A Note on Soft Actor-Critic

Hsieh Zong-You
Department of Computer Science
National Chiao Tung University
bang0856108.cs08g@nctu.edu.tw

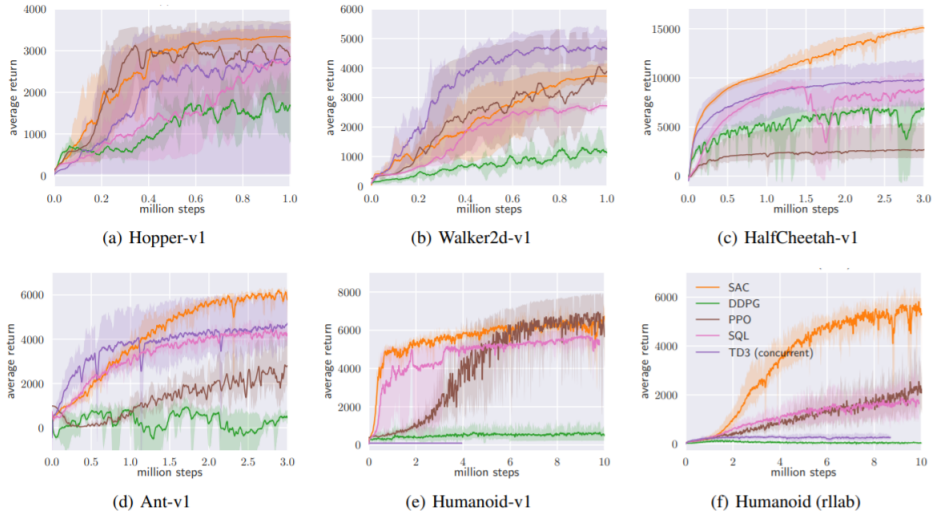
1 Introduction

- The main research challenges tackled by the paper
  Model-free deep reinforcement learning (RL) algorithms have been demonstrated on a range of challenging decision making and control tasks. However, these methods typically suffer from two major challenges: very high sample complexity and brittle convergence properties. First, the reason for the poor sample efficiency of deep RL methods is on-policy learning, it requires new samples to be collected for each gradient step. As the number of gradient steps and samples per step needed to learn an effective policy increases with task complexity, this quickly becomes extravagantly expensive. Second, to avoid the first problem, the combination of off-policy learning and high-dimensional, nonlinear function approximation with neural networks such as DDPG, provides for sample-efficient learning. However, it is notoriously challenging to use due to its extreme brittleness and hyperparameter sensitivity.

- The main contributions of the paper (compared to the prior works)
  Soft actor-critic algorithm incorporates three key ingredients: an actor-critic architecture with separate policy and value function networks, an off-policy formulation that enables reuse of previously collected data for efficiency, and entropy maximization to enable stability and exploration. This algorithm extends readily to very complex, high-dimensional tasks, such as the Humanoid benchmark (Duan et al., 2016) with 21 action dimensions, where off-policy methods such as DDPG typically struggle to obtain good results. SAC also avoids the complexity and potential instability associated with approximate inference in prior off-policy maximum entropy algorithms based on soft Q-learning (Haarnoja et al., 2017). Figure 1. is the comparison of SAC with on-policy and off-policy deep reinforcement learning algorithms. We can see that SAC

Preprint. Under review.

explores to high reward region more fast, and performs more stable than other algorithms.



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

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

Figure 1. Training curves on continuous control benchmarks. Soft actor-critic (yellow) performs consistently across all tasks and outperforming both on-policy and off-policy methods in the most challenging tasks.

- Your personal perspective on the proposed method
  The proposed method combines the statistical technique: maximum entropy to its algorithm. In the paper, there are lots of proofs and explanations for why SAC performs so well. Also, from the experiment results, SAC actually outperforms other classical RL algorithms. However, after reading this paper, I still can't understand why it's not brittle to the hyperparameters since its explanation can't persuade me.

