RL 논문 리뷰 스터디 4기

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

SooHan Kang

2021-05-03

Abstract

Abstract

- Model-free deep reinforcement learning (RL) two major challenges
  - very high sample complexity
  - brittle convergence properties
    - learning rates, exploration constants, and other settings etc…

- In this paper propose “soft actor-critic” an off-policy actor-critic deep RL algorithm based on the maximum entropy reinforcement learning framework.

- Actor aims to maximize expected reward while also maximizing entropy.

- Q-learning methods, off-policy updates + stable stochastic actor-critic formulation

Introduction

Introduction

- RL + Neural Network

- Two major challenges.
  - model-free deep RL methods are notoriously expensive in terms of their sample complexity.
  - Methods are often brittle with respect to their hyperparameters.

- Off-policy learning + Neural network + Continuous state & Action spaces -> exacerbated
  - Separate actor network(DDPG)

Introduction

- How to design an efficient and stable model free deep RL algorithm for continuous state and action spaces ?
  - maximum entropy
- Maximum entropy reinforcement learning alters the RL objective
- The original objective can be recovered using a temperature parameter
- Maximum entropy formulation -> improvement in exploration and robustness
- SAC avoids the complexity and potential instability
  - Soft Q-learning

Preliminaries

Preliminaries

- Standard RL maximizes the expected sum of rewards

$$\sum_t E_{(s_t,a_t)\sim\rho_t}[r(s_t,a_t)]$$

- General maximum entropy objective
  - Stochastic policies by augmenting the objective with the expected entropy of the policy over $\rho_\pi(s_t)$

$$J(\pi) = \sum_{t=0}^{T} E_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \alpha \mathrm{H}\left(\pi(\cdot\,|s_t)\right)\right]$$

$$H(Y|X) = E_{Y|X}[-logP(Y|X)]$$

GO

  - Temperature parameter α
    - If $\alpha = 0$ same with standard

Preliminaries

- This objective has a number of conceptual and practical advantages.

  - Incentivize to explore more widely, while give up on unpromising avenues.

  - Capture multiple modes of near-optimal behavior.
    - In problem settings where multiple actions seem equally attractive, the policy will commit equal probability mass to those actions.

  - Improves learning speed that optimize the conventional RL objective function.(Experimentally)

From soft policy iteration to soft actor-critic

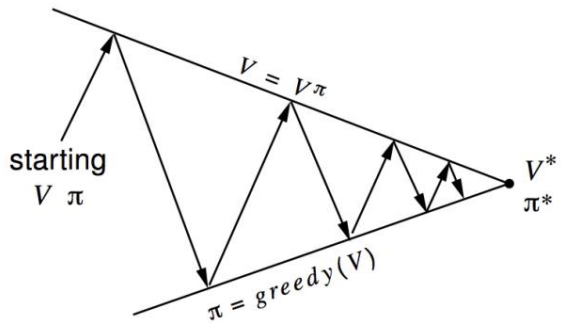
From soft policy iteration to soft actor-critic

- Off-policy soft actor-critic algorithm can be derived starting from a maximum entropy variant of the policy iteration method.

Derivation of Soft Policy Iteration

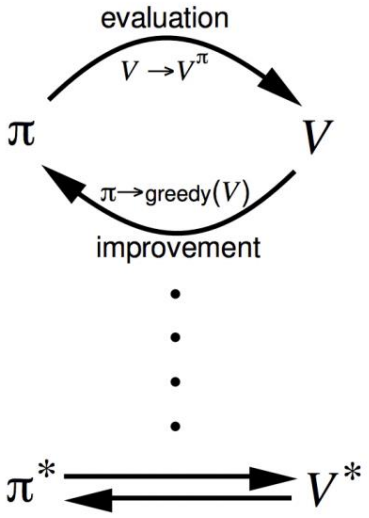
- Deriving soft policy iteration for learning optimal maximum entropy that alternates between policy evaluation and policy improvement in the maximum entropy.

- Policy Iteration



Policy evaluation Estimate v_π
Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
Greedy policy improvement



From soft policy iteration to soft actor-critic

- Policy evaluation step of soft policy iteration
  - Compute the value of a policy π according eq1.
  - Soft Q-value -> modified Bellman backup operator T^{π}

$$\mathrm{T}^{\pi}Q(s_t, a_t) \triangleq r(s_t, a_t) + \gamma E_{s_{t+1}\sim\rho}[V(s_{t+1})]$$

$$where, \qquad V(s_t) = E_{a_t\sim\pi}[Q(s_t, a_t) - log\pi(a_t|s_t)]$$

$$V_{basic}(s_t) = \sum_{a\in A} \pi(a|s) * Q^{\pi}(s, a) = E_{a_t\sim\pi}[Q(s_t, a_t)]$$

$$V(s_t) = E_{a_t\sim\pi}[Q(s_t, a_t)] + \alpha H\big(\pi(\cdot\,|s_t)\big) = E_{a_t\sim\pi}[Q(s_t, a_t) - \alpha log\pi(a_t|s_t)]$$

