A Brief Note on Deep Energy-Based Policies

YuChun Chien
Department of Computer Science
National Yang Ming Chiao Tung University
michael.cs08@nycu.edu.tw

1 Introduction

Learning the best way to solve a task seems to bring up one of the biggest problems in the reinforcement learning industry. A standard deep RL algorithm aims to master a single way to solve a given task, to which the solution typically is the first way that seems to work well. Tang and Haarnoja [2017] With standard deep RL algorithms, two high-rewarded policies could still act very differently. That's because once an agent gains a good reward, it will keep exploring the same path. Hence, it turns out that the policy is vulnerable to environmental changes.

To tackle this problem, a very impressive concept came out–Maximum-Entropy Policy. This technique allows an agent to try many different ways to solve a task while training. Therefore, the agent can adapt to changing situations where some solutions might have become infeasible.

By accomplishing maximum-entropy learning, this paper (Haarnoja et al. [2017]) shows how the policy is defined through the energy form, becoming an optimal solution for the maximum-entropy RL.

2 Problem Formulation

Please present the formulation in this section. You may want to cover the following aspects:

- Your notations (e.g. MDPs, value functions, function approximators,...etc)
- The optimization problem of interest
- The technical assumptions
- Preliminaries

2.1 Infinite-Horizon MDP

- consist of the tuple $(\mathcal{S}, \mathcal{A}, P, r)$
- state space $\mathcal{S}$ and action space $\mathcal{A}$ are continuous
- state transition probability $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$
- reward on each transition $r : \mathcal{S} \times \mathcal{A} \to [r_{\min}, r_{\max}]$
- To simplify notation, we denote $r_t \triangleq r(s_t, a_t)$

2.2 Maximum Entropy Reinforcement Learning

Learning policy using standard reinforcement learning,

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{s_t \sim d_\mu^{\pi_{\mathbf{new}}}, a_t \sim \pi_{\mathbf{new}}(\cdot|s_t)}[r(s_t, a_t)] \tag{1}$$

Learning policy using maximum entropy reinforcement learning,

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

$$where \;\; \mathcal{H}(\pi(\cdot|s_t)) = \mathbb{E}_{a_t \sim \pi}[-\log(\pi(a_t|s_t))]$$

where α is a parameter that can be used to determine the importance of the entropy.

2.3 Energy-Based Models

For complex, multimodal behaviors, we use energy-based policies of the form

$$\pi(a_t|s_t) \propto \exp(-\varepsilon(s_t, a_t)) \tag{2}$$

where ε is a function approximator of energy function, which we can represented using deep neural network.

From the theorem below, we can show that there is a close connection between energy-based models and soft Q-functions, representing as $\varepsilon(s_t, a_t) = -\frac{1}{\alpha} Q_{soft}(s_t, a_t)$.

Theorem 1 *Let soft Q-function be*

$$Q^*_{soft}(s_t, a_t) = r_t + \mathbb{E}_{(s_{t+1}, \ldots) \sim d_\mu^\pi} \left[\sum_{l=1}^{\infty} \gamma^l (r_{t+l} + \alpha \mathbb{H}(\pi^*_{MaxEnt}(\cdot|s_{t+l}))) \right] \tag{3}$$

and soft value function be

$$V^*_{soft}(s_t) = \alpha \log \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*_{soft}(s_t, a') da' \right) \tag{4}$$

Then the optimal policy would be

$$\pi^*_{MaxEnt}(a_t|s_t) \propto \exp\left(\frac{1}{\alpha} \left(Q^*_{soft}(s_t, a_t) - V^*_{soft}(s_t) \right) \right) \tag{5}$$

Theorem 2 *The soft Bellman equation can be written as*

$$Q^*_{soft}(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{P}(S_t)} \left[V^*_{soft}(s_{t+1}) \right] \tag{6}$$

3 Theoretical Analysis

3.1 Soft Q-Iteration

By iterating the process of updating estimates of V_{soft} and Q_{soft}, they will eventually converge to V^*_{soft} and Q^*_{soft}.

