From Policy Gradient to Actor-Critic methods

Soft Actor Critic

Olivier Sigaud
with the help of Thomas Pierrot

Sorbonne Université
http://people.isir.upmc.fr/sigaud
Soft Actor Critic: The best of two worlds

- **TRPO and PPO**: $\pi_\theta$ stochastic, on-policy, low sample efficiency, stable
- **DDPG and TD3**: $\pi_\theta$ deterministic, replay buffer, better sample efficiency, unstable
- SAC: “Soft” means “entropy regularized”, $\pi_\theta$ stochastic, replay buffer
- Adds entropy regularization to favor exploration (follow-up of several papers)
- Attempt to be stable and sample efficient
- Three successive versions

References:


Soft Actor-Critic

\textit{SAC} learns a \textbf{stochastic} policy $\pi^*$ maximizing both rewards and entropy:

$$
\pi^* = \arg \max_{\pi_\theta} \sum_t \mathbb{E}_{(s_t,a_t) \sim \rho_{\pi_\theta}} [r(s_t,a_t) + \alpha \mathcal{H}(\pi_\theta(\cdot|s_t))]
$$

- The entropy is defined as: $\mathcal{H}(\pi_\theta(\cdot|s_t)) = \mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)} [- \log \pi_\theta(a_t|s_t)]$
- \textit{SAC} changes the traditional MDP objective
- Thus, it converges toward different solutions
- Consequently, it introduces a new value function, the soft value function
- As usual, we consider a policy $\pi_\theta$ and a soft action-value function $\hat{Q}^{\pi_\theta}$

Soft policy evaluation

- Usually, we define $\hat{V}^\pi_\theta (s_t) = \mathbb{E}_{a_t \sim \pi_\theta (\cdot | s_t)} \left[ \hat{Q}^\pi_\phi (s_t, a_t) \right]$

- In soft updates, we rather use:

$$\hat{V}^\pi_\theta (s_t) = \mathbb{E}_{a_t \sim \pi_\theta (\cdot | s_t)} \left[ \hat{Q}^\pi_\phi (s_t, a_t) \right] + \alpha \mathcal{H} (\pi_\theta (\cdot | s_t))$$

$$= \mathbb{E}_{a_t \sim \pi_\theta (\cdot | s_t)} \left[ \hat{Q}^\pi_\phi (s_t, a_t) \right] + \alpha \mathbb{E}_{a_t \sim \pi_\theta (\cdot | s_t)} \left[ - \log \pi_\theta (a_t | s_t) \right]$$

$$= \mathbb{E}_{a_t \sim \pi_\theta (\cdot | s_t)} \left[ \hat{Q}^\pi_\phi (s_t, a_t) - \alpha \log \pi_\theta (a_t | s_t) \right]$$
Critic updates

▶ We define a standard Bellman operator:

\[
T^\pi \hat{Q}_{\phi}^\pi (s_t, a_t) = r(s_t, a_t) + \gamma \hat{V}_{\phi}^\pi (s_{t+1})
\]

\[
= r(s_t, a_t) + \gamma \mathbb{E}_{a_t \sim \pi(\cdot | s_{t+1})} \left[ \hat{Q}_{\phi}^\pi (s_{t+1}, a_t) - \alpha \log \pi(\theta) (a_t | s_{t+1}) \right]
\]

Critic parameters can be learned by minimizing:

\[
J_Q(\theta) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim D} \left[ \left( r(s_t, a_t) + \gamma \hat{V}_{\phi}^\pi (s_{t+1}) - \hat{Q}_{\phi}^\pi (s_t, a_t) \right)^2 \right]
\]

where \( \hat{V}_{\phi}^\pi (s_{t+1}) = \mathbb{E}_{a \sim \pi(\cdot | s_{t+1})} \left[ \hat{Q}_{\phi}^\pi (s_{t+1}, a) - \alpha \log \pi(\theta) (a | s_{t+1}) \right] \)

▶ Similar to DDPG update, but with entropy
Actor updates

- Update policy such as to become greedy w.r.t to the soft Q-value
- Choice: update the policy towards the exponential of the soft Q-value

\[ J_\pi(\theta) = \mathbb{E}_{s_t \sim D}[KL(\pi_\theta(.|s_t)) \| \frac{\exp(\frac{1}{\alpha} \hat{Q}_{\phi}(s_t, .))}{Z_\theta(s_t)}]. \]

- \( Z_\theta(s_t) \) is just a normalizing term to have a distribution
- SAC does not minimize directly this expression but a surrogate one that has the same gradient w.r.t \( \theta \)

The policy parameters can be learned by minimizing:

\[ J_\pi(\theta) = \mathbb{E}_{s_t \sim D} \left[ \mathbb{E}_{a_t \sim \pi_\theta(.|s_t)} \left[ \alpha \log \pi_\theta(a_t|s_t) - \hat{Q}_{\phi}(s_t, a_t) \right] \right]. \]
Continuous vs discrete actions setting

▶ SAC works in both the discrete action and the continuous action setting

▶ Discrete action setting:
  ▶ The critic takes a state and returns a Q-value per action
  ▶ The actor takes a state and returns probabilities over actions

▶ Continuous action setting:
  ▶ The critic takes a state and an action vector and returns a scalar Q-value
  ▶ Need to choose a distribution function for the actor
  ▶ SAC uses a squashed Gaussian: $a = \tanh(n)$ where $n \sim \mathcal{N}(\mu_\phi, \sigma_\phi)$
Continuous vs discrete actions setting

- In $J_\pi(\theta) = \mathbb{E}_{s_t \sim D} \left[ \mathbb{E}_{a_t \sim \pi_\theta(.|s_t)} \left[ \alpha \log \pi_\theta(a_t|s_t) - \hat{Q}_\phi(s_t, a_t) \right] \right]$
- SAC updates require to estimate an expectation over actions sampled from the actor,
- That is $\mathbb{E}_{a_t \sim \pi_\theta(.|s_t)} [F(s_t, a_t)]$ where $F$ is a scalar function.

