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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine

***Abstract*—The paper attempts to address very high sample complexity and brittle convergence properties, 2 major challenges in the area of model free RL by proposing the first off-policy actor-critic deep RL algorithm in the maximum entropy framework called *Soft Actor-Critic*, in which the actor aims to maximize entropy along with the expected reward. It outperforms state-of-the-art model-free deep RL methods, including the off-policy DDPG algorithm and the on-policy PPO.**

I. Maximum Entropy RL Framework

The maximum entropy objective function is given by,

$$\mathcal{J}(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(\boldsymbol{s_t},\boldsymbol{a_t})\sim\rho_\pi}[r(\boldsymbol{s_t},\boldsymbol{a_t}) + \alpha\mathcal{H}(\pi(\cdot|\boldsymbol{s_t}))]$$

Temperature parameter α controls the stochasticity of the optimal policy and here, it has been subsumed into the reward by scaling the objective by α^{-1}. The above objective function has been observed to improve exploration, capture multiple modes of near optimal behavior and considerably improve the learning speed.

II. Soft Policy Iteration

The soft actor-critic algorithm is derived from the maximum entropy variant of the policy iteration method, called *soft policy iteration* (proved in tabular setting and extended to continuous setting using function approximation for SAC) for learning optimal maximum entropy policies by alternating between policy evaluation and policy improvement in the maximum entropy framework. The modified Bellman backup operator is given by,

$$\mathcal{T}^\pi\mathcal{Q}(\boldsymbol{s_t},\boldsymbol{a_t}) \triangleq r(\boldsymbol{s_t},\boldsymbol{a_t}) + \gamma\mathbb{E}_{\boldsymbol{s}_{t+1}\sim p}[V(\boldsymbol{s}_{t+1})]$$

The soft value function is given by,

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

- According to *soft policy evaluation* lemma, $\mathcal{Q}^{k+1} = \mathcal{T}^\pi\mathcal{Q}^k$ with $\mathcal{Q}^0 : \mathcal{S}\times\mathcal{A}\rightarrow\mathbb{R}$ and $|\mathcal{A}| < \infty$, $\mathcal{Q}^k$ will converge to the soft Q-value of π as $k\rightarrow\infty$.
- According to *soft policy improvement* lemma, $\mathcal{Q}^{\pi_{\text{new}}}(\boldsymbol{s_t},\boldsymbol{a_t}) \geq \mathcal{Q}^{\pi_{\text{old}}}(\boldsymbol{s_t},\boldsymbol{a_t})\ \forall(\boldsymbol{s_t},\boldsymbol{a_t}) \in \mathcal{S}\times\mathcal{A}$ with $|\mathcal{A}| < \infty$, where $\pi_{\text{old}} \in \Pi$ (restrict the policy to some set of policies) and π_{new} is the optimal minimizer of the KL divergence between the new policy and normalized exponential of action value of the old policy.
- According to *soft policy iteration*, in finite action space, repeated application of *soft policy evaluation* and *soft policy improvement* from any $\pi\in\Pi$ converges to optimal policy π^* s.t $\mathcal{Q}^{\pi^*}(\boldsymbol{s_t},\boldsymbol{a_t}) \geq \mathcal{Q}^\pi(\boldsymbol{s_t},\boldsymbol{a_t})\ \forall\pi\in\Pi$ and $(\boldsymbol{s_t},\boldsymbol{a_t})\in\mathcal{S}\times\mathcal{A}$

III. Soft Actor-Critic

The architecture comprises of 3 function approximators- state value function(soft value) $V_\psi(s_t)$, soft Q-function $\mathcal{Q}_\theta(s_t,a_t)$, and policy $\pi_\phi(a_t|s_t)$. Here, they alternate between optimizing Q-function and the policy networks with stochastic gradient descent. Separate network for the soft value stabilizes training and can be trained simultaneously with the other networks. Experience is collected from the environment and the networks are updated using batches sampled from the replay buffer. Hence, a single environment step is followed by gradient steps. They evaluate the Q-function of the current policy and update the policy through an off-policy gradient update

The soft value function is trained to minimize the expectation of the squared residual error by sampling from the replay buffer and the update rule (unbiased estimator)is computed using the actions sampled according to the current policy, instead of the replay buffer.

The soft Q function is trained to minimize the soft Bellman residual. The target value for the update is computed by maintaining a separate target value network, updated as :

- exponentially moving average, with a smoothing constant τ, to update the target value network weights. (Stabilize training: Large τ can result in instabilities- weights copied more frequently, while small τ can cause training to be slower.)
- copy over the current network weights directly into the target network every 1000 gradient steps (improved performance but increase in computational cost)

Policy network given by $\boldsymbol{a}_t = f_\phi(\epsilon_t,\boldsymbol{s}_t)$ (ϵ_t is an input noise vector from a spherical Gaussian- reparameterization trick) is optimized by minimizing the expected KL divergence as specified before in soft policy iteration extended to continuous space, by sampling the states from the replay buffer. (The partition function is independent of ϕ and can be omitted). The algorithm uses 2 Q-functions to reduce positive bias in the policy improvement step and observed that it significantly speeds up training, especially on harder tasks. The minimum of the Q-functions was used for the value gradient.

The paper suggests that learning a stochastic policy with entropy maximization can drastically stabilize training and that deterministic evaluation can yield better performance (mean of policy distribution). Reward scaling was found to the only hyperparameter requiring tuning(for small values, it fails to exploit reward signals- seems uniform; for large values, fails to explore and gives poor performance in the long run)