  - $Q^{k+1} = \mathrm{T}^{\pi}Q^{k}$

  - Then the sequence Q^{k} will converge to the soft Q-value of π

From soft policy iteration to soft actor-critic

- Policy Improvement step of soft policy iteration
  - Update the policy towards the exponential of the new Q-function
  - restrict the policy to some set of policies Π to a parameterized family of distributions such as Gaussians.
  - Turn out to be convenient to use the information projection defined in terms of the Kullback-Leibler divergence.

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

  - Z can be ignored.
  - Let $\pi_{old} \in \Pi$, π_{new} -> $Q^{\pi_{new}}(s_t, a_t) \geq Q^{\pi_{old}}(s_t, a_t)\ for\ all\ (s_t, a_t) \in S\ X\ A$

From soft policy iteration to soft actor-critic

- Soft policy iteration -> soft policy evaluation and soft policy improvement
- it will provably converge to the optimal maximum entropy policy among the policies in Π
- Although this algorithm will provably find the optimal solution, we can perform it in its exact form only in the tabular case.
- Therefore, we will next approximate the algorithm for continuous domains, where we need to rely on a function approximator to represent the Q-values, and running the two steps until convergence would be computationally too expensive.
- The approximation gives rise to a new practical algorithm, called soft actor-critic.

Soft Actor-Critic

- As discussed above, large continuous domains require us to derive a practical approximation to soft policy iteration.
- Parameterized state value function $V_\psi(s_t)$, soft Q-function $Q_\theta(s_t, a_t)$, and a tractable policy $\pi_\varphi(a_t|s_t)$
- State value function approximates the soft value.
- There is no need in principle to include a separate function approximator for the state value, since it is related to the Q-function and policy according to Equation 3.
- Separate function approximator for the soft value can stabilize training and is convenient to train simultaneously with the other networks.

Soft Actor-Critic

- Soft value function : Minimize the squared residual error.

$$J_V(\psi) = \mathbb{E}_{\mathbf{s}_t \sim \mathcal{D}} \left[\frac{1}{2} \left(V_\psi(\mathbf{s}_t) - \mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} \left[Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) \right] \right)^2 \right]$$

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(\mathbf{s}_t) \left(V_\psi(\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t) + \log \pi_\phi(\mathbf{a}_t | \mathbf{s}_t) \right)$$

- D distribution of previously sampled states and actions, or a replay buffer.

- Actions are sampled according to the current policy, instead of the replay buffer.

- Soft Q-function : Minimize the soft Bellman residual error.

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

$$\hat{\nabla}_\theta J_Q(\theta) = \nabla_\theta Q_\theta(\mathbf{a}_t, \mathbf{s}_t) \left(Q_\theta(\mathbf{s}_t, \mathbf{a}_t) - r(\mathbf{s}_t, \mathbf{a}_t) - \gamma V_{\bar{\psi}}(\mathbf{s}_{t+1}) \right)$$

Soft Actor-Critic

- Target value network $V_{\tilde{\psi}}$, where $\tilde{\psi}$ can be an exponentially moving average of the value network weights
  - Asynchronous Methods for Deep Reinforcement Learning (Mnih et al., 2015)

- Policy parameters : Minimizing the expected KL-divergence

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

- Q-function is represented by a neural network and can be differentiated, thus convenient to apply the reparameterization trick.

- To that end, we reparameterize the policy using a neural network transformation.

$$a_t = f_\phi(\epsilon_t; s_t)$$

Soft Actor-Critic

- Where ϵ_t is an input noise vector, sampled from some fixed distribution, such as a spherical Gaussian.

$$a_t = f_\phi(\epsilon_t; s_t)$$

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

$$\Downarrow$$

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

$$KL(p\|q) = -\int p(x) \log g(x) dx + \int p(x) \log p(x) dx$$

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t) + \left(\nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t) - \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t) \right) \nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t)$$

Soft Actor-Critic

- Our algorithm also makes use of two Q-functions to mitigate positive bias in the policy improvement step that is known to degrade performance of value based method.

