最大熵强化学习:SQL 和 SAC

高辰潇

NJUAI

March 18, 2021

目录

目录

Maximum Entropy Reinforcement Learning

- 对于正常的 MDP 问题，强化学习的目标为最大化一段 tranjectory 中的累积期望奖赏

$$\pi^*_{std} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \rho_\pi} [r(\boldsymbol{s}_t, \boldsymbol{a}_t)]$$

- 最大熵强化学习框架中，使用熵来正则化优化目标

$$\pi^*_{MaxEnt} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \rho_\pi} [r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \alpha \mathcal{H}(\pi(\cdot|\boldsymbol{s}_t))]$$

- 在最大化期望收益的同时引入最大熵，促进 Agent 的探索，同时让策略面对扰动的时候更加稳定。

目录

Soft Value Functions and Energy-based Model

- **定义 Soft Q 函数**

$$Q^*_{soft}(s_t, a_t) = r_t + \mathbb{E}_{(s_{t+1},\ldots)\sim\rho_\pi}[\sum_{l=1}^{\infty}\gamma^l(r_{t+l} + \alpha\mathcal{H}(\pi^*_{MaxEnt}(\cdot|s_{t+l})))]$$

- **以及 Soft 值函数**

$$V^*_{soft}(s_t) = \alpha\log\int_{\mathcal{A}}\exp(\frac{1}{\alpha}Q^*_{soft}(s_t, a'))\mathrm{d}a'$$

- **那么最优策略 π^*_{MaxEnt} 为**

$$\pi^*_{MaxEnt}(s_t, a_t) = \exp(\frac{1}{\alpha}(Q^*_{soft}(s_t, a_t) - V^*_{soft}(s_t)))$$

Soft Value Iteration and Soft Bellman Equation

- **证明** Soft Q **函数满足** Soft Bellman Equation

$$Q^*_{soft}(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim p_s}[V^*_{soft}(s_{t+1})]$$

- **证明**

$$\begin{aligned}
&Q^\pi_{soft}(s_t, a_t) \\
&= r(s_t, a_t) + \mathbb{E}_{(s_{t+1}, \ldots) \sim \rho_\pi}[\sum_{l=1}^{\infty} \gamma^l (r_{t+l} + \alpha \mathcal{H}(\pi^*_{MaxEnt}(\cdot|s_{t+l})))] \\
&= r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p_s}[\alpha \mathcal{H}(\pi(\cdot|s_{t+1})) + \mathbb{E}_{a_{t+1} \sim \pi(\cdot|s_{t+1})}[Q^\pi_{soft}(s_{t+1}, a_{t+1})]] \\
&= r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}, a_{t+1}}[Q^\pi_{soft}(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1})] \\
&= r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}, a_{t+1}}[Q^\pi(s_{t+1}, a_{t+1}) - \alpha \log(\exp(\frac{1}{\alpha}(Q^\pi_{soft}(s_{t+1}, a_{t+1}) - V^\pi_{soft}(s_{t+1}))))] \\
&= r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}}[V^\pi_{soft}(s_{t+1})]
\end{aligned}$$

Soft Value Iteration and Soft Bellman Equation (cont'd)

- **定义** Soft Value Iteration **算子** $\mathcal{T}$，**则** $\mathcal{T}$ **满足** contraction。

The following proof has also been presented by Fox et al. (2016). Define a norm on Q-values as $\|Q_1 - Q_2\| \triangleq \max_{\mathbf{s},\mathbf{a}} |Q_1(\mathbf{s},\mathbf{a}) - Q_2(\mathbf{s},\mathbf{a})|$. Suppose $\varepsilon = \|Q_1 - Q_2\|$. Then

$$\begin{aligned}
\log \int \exp(Q_1(\mathbf{s}',\mathbf{a}'))\, d\mathbf{a}' &\leq \log \int \exp(Q_2(\mathbf{s}',\mathbf{a}') + \varepsilon)\, d\mathbf{a}' \\
&= \log \left(\exp(\varepsilon) \int \exp Q_2(\mathbf{s}',\mathbf{a}')\, d\mathbf{a}' \right) \\
&= \varepsilon + \log \int \exp Q_2(\mathbf{a}',\mathbf{a}')\, d\mathbf{a}'. \qquad (25)
\end{aligned}$$

Similarly, $\log \int \exp Q_1(\mathbf{s}',\mathbf{a}')\, d\mathbf{a}' \geq -\varepsilon + \log \int \exp Q_2(\mathbf{s}',\mathbf{a}')\, d\mathbf{a}'$. Therefore $\|\mathcal{T}Q_1 - \mathcal{T}Q_2\| \leq \gamma\varepsilon = \gamma\|Q_1 - Q_2\|$. So $\mathcal{T}$ is a contraction. As a consequence, only one Q-value satisfies the soft Bellman equation, and thus the optimal policy presented in Theorem 1 is unique.

