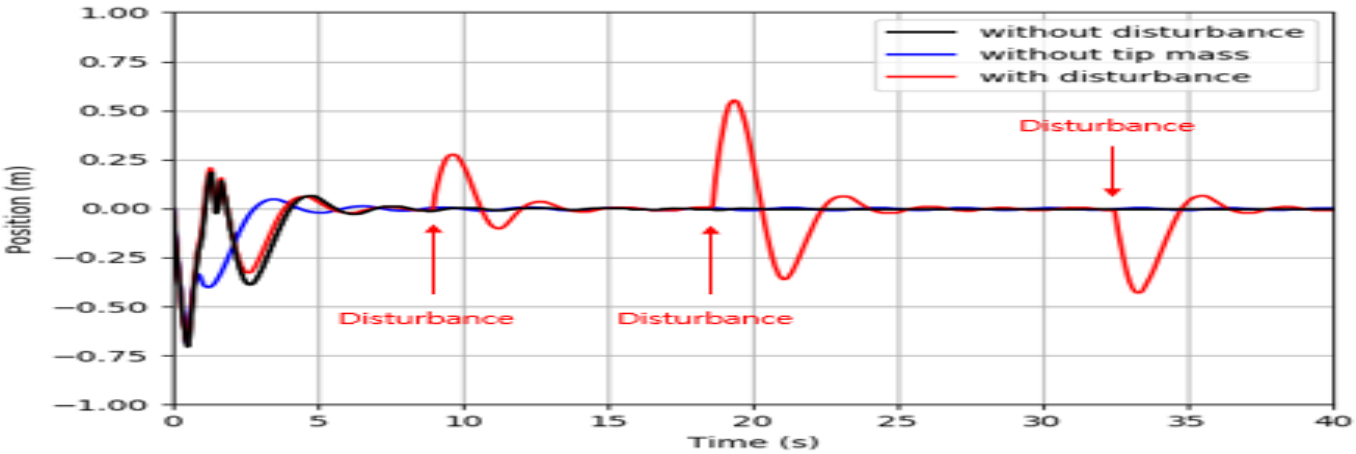


Figure: Front view of the inverted pendulum on a cart.



(a)



(b)

Figure: Disturbance rejection ability of the learned policy based on SVDPG: (a) The vertical angle of the pendulum; (b) The position of the cart.