- In the discrete action setting, $\pi_\theta(.|s_t)$ is a vector of probabilities
  - $\mathbb{E}_{a_t \sim \pi_\theta(.|s_t)} [F(s_t, a_t)] = \pi_\theta(.|s_t)^T F(s_t, .)$

- In the continuous action setting:
  - The actor returns $\mu_\theta$ and $\sigma_\theta$
  - Re-parameterization trick: $a_t = \tanh(\mu_\theta + \epsilon \sigma_\theta)$ where $\epsilon \sim \mathcal{N}(0, 1)$
  - Thus, $\mathbb{E}_{a_t \sim \pi_\theta(.|s_t)} [F(s_t, a_t)] = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, 1)} [F(s_t, \tanh(\mu_\theta + \epsilon \sigma_\theta))]$
  - This trick reduces the variance of the expectation estimate
  - And allows to backprop through the expectation w.r.t $\theta$
Critic update improvements (from TD3)

- As in TD3, SAC uses two critics $\hat{Q}_{\phi_1}^{\pi\theta}$ and $\hat{Q}_{\phi_2}^{\pi\theta}$
- The TD-target becomes:

$$y_t = r + \gamma \mathbb{E}_{a_{t+1} \sim \pi(s_{t+1} | s_{t})} \left[ \min_{i=1,2} \hat{Q}_{\phi_i}^{\pi\theta}(s_{t+1}, a_{t+1}) - \alpha \log \pi_{\theta}(a_{t+1} | s_{t+1}) \right]$$

And the losses:

$$J(\theta) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim D} \left[ \left( \hat{Q}_{\phi_1}^{\pi\theta}(s_t, a_t) - y_t \right)^2 + \left( \hat{Q}_{\phi_2}^{\pi\theta}(s_t, a_t) - y_t \right)^2 \right]$$

$$J(\theta) = \mathbb{E}_{s \sim D} \left[ \mathbb{E}_{a_t \sim \pi(s_t | s_t)} \left[ \alpha \log \pi_{\theta}(a_t | s_t) - \min_{i=1,2} \hat{Q}_{\phi_i}^{\pi\theta}(s_t, a_t) \right] \right]$$

Automatic Entropy Adjustment

- The temperature $\alpha$ needs to be tuned for each task
- Finding a good $\alpha$ is non trivial
- Instead of tuning $\alpha$, tune a lower bound $H_0$ for the policy entropy
- And change the optimization problem into a constrained one

$$
\begin{align*}
\pi^* &= \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t) \sim \rho_{\pi \theta}} [r(s_t,a_t)] \\
&\text{s.t. } \forall t \mathbb{E}_{(s_t,a_t) \sim \rho_{\pi \theta}} [-\log \pi_{\theta}(a_t | s_t)] \geq H_0,
\end{align*}
$$

- Use heuristic to compute $H_0$ from the action space size

$\alpha$ can be learned to satisfy this constraint by minimizing:

$$
J(\alpha) = \mathbb{E}_{s_t \sim D} [\mathbb{E}_{a_t \sim \pi_{\theta}(\cdot|s_t)} [-\alpha \log \pi_{\theta}(a_t | s_t) - \alpha H_0]]
$$
Practical algorithm

- Initialize neural networks $\pi_\theta$ and $\hat{Q}_{\phi}^{\pi_\theta}$ weights
- Play $k$ steps in the environment by sampling actions with $\pi_\theta$
- Store the collected transitions in a replay buffer
- Sample $k$ batches of transitions in the replay buffer
- Update the temperature $\alpha$, the actor and the critic using SGD
- Repeat this cycle until convergence
Truncated Quantile Critics

- Using a distribution of estimates is more stable than a single estimate
- To fight overestimation bias, TD3 and SAC take the min over two critics
- Truncating the higher quantiles is another option
- No need for two critics
- Better performance than SAC

Any question?

Send mail to: Olivier.Sigaud@upmc.fr
Scott Fujimoto, Herke van Hoof, and Dave Meger.
Addressing function approximation error in actor-critic methods.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine.
Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al.
Soft actor-critic algorithms and applications.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov.
Controlling overestimation bias with truncated mixture of continuous distributional quantile critics.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu.
Asynchronous methods for deep reinforcement learning.
Practical implementation of neural critics

- $\hat{V}^\pi_\phi(s)$ is smaller, but not necessarily easier to estimate
- Given the implicit max in $\hat{V}^\pi_\phi(s)$, approx. may be less stable than $\hat{Q}^\pi_\phi(s, a)$ (?)
- Note: a critic network provides a value even in unseen states