Policy Iteration (from SAC)

- 给定当前策略 π，定义新策略 $\hat{\pi}$ 为 $\hat{\pi}(\cdot|s) \propto \exp(Q^{\pi}_{soft}(s,\cdot))$，则有 $Q^{\hat{\pi}}_{soft}(s,a) \geq Q^{\pi}_{soft}(s,a) \quad \forall (s,a)$
- 证明：使用 π 和 Q 之间分布的 KL 散度作为优化目标，则有

$$\mathbb{E}_{a\sim\hat{\pi}}[\log\hat{\pi}(a|s) - Q^{\pi}(s,a) + V^{\pi}(s)] \leq \mathbb{E}_{a\sim\pi}[\log\pi(a|s) - Q^{\pi}(s,a) + V^{\pi}(s)]$$

代入 V^{π} 和 π, Q^{π} 的关系，得到

$$\mathbb{E}_{a\sim\hat{\pi}}[Q^{\pi}(s,a) - \log\hat{\pi}(a|s)] \geq V^{\pi}(s)$$

因此有

$$\begin{aligned}
&Q^{\pi}(s_t,a_t)\\
&= r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}}[V^{\pi}(s_{t+1})]\\
&\leq r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}}\mathbb{E}_{a_{t+1}\sim\hat{\pi}}[Q^{\pi}(s_{t+1},a_{t+1}) - \log\hat{\pi}(a_{t+1}|s_{t+1})]\\
&\leq \cdots \leq Q^{\hat{\pi}}(s_t,a_t)
\end{aligned}$$

Soft Q-learning: Algorithm

Algorithm 1 Soft Q-learning

$\theta, \phi \sim$ some initialization distributions.
Assign target parameters: $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$.
$\mathcal{D} \leftarrow$ empty replay memory.

for each epoch **do**
  for each t **do**
    Collect experience
    Sample an action for $\mathbf{s}_t$ using f^{ϕ}:
    $\mathbf{a}_t \leftarrow f^{\phi}(\xi; \mathbf{s}_t)$ where $\xi \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
    Sample next state from the environment:
    $\mathbf{s}_{t+1} \sim p_{\mathbf{s}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.
    Save the new experience in the replay memory:
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$.
    Sample a minibatch from the replay memory
    $\{(\mathbf{s}_t^{(i)}, \mathbf{a}_t^{(i)}, r_t^{(i)}, \mathbf{s}_{t+1}^{(i)})\}_{i=0}^{N} \sim \mathcal{D}$.
    Update the soft Q-function parameters
    Sample $\{\mathbf{a}^{(i,j)}\}_{j=0}^{M} \sim q_{\mathbf{a}'}$ for each $\mathbf{s}_{t+1}^{(i)}$.
    Compute empirical soft values $\hat{V}_{\text{soft}}^{\bar{\theta}}(\mathbf{s}_{t+1}^{(i)})$ in (10).
    Compute empirical gradient $\hat{\nabla}_{\theta} J_Q$ of (11).
    Update θ according to $\hat{\nabla}_{\theta} J_Q$ using ADAM.
    Update policy
    Sample $\{\xi^{(i,j)}\}_{j=0}^{M} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for each $\mathbf{s}_t^{(i)}$.
    Compute actions $\mathbf{a}_t^{(i,j)} = f^{\phi}(\xi^{(i,j)}, \mathbf{s}_t^{(i)})$.
    Compute Δf^{ϕ} using empirical estimate of (13).
    Compute empiricial estimate of (14): $\hat{\nabla}_{\phi} J_{\pi}$.
    Update ϕ according to $\hat{\nabla}_{\phi} J_{\pi}$ using ADAM.
  end for
  if epoch *mod* update_interval $= 0$ **then**
    Update target parameters: $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$.
  end if
end for