2 Problem Formulation

- Your notations (e.g. MDPs, value functions, function approximators,...etc)
  We address policy learning in continuous action spaces. We consider an infinite-horizon Markov decision process (MDP), defined by the tuple (S, A, p, r), where the state space S and the action space A are continuous, and the unknown state transition probability $p : S \times S \times A \to [0, \infty)$ represents the probability density of the next state $s_{t+1} \in S$ given the current state $s_t \in S$ and action $a_t \in A$. The environment emits a bounded reward $r : S \times A \to [r_{min}, r_{max}]$ on each transition. We will use $\rho_\pi(s_t)$ and $\rho_\pi(s_t, a_t)$ to denote the state and state-action marginals of the trajectory distribution induced by a policy $\pi(a_t|s_t)$.
  Soft state value function:

$$V(s_t) = \mathbb{E}_{a_t \sim \pi}[r(s_t, a_t) - \alpha \log \pi(a_t|s_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho_\pi(s)} V(s_{t+1})]$$
$$= \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)]$$

  Soft action value function:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \rho_\pi(s)}[V(s_{t+1})]$$
$$= r(s_t, a_t) + \gamma \mathbb{E}_{(s_{t+1}, a_{t+1}) \sim \rho_\pi}[Q(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1})]$$

  Function approximators:
  - The policy with parameter θ, π_θ.
  - Soft Q-value function parameterized by w, Q_w.
  - Soft state value function parameterized by ψ, V_ψ; theoretically we can infer V by knowing Q and π, but in practice, it helps stabilize the training.
- The optimization problem of interest
  Standard RL maximizes the expected sum of rewards $\sum_{t=1}^{T} \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi_\theta}}[r(s_t, a_t)$. We will

consider a more general maximum entropy objective, which favors stochastic policies by augmenting the objective with the expected entropy of the policy over $\rho_\pi(s_t)$:

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}[r(s_t,a_t) + \alpha\mathcal{H}(\pi_\theta(.|s_t))]$$

where $\mathcal{H}(.)$ is the entropy measure and α controls how important the entropy term is, known as temperature parameter.
The soft state value function is trained to minimize the mean squared error:

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

$$\nabla_\psi J_V(\psi) = \nabla_\psi V_\psi(s_t)\big(V_\psi(s_t) - Q_w(s_t,a_t) + \log\pi_\theta(a_t|s_t)\big)$$

where D is the distribution of previously sampled states and actions, or a replay buffer.
The soft Q function is trained to minimize the soft Bellman residual:

$$J_Q(w) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}[\frac{1}{2}\big(Q_w(s_t,a_t) - (r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\rho_\pi(s)}[V_{\bar{\psi}}(s_{t+1})])\big)^2]$$

$$\nabla_w J_Q(w) = \nabla_w Q_w(s_t,a_t)\big(Q_w(s_t,a_t) - r(s_t,a_t) - \gamma V_{\bar{\psi}}(s_{t+1})\big)$$

where $\bar{\psi}$ is the target value function which is the exponential moving average (or only gets updated periodically), just like how the parameter of the target Q network is treated in DQN to stabilize the training.
SAC updates the policy to minimize the KL-divergence:

$$\begin{aligned}
\pi_{\text{new}} &= \arg\min_{\pi'\in\Pi} D_{\text{KL}}\Big(\pi'(.|s_t)\|\frac{\exp(Q^{\pi_{\text{old}}}(s_t,.))}{Z^{\pi_{\text{old}}}(s_t)}\Big) \\
&= \arg\min_{\pi'\in\Pi} D_{\text{KL}}\big(\pi'(.|s_t)\|\exp(Q^{\pi_{\text{old}}}(s_t,.) - \log Z^{\pi_{\text{old}}}(s_t))\big) \\
\nabla_\theta J_\pi(\theta) &= \nabla_\theta D_{\text{KL}}\big(\pi_\theta(.|s_t)\|\exp(Q_w(s_t,.) - \log Z_w(s_t))\big) \\
&= \mathbb{E}_{a_t\sim\pi}\Big[-\log\big(\frac{\exp(Q_w(s_t,a_t) - \log Z_w(s_t))}{\pi_\theta(a_t|s_t)}\big)\Big] \\
&= \mathbb{E}_{a_t\sim\pi}[\log\pi_\theta(a_t|s_t) - Q_w(s_t,a_t) + \log Z_w(s_t)]
\end{aligned}$$

Then I stuck here, the paper says that use the reparametrization trick and omit the partition function Z, we can approximate the gradient of policy. But I don't how to get the result in the paper from my above proof.

- The technical assumptions
  Actually, I only find one assumption in this paper. It assumes the action space $|A| < \infty$. The assumption $|A| < \infty$ is required to guarantee that the entropy augmented reward is bounded.

