Twin-Delayed Deep Deterministic Policy Gradient (TD3)

Anonymous Authors[1]

Abstract

blar

1. Paper Critique

(Fujimoto et al., 2018) (Haarnoja et al., 2018) (Nachum et al., 2018)

2. Implementation Results

3. Implementation Goal

I have chosen to implement the reinforcement learning scheme suggested by (**?**) for my final project. This off-policy algorithm, called Twin-Delayed Deep Deterministic Policy Gradient (TD3) is a powerful approach to continuous control tasks. My goal is to implement the algorithm itself (using my own implementation approach), and test in on various toy problems provided through AIGym. Ideally, I would like to apply to the learning framework to MuJoCo robotics tasks. Through these toy problems I hope to get a better sense of hyperparameter tuning, and understand at a higher level the stochasticity of the underlying learning framework. Lastly, I want to see how customizations, such as input standardization, can affect learning performance.

4. Implementation Plan

Many implementations exist online for TD3. I would like to create my own, using two specific implementations as references. In particular, the authors provide a version of their code on Github[1], as well as a version from OpenAI Spinning Up[2] – an open source educational tool created by OpenAI. While I will use their code as a reference, I will write my own version of both the learning scheme, and the script for training the algorithm. Having implemented my own version of this algorithm, I will test it on a variety of baseline toy problems. I would like to include some classical control problems, such as the inverted pendulum, as well as robotics problems provided by MuJoCo. By using standard toy problems, I can evaluate my implementation and hyperparameters to baseline results published by both (**?**) and by OpenAI Spinning Up, to verify my implementation works as intended.

5. Preliminary Implementation

I have implemented my version of TD3, and tested it with some basic problems. In particular, I have used the inverted pendulum problem to assess how well the learning performs. My implementation is based on OpenAI Spinning Up's tensorflow version of TD3, with significant changes in the overall code architecture and runner script. My implementation seems to be performing as intended, but more a more rigorous analysis will be needed to determine the efficacy of my code. For now, I am just using the hyperparamters and network structure suggested by the paper, but longer term I'm hoping to analyze those hyperparameters to see which parameters the algorithm is most sensitive to.

6. Preliminary Results

To demonstrate preliminary results, consider the classical controls problem of the inverted pendulum. The agent is tasks with spinning up and balancing the inverted pendulum. A screenshot from one arbitrary trial is depicted in Figure **??**. At first, the agent is unable to perform this task. However, through training, the agent quickly learns an effective policy, and is able to spin up and balance the pendulum. This learning can be visualized in two ways. First, a running average for future discounted reward over episodes in plotted in Figure **??**. The immediate upward trend shows that the learning algorithm is working (though future work will be needed to analyze how well it works). Next, the loss function values for the actor and critic networks over training are plotted in Figure **??**. These plots show convergence toward zero for the loss functions, indicating both networks are being trained properly.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]https://github.com/sfujim/TD3

[2]https://github.com/openai/spinningup/

References

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/pdf/1802.09477.pdf.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcementlearning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/pdf/1801.01290.pdf.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 3303–3313, 2018. URL https://proceedings.neurips.cc/paper/2018/file/e6384711491713d29bc63fc5eeb5ba4f-Paper.pdf.