Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov

***Abstract*—The paper proposes an algorithm derived from TRPO (*trust region policy optimization*) which is more robust, has simpler implementation and better sample complexity (data efficiency). The algorithm alternates between sampling data from the environment, and optimizing a *surrogate* objective function using *stochastic gradient ascent*.**

I. Objective Functions

Policy gradient methods : It computes an estimation of the policy gradient and uses it in the stochastic gradient ascent algorithm. The learning objective is given by :

$$L^{PG}(\theta) = \hat{\mathbf{E}}_t\Big[\log \pi_\theta(a_t|s_t)\hat{A}_t\Big]$$

where π_θ denotes the stochastic policy and A_t is the estimation of the advantage function at time t.

Trust Region Methods : TRPO has a *surrogate* objective which is maximized subject to a constraint on the size of the policy update (Solved using conjugate gradient algorithm; linear approximator for objective and quadratic approximator for constraint). The proposed algorithm converts the constraint into a penalty and solves the unconstrained optimization problem :

$$\underset{\theta}{\text{maximize}}\hat{\mathbf{E}}_t\Big[r_t(\theta)\hat{A}_t - \beta\text{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]\Big] \qquad (1)$$

where $r_t(\theta)$ denotes the probability ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. TRPO uses a hard constraint instead of a penalty because of the difficulty involved in choosing β which works for different problems or single problem with changing characteristics.

Note : "certain surrogate objectives computing max KL over states forms a lower bound on the performance of the policy".

Clipped Surrogate Objective : Upon modifying the *conservative policy iteration* (CPI) objective ($\hat{\mathbf{E}}_t[r_t(\theta)\hat{A}_t]$), to penalize changes to the policy that move $r_t(\theta)$ away from $1(r_t(\theta_{old}))$, the probability ratio was clipped as :

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

forming a lower bound (pessimistic bound) of the unclipped objective (L^{CPI}) (L^{CLIP} has a penalty for too large of a policy update).

Adaptive KL Penalty Coefficient : Using Equation (1) as the KL-penalized objective ($L^{KLPEN}(\theta)$), a KL divergence d is calculated as

$$d = \hat{\mathbf{E}}_t[\text{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]]$$

and compared to a target KL divergence d_{targ} before each policy update inorder to modify β as

$$\beta = \begin{cases}\beta/2 \text{ If } d < d_{targ}/1.5\\ 2\beta \text{ If } d > d_{targ} \times 1.5\end{cases}$$

It is to be noted that the algorithm is not very sensitive to the initial values of β and the values 2 and 1.5. However, this objective was found to perform worse than the clipped surrogate objective

II. PPO Algorithm

The algorithm performs gradient ascent on the objectives L^{CLIP} and L^{KLPEN}. When using in the context of a neural network which shares parameters between policy and value function, the total loss is give by :

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbf{E}}_t[L_t^{CLIP}(\theta) + c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)(\theta)]$$

where VF refers to value function error $(V_\theta(s_t) - V_t^{targ})^2$ and S refers to the entropy bonus (for sufficient exploration). The truncated version of generalized advantage estimation is given by,

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}$$

where T denotes the trajectory length and $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. In the context of *Actor-Critic* algorithm, at each iteration PPO uses N(parallel) actors to collect T timesteps of data, constructs the surrogate objective and optimises it using mini-batch SGD.

III. Experiments

The three objectives considered were :

1 No clipping or penalty : CPI
2 Clipping : Clipped Surrogate Objective
3 KL penalty (fixed or adaptive β)

The experiments were run without sharing the parameters between the policy and value function and without using an entropy bonus, on 7 simulated robotics tasks implemented in OpenAI Gym using MuJoCo physics engine. The scores for each environment was shifted and scaled so that the random policy gave a score of 0 and the best result was set to 1. It was observed that the objective (1) gave a negative average score whereas objective (2) gave the best average normalised score. Objective (3) performed worse that (2) but much better than (1). Upon experimenting in continuous domain, PPO outperformed previous methods such as TRPO, Cross-Entropy method, Policy gradient with adaptive step size, A2C and A2C with trust region on almost all the continuous control environments. The authors also give results on the performance of PPO in high-dimensional continuous control problems such as 3D humanoid control tasks, using Roboschool.

References

[1] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).