- we parameterize two Q-functions, with parameters θ_i , and train them independently to optimize $J_Q(\theta_i)$

- We then use the minimum of the Q-functions for the value gradient in Equation 6 and policy gradient in Equation 13.

$$\hat{\nabla}_\psi J_V(\psi) = \nabla_\psi V_\psi(\mathbf{s}_t)\left(V_\psi(\mathbf{s}_t) - Q_\theta(\mathbf{s}_t, \mathbf{a}_t) + \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)\right)$$

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t) + \left(\nabla_{\mathbf{a}_t} \log \pi_\phi(\mathbf{a}_t|\mathbf{s}_t) - \nabla_{\mathbf{a}_t} Q(\mathbf{s}_t, \mathbf{a}_t)\right)\nabla_\phi f_\phi(\epsilon_t; \mathbf{s}_t)$$

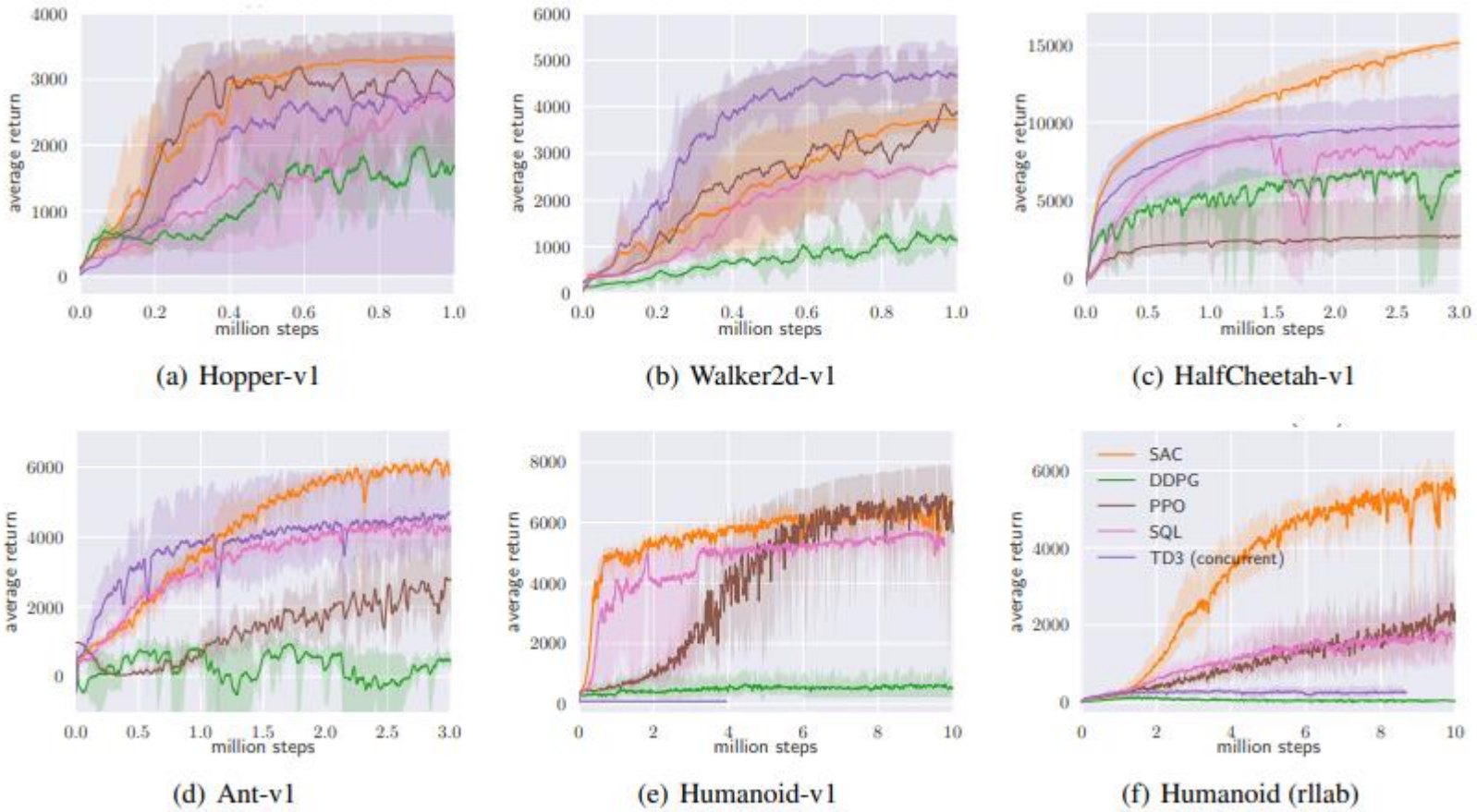
Soft Actor-Critic

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

Experiments

Experiments



(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.

Ablation Study

- Stochastic vs. deterministic policy.