Theorem 3 *Soft Q-Iteration. Let $Q_{soft}(\cdot, \cdot)$ and $V_{soft}(\cdot)$ be bounded. Assuming that $\int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*_{soft}(s_t, a') da' \right) < \infty$ and $Q^*_{soft} < \infty$ exist. Then the fixed-point iteration*

$$Q_{soft}(s_t, a_t) \leftarrow r_t + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{P}(S_t)} \left[V_{soft}(s_{t+1}) \right], \forall s_t, a_t \tag{7}$$

$$V_{soft}(s_t) \leftarrow \alpha \log \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q_{soft}(s_t, a') da' \right), \forall s_t \tag{8}$$

converges to Q^*_{soft} and V^*_{soft}, respectively.

3.2 Problems in Practice

Here, we discuss how to implement Bellman backup mentioned above in a practical algorithm. There are two intractable problems.

- The soft Bellman backup cannot be performed exactly in continuous or large state and action spaces.
- Sampling from the energy-based model (eqation 5) is not easy in general.

3.2.1 Soft Q-Learning

To address the first problem, we can convert Theorem 3 into a stochastic optimization problem. Thus, it leads to a stochastic gradient descent update procedure.

First, we express the soft value function in terms of an expectation via importance sampling:

$$V^{\theta}_{soft}(s_t) = \alpha \log \mathbb{E}_{q_{a'}} \left[\frac{\exp\left(\frac{1}{\alpha} Q^{\theta}_{soft}(s_t, a')\right)}{q_{a'}}(a') \right] \tag{9}$$

where $q_{a'}$ is an arbitrary distribution over the action space and θ is the approximator parameters.

Second, we define the loss function as follows:

$$J_Q(\theta) = \mathbb{E}_{s \sim q_{s_t}, a \sim q_{a_t}} \left[\frac{1}{2} \left(\hat{Q}^{\bar{\theta}}_{soft}(s_t, a_t) - Q^{\theta}_{soft}(s_t, a_t) \right)^2 \right] \tag{10}$$

where the target Q-value is

$$\hat{Q}^{\bar{\theta}}_{soft}(s_t, a_t) = r_t + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{P}(S_t)} \left[V^{\bar{\theta}}_{soft}(s_{t+1}) \right] \tag{11}$$

3.2.2 Sampling

By sampling from energy-based model, there are several approximate techniques. First, we can use Markov chain Monte Carlo (MCMC) based sampling (Sallans and Hinton [2004]). Since the MCMC is not feasible for on-policy training, the authors resorted to sampling network based on Stein variational gradient descent (SVGD) (Liu and Wang [2016]). The network is trained to generate approximate samples.

4 Conclusion

Please provide succinct concluding remarks for your report. You may discuss the following aspects:

- The potential future research directions
- Any technical limitations
- Any latest results on the problem of interest

Soft Q-learning is a marvelous RL method. An amazing work Tang and Haarnoja [2017], presented by BAIR (Berkeley Artificial Intelligence Research), shows the composition ability of two soft policies. The combined policy is approximately optimal for the combined task. This result is quite shocking since the standard RL method is totally not capable of composition.

Also, this paper takes the SVGD approach to generate approximate samples. It has the potential to be implemented in the actor-critic method. In fact, some research did accomplished this work (Haarnoja et al. [2018a], Haarnoja et al. [2018b]).

References

Haoran Tang and Tuomas Haarnoja, 2017. URL `https://bair.berkeley.edu/blog/2017/10/06/soft-q-learning/`.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017.

Brian Sallans and Geoffrey E Hinton. Reinforcement learning with factored states and actions. *The Journal of Machine Learning Research*, 5:1063–1088, 2004.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018a.

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. *arXiv preprint arXiv:1812.05905*, 2018b.