Paper Presentation

Group 2

309505018 郭俊廷
309551026 黃柏愷

Soft Actor-Critic:
Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

ICML 2018

Outline

1. Problem we want to solve
2. What’s different between Soft Actor-Critic and others
3. Advantages of Maximum Entropy
4. Meticulous hyperparameter tuning
5. Experiment Result

Problem we want to solve

Some famous model-free algorithm in Reinforcement Learning, like DDPG and PPO, will learn a policy to do a continuous control or a series of action.

PPO (Proximal Policy Optimization) is an on-policy algorithm, it perfoms very well on discrete and continuous controll problem. But the weaks of PPO is sample inefficiency problem.

DDPG (Deep Deterministic Policy Gradient) is an off-policy algorithm, more sample efficient than PPO. But the weaks of DDPG is a deterministic policy, that means it will select best action when policy gradiant, and lacks of stochastic.

Problem we want to solve

Soft Actor-Critic (SAC) is an off-policy algorithm, and it used “Maximum Entropy” method to update policy.

Compared to PPO, it solved sample inefficiency problem.

Compared to DDPG, it used stochastic policy which performs better than deterministic policy in many environment.

The most important, Soft Actor-Critic (SAC) is an Open Source project, so that we can replicate it easy. It can use in robotics to solve real world problem.

What's different between Soft Actor-Critic and others

Original Policy Gradient

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [\sum_t R(s_t, a_t)]$$

Add Maximum Entropy method

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [\sum_t \underbrace{R(s_t, a_t)}_{reward} + \alpha \underbrace{H(\pi(\cdot|s_t))}_{entropy}]$$

Soft Policy Iteration

$$\mathcal{T}^{\pi} Q(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma \, \mathbb{E}_{\mathbf{s}_{t+1} \sim p} \left[V(\mathbf{s}_{t+1}) \right],$$

where

$$V(\mathbf{s}_t) = \mathbb{E}_{\mathbf{a}_t \sim \pi} \left[Q(\mathbf{s}_t, \mathbf{a}_t) - \log \pi(\mathbf{a}_t | \mathbf{s}_t) \right]$$

$$r_{soft}(s_t, a_t) = r(s_t, a_t) + \gamma\alpha\mathbb{E}_{s_{t+1}\sim\rho}H(\pi(\cdot|s_{t+1}))$$

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}, a_{t+1}}[Q(s_{t+1}, a_{t+1})]$$

$$\begin{aligned}
Q_{soft}(s_t, a_t) &= r(s_t, a_t) + \gamma\alpha\mathbb{E}_{s_{t+1}\sim\rho}H(\pi(\cdot|s_{t+1})) + \gamma\mathbb{E}_{s_{t+1}, a_{t+1}}[Q_{soft}(s_{t+1}, a_{t+1})] \\
&= r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\rho, a_{t+1}\sim\pi}[Q_{soft}(s_{t+1}, a_{t+1})] + \gamma\alpha\mathbb{E}_{s_{t+1}\sim\rho}H(\pi(\cdot|s_{t+1})) \\
&= r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\rho, a_{t+1}\sim\pi}[Q_{soft}(s_{t+1}, a_{t+1})] + \gamma\mathbb{E}_{s_{t+1}\sim\rho}\mathbb{E}_{\mathrm{a}_{t+1}\sim\pi}[-\alpha\log\pi(a_{t+1}|s_{t+1})] \\
&= r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\rho}[\mathbb{E}_{a_{t+1}\sim\pi}[Q_{soft}(s_{t+1}, a_{t+1}) - \alpha\log(\pi(a_{t+1}|s_{t+1}))]] \\
&= r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}, a_{t+1}}[Q_{soft}(s_{t+1}, a_{t+1}) - \alpha\log(\pi(a_{t+1}|s_{t+1}))]
\end{aligned}$$