3 Theoretical Analysis

- Proof for soft policy evaluation and soft policy improvement in the paper.

Lemma 1 (Soft Policy Evaluation). *Consider the soft Bellman backup operator $\mathcal{T}^\pi$ in Equation 2 and a mapping $Q^0 : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ with $|\mathcal{A}| < \infty$, and define $Q^{k+1} = \mathcal{T}^\pi Q^k$. Then the sequence Q^k will converge to the soft Q-value of π as $k \rightarrow \infty$.*

Proof. Define the entropy augmented reward as $r_\pi(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[\mathcal{H}\left(\pi(\cdot|\mathbf{s}_{t+1})\right)\right]$ and rewrite the update rule as

$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow r_\pi(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1} \sim p, \mathbf{a}_{t+1} \sim \pi}\left[Q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right] \tag{15}$$

and apply the standard convergence results for policy evaluation (Sutton & Barto, 1998). The assumption $|\mathcal{A}| < \infty$ is required to guarantee that the entropy augmented reward is bounded. □

B.2. Lemma 2

Lemma 2 (Soft Policy Improvement). *Let $\pi_{\text{old}} \in \Pi$ and let π_{new} be the optimizer of the minimization problem defined in Equation 4. Then $Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t) \geq Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t)$ for all $(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{S} \times \mathcal{A}$ with $|\mathcal{A}| < \infty$.*

Proof. Let $\pi_{\text{old}} \in \Pi$ and let $Q^{\pi_{\text{old}}}$ and $V^{\pi_{\text{old}}}$ be the corresponding soft state-action value and soft state value, and let π_{new} be defined as

$$\begin{aligned}\pi_{\text{new}}(\cdot|\mathbf{s}_t) &= \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\cdot|\mathbf{s}_t) \,\|\, \exp\left(Q^{\pi_{\text{old}}}(\mathbf{s}_t, \cdot) - \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right)\right) \\ &= \arg\min_{\pi' \in \Pi} J_{\pi_{\text{old}}}(\pi'(\cdot|\mathbf{s}_t)).\end{aligned} \tag{16}$$

It must be the case that $J_{\pi_{\text{old}}}(\pi_{\text{new}}(\cdot|\mathbf{s}_t)) \leq J_{\pi_{\text{old}}}(\pi_{\text{old}}(\cdot|\mathbf{s}_t))$, since we can always choose $\pi_{\text{new}} = \pi_{\text{old}} \in \Pi$. Hence

$$\mathbb{E}_{\mathbf{a}_t \sim \pi_{\text{new}}}\left[\log \pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right] \leq \mathbb{E}_{\mathbf{a}_t \sim \pi_{\text{old}}}\left[\log \pi_{\text{old}}(\mathbf{a}_t|\mathbf{s}_t) - Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) + \log Z^{\pi_{\text{old}}}(\mathbf{s}_t)\right], \tag{17}$$

and since partition function $Z^{\pi_{\text{old}}}$ depends only on the state, the inequality reduces to

$$\mathbb{E}_{\mathbf{a}_t \sim \pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_{\text{new}}(\mathbf{a}_t|\mathbf{s}_t)\right] \geq V^{\pi_{\text{old}}}(\mathbf{s}_t). \tag{18}$$

Next, consider the soft Bellman equation:

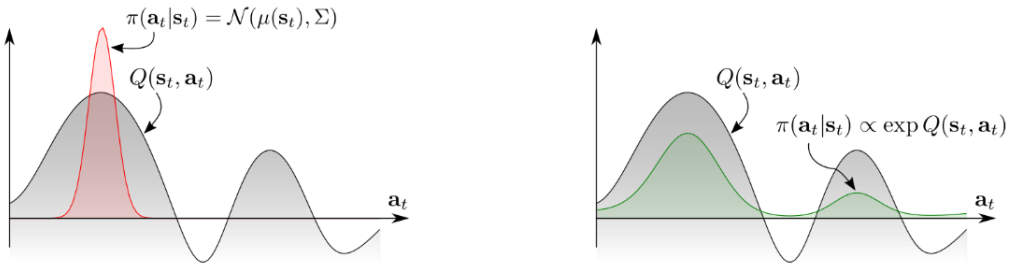
$$\begin{aligned}Q^{\pi_{\text{old}}}(\mathbf{s}_t, \mathbf{a}_t) &= r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[V^{\pi_{\text{old}}}(\mathbf{s}_{t+1})\right] \\ &\leq r(\mathbf{s}_t, \mathbf{a}_t) + \gamma\, \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[\mathbb{E}_{\mathbf{a}_{t+1} \sim \pi_{\text{new}}}\left[Q^{\pi_{\text{old}}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \log \pi_{\text{new}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})\right]\right] \\ &\;\;\vdots \\ &\leq Q^{\pi_{\text{new}}}(\mathbf{s}_t, \mathbf{a}_t),\end{aligned} \tag{19}$$