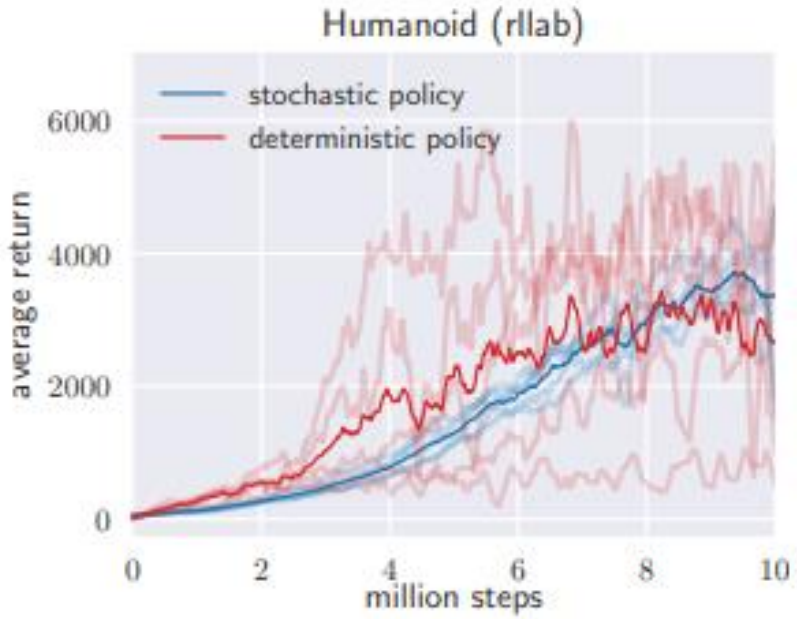


Figure 2. Comparison of SAC (blue) and a deterministic variant of SAC (red) in terms of the stability of individual random seeds on the Humanoid (rllab) benchmark. The comparison indicates that stochasticity can stabilize training as the variability between the seeds becomes much higher with a deterministic policy.

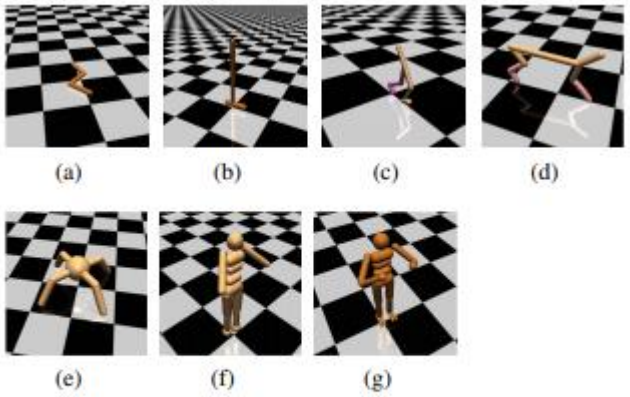
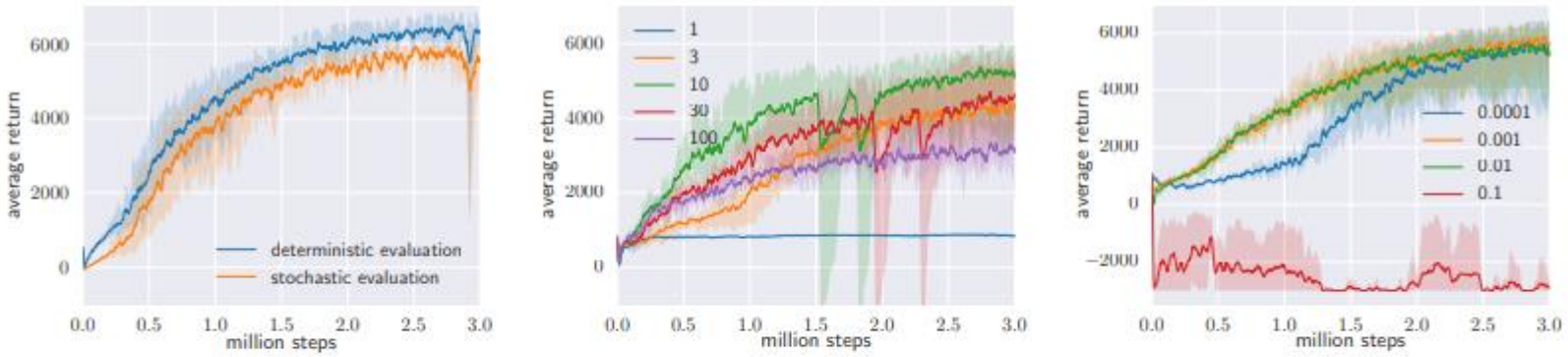


Figure 1. Illustration of locomotion tasks: (a) Swimmer; (b) Hopper; (c) Walker; (d) Half-Cheetah; (e) Ant; (f) Simple Humanoid; and (g) Full Humanoid.

Ablation Study



(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).

Conclusion

Soft Actor-Critic

- Soft actor-critic (SAC) : off-policy maximum entropy deep reinforcement learning algorithm
  - Sample-efficient learning
  - Stability

- Our theoretical results derive soft policy iteration, which we show to converge to the optimal policy.

- In fact, the sample efficiency of this approach actually exceeds that of DDPG by a substantial margin.

Thank you