其他细节

- Soft 值函数 V 需要在当前状态的行动空间上积分，在实际操作中没有办法做到。在实际操作中，值函数通过重要性采样的方法去估计

$$V^{\theta}_{soft}(s_t) = \alpha \log \mathbb{E}_{q_{a'}}[\frac{\exp(\frac{1}{\alpha} Q^{\theta}_{soft}(s_t, a'))}{q_{a'}(a')}]$$

- 基于能量的策略模型 $\pi(\cdot|s) \propto \exp(Q^{\pi}_{soft}(s, \cdot))$ 难以直接采样，实际操作中使用近似推理，例如 MCMC。
- 直观来看，Soft Q-learning 是将 Q-learning 推广到大规模连续动作空间的算法。其中 softmax（LogSumExp）对应 Q-learning 中的 hardmax 操作。
- 尽管 SQL 算法中存在 actor 和 Q，但 Q 函数是对最优 Q 函数的估计，而 actor 由 Q 函数直接导出，不会对 Q 函数产生直接影响。

目录

Soft Value Functions

- 为了避免 V 需要对动作空间进行积分的问题，SAC 中将 Soft Value Function 修改为

$$V(s_t) = \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \log \pi(a_t|s_t)]$$

- Soft Q 函数的定义基本不变，为

$$Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s_{t+1}}[V(s_{t+1})]$$

- 策略模型仍为基于能量的模型，在给定 $Q^{\pi_{old}}$ 后，更新策略为

$$\pi_{new} = \arg\min_{\pi' \in \Pi} \left(\pi'(\cdot|s_t) \middle|\middle| \frac{\exp(Q^{\pi_{old}}(s_t, \cdot))}{Z^{\pi_{old}}(s_t)} \right)$$

Objectives

- Q、V、π 其中任意两者皆可导出第三者。而在最初的 SAC 中，作者发现使用三个网络分别近似 V、Q、π 可提高训练稳定性

$$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}}[\frac{1}{2}(V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi}[Q_\theta(s_t, a_t) - \log \pi(a_t|s_t)])^2]$$

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}}[\frac{1}{2}(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t))^2]$$

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}}\left[D_{KL}\left(\pi'(\cdot|s_t) \middle|\middle| \frac{\exp(Q^{\pi_{old}}(s_t, \cdot))}{Z^{\pi_{old}}(s_t)}\right)\right]$$

$$\hat{Q}(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1}}[V_{\bar{\psi}}(s_{t+1})]$$

SAC: Algorithm

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

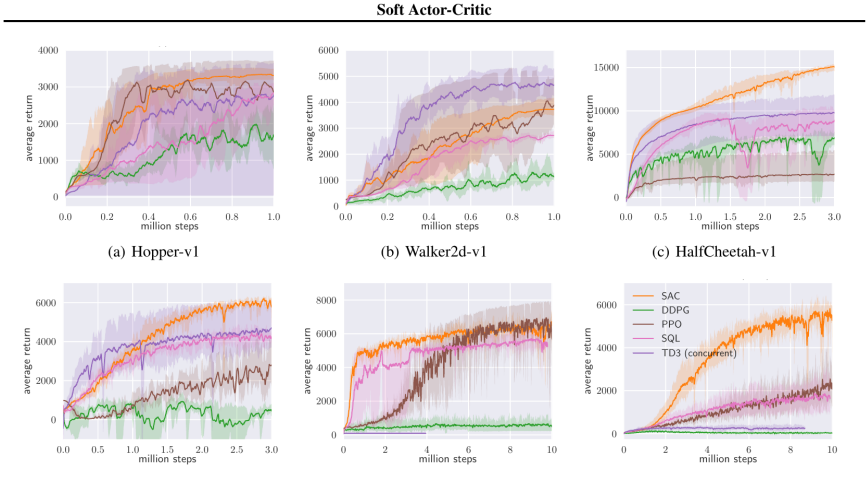
其他细节

- 将策略空间限制为某一类函数空间，并使用 reparameterization trick。具体而言，SAC 中的策略为 $a_t = f_\phi(\epsilon; s_t)$，其中 ϵ 为从 Gaussian 中采样得到的噪声。进一步将 actor 的优化目标重写为

$$J(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}, \epsilon \sim \mathcal{N}}[\log \pi_\phi(f_\phi(\epsilon; s_t)|s_t) - Q_\theta(s_t, f_\phi(\epsilon; s_t))]$$