where we have repeatedly expanded $Q^{\pi_{\text{old}}}$ on the RHS by applying the soft Bellman equation and the bound in Equation 18. Convergence to $Q^{\pi_{\text{new}}}$ follows from Lemma 1. □

Theorem 1 (Soft Policy Iteration). *Repeated application of soft policy evaluation and soft policy improvement to any $\pi \in \Pi$ converges to a policy π^* such that $Q^{\pi^*}(\mathbf{s}_t, \mathbf{a}_t) \geq Q^\pi(\mathbf{s}_t, \mathbf{a}_t)$ for all $\pi \in \Pi$ and $(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{S} \times \mathcal{A}$, assuming $|\mathcal{A}| < \infty$.*

Proof. Let π_i be the policy at iteration i. By Lemma 2, the sequence Q^{π_i} is monotonically increasing. Since Q^π is bounded above for $\pi \in \Pi$ (both the reward and entropy are bounded), the sequence converges to some π^*. We will still need to show that π^* is indeed optimal. At convergence, it must be case that $J_{\pi^*}(\pi^*(\cdot|\mathbf{s}_t)) < J_{\pi^*}(\pi(\cdot|\mathbf{s}_t))$ for all $\pi \in \Pi$, $\pi \neq \pi^*$. Using the same iterative argument as in the proof of Lemma 2, we get $Q^{\pi^*}(\mathbf{s}_t, \mathbf{a}_t) > Q^\pi(\mathbf{s}_t, \mathbf{a}_t)$ for all $(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{S} \times \mathcal{A}$, that is, the soft value of any other policy in Π is lower than that of the converged policy. Hence π^* is optimal in Π. □

- Why maximum entropy works?



A conventional RL approach is to specify a unimodal policy distribution, centered at the maximal Q-value and extending to the neighbouring actions to provide noise for exploration

(red distribution). Since the exploration is biased towards the upper passage, the agent refines its policy there and ignores the lower passage completely. An obvious solution, at the high level, is to ensure the agent explores all promising states while prioritizing the more promising ones. One way to formalize this idea is to define the policy directly in terms of exponentiated Q-values (green distribution):

$$\pi(\mathbf{a}|\mathbf{s}) \propto \exp Q(\mathbf{s}, \mathbf{a})$$

This density has the form of the Boltzmann distribution, where the Q-function serves as the negative energy, which assigns a non-zero likelihood to all actions. As a consequence, the agent will become aware of all behaviours that lead to solving the task, which can help the agent adapt to changing situations in which some of the solutions might have become infeasible. In fact, we can show that the policy defined through the energy form is an optimal solution for the maximum-entropy RL objective, which simply augments the conventional RL objective with the entropy of the policy.

$$\pi^*_{\mathrm{MaxEnt}} = \operatorname*{argmax}_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{T} r_t + \mathcal{H}(\pi(\cdot|\mathbf{s}_t)) \right]$$

4 Conclusion

- The potential future research directions
  Our results suggest that stochastic, entropy maximizing reinforcement learning algorithms can provide a promising avenue for improved robustness and stability, and further exploration of maximum entropy methods, including methods that incorporate second order information (e.g., trust regions (Schulman et al., 2015)) or more expressive policy classes is an exciting avenue for future work.
- Any technical limitations
  Although I don't fully understand the proof and there are still few problems. I think it's OK for me to implement the SAC algorithm to solve RL problems because the author provides the source code online. In addition, the SAC algorithm is not too brittle for the hyperparameter setting. Therefore, I think there may not be huge limitations for me.
- Any latest results on the problem of interest SAC is brittle with respect to the temperature parameter. Unfortunately it is difficult to adjust temperature, because the entropy can vary unpredictably both across tasks and during training as the policy becomes better.As a result, the authors publish another paper called Soft Actor-Critic Algorithms and Applications in the same year. The new paper says that SAC can automatically adjusted the temperature parameter.

References