Soft Policy Iteration

$$\pi_{\text{new}}(\cdot|\mathbf{s}_t) = \arg\min_{\pi'\in\Pi} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\cdot|\mathbf{s}_t) \,\Big\|\, \exp\left(Q^{\pi_{\text{old}}}(\mathbf{s}_t,\cdot) - \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right)\right)$$
$$= \arg\min_{\pi'\in\Pi} J_{\pi_{\text{old}}}(\pi'(\cdot|\mathbf{s}_t)).$$

$$J_{\pi_{\text{old}}}(\pi_{\text{new}}(\,\cdot\,|\mathbf{s}_t)) \leq J_{\pi_{\text{old}}}(\pi_{\text{old}}(\,\cdot\,|\mathbf{s}_t)),$$

$$\mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{new}}}\left[\log \pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right] \leq$$

$$\mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{old}}}\left[\log \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right]$$

$$\mathbb{E}_{\mathbf{a}_t\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)\right] \geq V^{\pi_{\text{old}}}(\mathbf{s}_t)$$

$$
\begin{aligned}
Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) &= r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1})\right] \\
&\leq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[\mathbb{E}_{\mathbf{a}_{t+1}\sim\pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \log \pi_{\text{new}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})\right]\right] \\
&\;\;\vdots \\
&\leq Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t),
\end{aligned}
$$

Advantages of Maximum Entropy Reinforcement Learning

1. Through maximum entropy, policy not only learns a way to solve tasks, but all. Therefore, such a policy is more conducive to learning new tasks.

2. Stronger exploration ability, it is easier to find better modes under multimodal reward.

3. More robust and generalized. Because it is necessary to explore various optimal possibilities in different ways, it is easier to make adjustments in the face of interference.

Implementation of SAC

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

$$J_Q(\theta) = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\mathbf{s}_t,\mathbf{a}_t) - \hat{Q}(\mathbf{s}_t,\mathbf{a}_t)\right)^2\right]$$

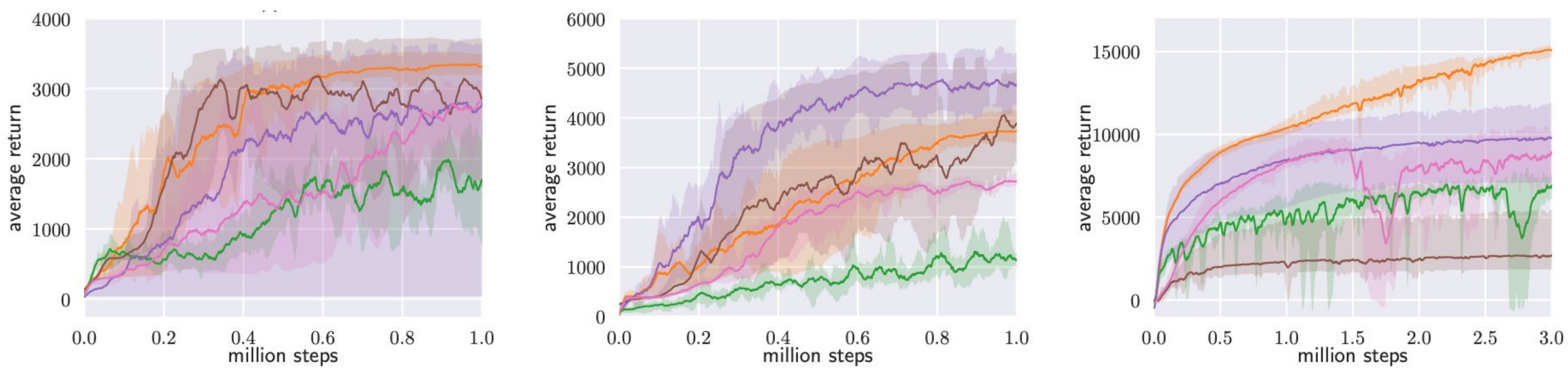
$$\hat{Q}(\mathbf{s}_t,\mathbf{a}_t) = r(\mathbf{s}_t,\mathbf{a}_t) + \gamma\,\mathbb{E}_{\mathbf{s}_{t+1}\sim p}\left[V_{\bar{\psi}}(\mathbf{s}_{t+1})\right]$$

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\phi(\,\cdot\,|\mathbf{s}_t)\,\middle\|\,\frac{\exp\left(Q_\theta(\mathbf{s}_t,\,\cdot\,)\right)}{Z_\theta(\mathbf{s}_t)}\right)\right]$$

