DS8010: Interactive Learning in Decision Process

Course Project: Evaluation of Proximal Policy Optimization Algorithms

Shansong (Sam) Huang and Qinyun (Peter) Yu

ID: 500715634 and 501137007

March 4th, 2022

In the past couple years, research in reinforcement learning has furthered through combined efforts of neural networks and Q-learning [Mnih et al, (2015)]. Despite such efforts, traditional issues like convergence failure, lack of scalability and robustness plague such approaches. One of the newer approaches is through Policy Gradient (PG) methods (e.g., REINFORCE) which learn parametrized policies that select actions without consulting value functions [Sutton and Barto, (2020)]. One recent advanced version of such an approach is the Proximal Policy Optimization (PPO). By tweaking the original objective function (below) that optimizes the neural network:

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

and by introducing stabilizing properties like a penalty component (KL penalty coefficient), and altering the reward function, the clipped surrogate objective is formulated:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

which results in an algorithm that allows for simplicity, as well as achieving greater performance than previous online policy gradient methods [Schulman et al, (2017)].

Essentially, through the aforementioned tweaks and other optimizations, this objective function "clips" the amount of change you can make at each episode and allows for minimization to reverse poorly performing actions. In addition to the above changes, through such clipping function, gradient ascent is seen as possible, whereas before such action would result in massive deviations in policy updates, allowing for greater scalability and more "value" for less data.

We plan on replicating the previously mentioned PPO algorithm and its paper in Python with Tensorflow and testing its performance and tweaking its hyperparameters under available Gym environments and matching the paper's results (e.g., Atari games, and various benchmarks).

REFERENCES:

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Sutton, R. S., & Barto, A. G. (2020). *Reinforcement learning: An introduction*. pg 73-78.The MIT Press.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017, August 28). *Proximal policy optimization algorithms*. arXiv.org. https://arxiv.org/abs/1707.06347