- SAC 是 off-policy 的算法，因此可使用 Experience Replay 提高 sample efficiency。

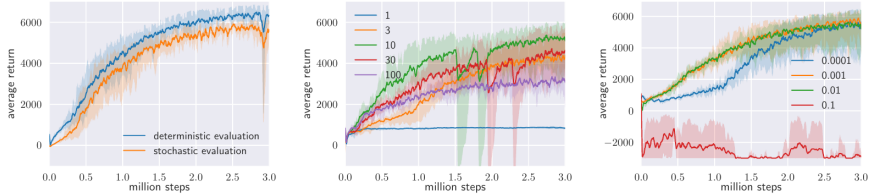
实验



(a) Hopper-v1 (b) Walker2d-v1 (c) HalfCheetah-v1

(d) Ant-v1 (e) Humanoid-v1 (f) Humanoid (rllab)

实验 (cont'd)



(a) Evaluation (b) Reward Scale (c) Target Smoothing Coefficient (τ)

Figure 3. Sensitivity of soft actor-critic to selected hyperparameters on Ant-v1 task. (a) Evaluating the policy using the mean action generally results in a higher return. Note that the policy is trained to maximize also the entropy, and the mean action does not, in general, correspond the optimal action for the maximum return objective. (b) Soft actor-critic is sensitive to reward scaling since it is related to the temperature of the optimal policy. The optimal reward scale varies between environments, and should be tuned for each task separately. (c) Target value smoothing coefficient τ is used to stabilize training. Fast moving target (large τ) can result in instabilities (red), whereas slow moving target (small τ) makes training slower (blue).

目录

Adaptive Temperature

- 温度系数 α 控制对熵项的重视程度，系数越高越倾向探索，得到随机性较高的策略。

$$\pi^*_{MaxEnt} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{a}_t) \sim \rho_\pi} [r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \alpha \mathcal{H}(\pi(\cdot|\boldsymbol{s}_t))]$$

- 在不同任务、训练过程中的不同时刻下，Agent 取得的累积奖赏的数值规模是不同的；因此需要找到一个自适应调节温度系数 α 的算法。

Revisiting SAC

- 将 SAC 算法形式化为如下优化问题

$$\max_{\pi_0,\ldots,\pi_T} \mathbb{E}[\sum_{t=0}^{T} r(s_t, a_t)] \qquad \text{s.t. } \forall t, \mathcal{H}(\pi_t) \geq \mathcal{H}_0$$

- 使用动态规划分解为数个子问题，并按照时间逆序逐个优化

$$\max_{\pi_0} \Big(\mathbb{E}[r(s_0,a_0)] + \underbrace{\max_{\pi_1} \Big(\mathbb{E}[\ldots] + \underbrace{\max_{\pi_T} \mathbb{E}[r(s_T,a_T)]}_{\text{1st maximization}} \Big)}_{\text{second but last maximization}} \Big)}_{\text{last maximization}}$$

Revisiting SAC (con't)

- 考虑第一层优化问题，由拉格朗日对偶法可得

$$\begin{aligned}
&\max_{\pi_T} \mathbb{E}[r(s_T, a_T)] \\
&= \min_{\alpha_T \geq 0} \max_{\pi_T} \mathbb{E}_{(s_T, a_T) \sim \rho_\pi}[r(s_T, a_T)] + \alpha_T(\mathbb{E}_{(s_t, a_t) \sim \rho_\pi}[-\log \pi_T(a_T|s_T)] - \mathcal{H}_0) \\
&= \min_{\alpha_T \geq 0} \max_{\pi_T} \mathbb{E}_{(s_T, a_T) \sim \rho_\pi}[r(s_T, a_T) + \alpha_T \mathcal{H}(\pi_T) - \alpha_T \mathcal{H}_0]
\end{aligned}$$