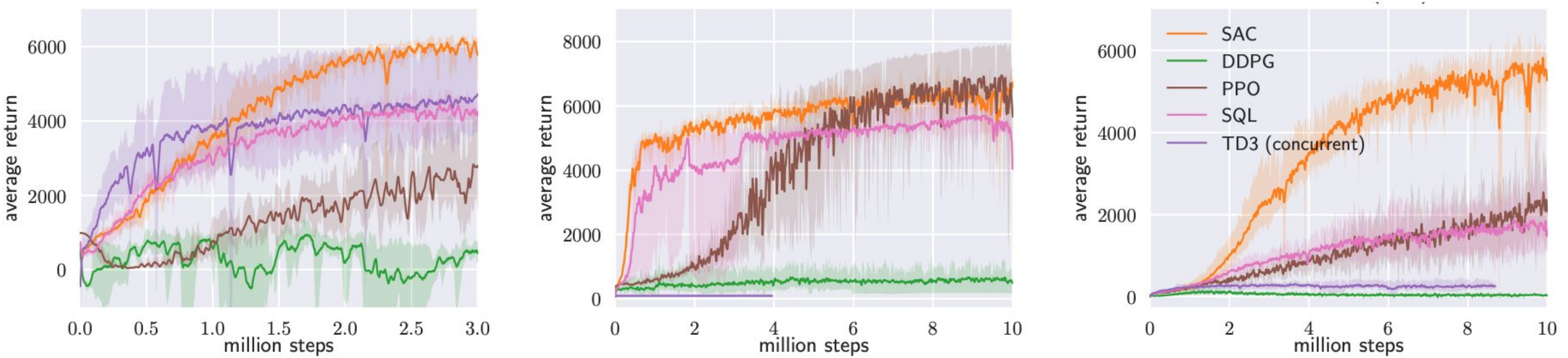
$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t\sim\mathcal{D},\epsilon_t\sim\mathcal{N}}\left[\log\pi_\phi(f_\phi(\epsilon_t;\mathbf{s}_t)|\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, f_\phi(\epsilon_t;\mathbf{s}_t))\right]$$

$$\tilde{a}_\theta(s,\xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s)\odot\xi\right), \qquad \xi\sim\mathcal{N}(0,I).$$

Experiment Result

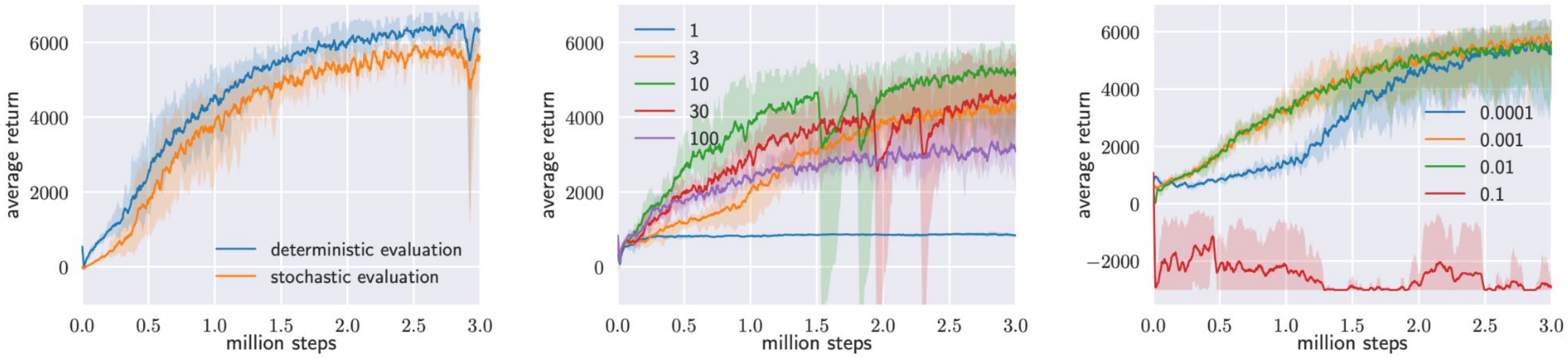


(a) Hopper-v1 (b) Walker2d-v1 (c) HalfCheetah-v1



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

Ablation Study



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