- 文章中使用交替优化的方式求解最优 π_T^* 和 α_T^*: 首先固定 α_T，对内层优化问题进行求解得到 α_T^*，然后固定 α_T^*，求解 α_T^*。

$$\pi_T^* = \arg\max_{\pi_T} \mathbb{E}_{(s_T, a_T) \sim \rho_\pi}[r(s_T, a_T) + \alpha_T \mathcal{H}(\pi_T) - \alpha_T \mathcal{H}_0]$$

$$\alpha_T^* = \arg\min_{\alpha_T \geq 0} \mathbb{E}_{(s_T, a_T) \sim \rho_{\pi^*}}[\alpha_T \mathcal{H}(\pi_T) - \alpha_T \mathcal{H}_0]$$

Revisiting SAC (con't)

将 π_T^* 插入到 Q 函数中

$$Q_{T-1}(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \mathbb{E}[Q(s_T, a_T) - \alpha_T \log \pi(a_T|s_T)]$$
$$Q_{T-1}^*(s_{T-1}, a_{T-1}) = r(s_{T-1}, a_{T-1}) + \max_{\pi_T} \mathbb{E}[r(s_T, a_T)] + \alpha_T^* \mathcal{H}(\pi_T^*)$$

Revisiting SAC (con't)

- 考虑两层优化

$$
\begin{aligned}
&\max_{\pi_{T-1}} \left(\mathbb{E}[r(s_{t-1}, a_{t-1})] + \max_{\pi_T} \mathbb{E}[r(s_T, a_T)] \right) \\
&= \max_{\pi_{T-1}} \left(Q^*_{T-1}(s_{T-1}, a_{T-1}) - \alpha^*_T \mathcal{H}(\pi^*_T) \right) \\
&= \min_{\alpha_{T-1} \geq 0} \max_{\pi_{T-1}} \left(Q^*_{T-1}(s_{T-1}, a_{T-1}) - \alpha^*_T \mathcal{H}(\pi^*_T) + \alpha_{T-1}(\mathcal{H}(\pi_{T-1}) - \mathcal{H}_0) \right) \\
&= \min_{\alpha_{T-1} \geq 0} \max_{\pi_{T-1}} \left(Q^*_{T-1}(s_{T-1}, a_{T-1}) + \alpha_{T-1}\mathcal{H}(\pi_{T-1}) - \alpha_{T-1}\mathcal{H}_0 \right) - \alpha^*_T \mathcal{H}(\pi^*_T)
\end{aligned}
$$

- 和前面类似，α^*_{T-1} 满足

$$
\alpha^*_{T-1} = \arg \min_{\alpha_{T-1} \geq 0} \mathbb{E}_{(s_{T-1}, a_{T-1}) \sim \rho_{\pi^*}} [\alpha_{T-1}\mathcal{H}(\pi^*_{T-1}) - \alpha_{T-1}\mathcal{H}_0]
$$

Revisiting SAC (con't)

- 可归纳出 α 的更新目标为

$$\min \ J(\alpha) = \mathbb{E}_{a_t \sim \pi_t}[-\alpha \log \pi_t(a_t|s_t) - \alpha \mathcal{H}_0]$$

- 据此修改原本 SAC 的算法流程，将 α 的 gradient step 插入到 π 的更新过程中即可。

SAC with Automatically Adjusted Temperature

Algorithm 1 Soft Actor-Critic

Input: θ_1, θ_2, ϕ ▷ Initial parameters
$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ ▷ Initialize target network weights
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay pool
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$ ▷ Sample action from the policy
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$ ▷ Store the transition in the replay pool
  end for
  for each gradient step **do**
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$ ▷ Update the Q-function parameters
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Update policy weights
    $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$ ▷ Adjust temperature
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i$ for $i \in \{1, 2\}$ ▷ Update target network weights
  end for
end for
Output: θ_1, θ_2, ϕ ▷ Optimized parameters

Reference

- Haarnoja, Tuomas, Haoran Tang, Pieter Abbeel, and Sergey Levine. "Reinforcement Learning with Deep Energy-Based Policies." ArXiv:1702.08165 [Cs], July 21, 2017. http://arxiv.org/abs/1702.08165.
- Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor." ArXiv:1801.01290 [Cs, Stat], August 8, 2018. http://arxiv.org/abs/1801.01290.
- Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, et al. "Soft Actor-Critic Algorithms and Applications." ArXiv:1812.05905 [Cs, Stat], January 29, 2019. http://arxiv.org/abs/1812.05905.