Summary of the paper:

Proximal Policy Optimization Algorithms

Ignacio Monardes

January 19, 2026

Referencia

Authors: John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
Title: *Proximal Policy Optimization Algorithms*
Conference/Journal:
Link: `https://arxiv.org/pdf/1707.06347`

1 Motivation

Current Problem

TRPO is very complex to implement in many scenarios and bad with noisy environments. Deep Q-Learning is difficult is not very good on many simple and continuous tasks.

There is room for a robust, scalable and efficient model.

Relevance of the problem

An scalable algorithm for many environments and easy to implement.

Hole in the literature / Previous limitations

2 Main Idea

Reinforce Gradient

First of all, the expected value to maximize is:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)] = \sum_{\tau} p_\theta(\tau) R(\tau)$$

We can notice that:

$$\nabla_\theta p(\tau) = p(\tau) \nabla_\theta \log p(\tau)$$

Demonstration:

$$\nabla_\theta \log p(\tau) = \frac{\nabla_\theta p(\tau)}{p(\tau)} \Rightarrow \nabla_\theta p(\tau) = p(\tau) \nabla_\theta \log p(\tau)$$

The probability of a trajectory τ is defined by:

$$
\begin{aligned}
p_\theta(\tau) &= p(s_0)\prod_t \pi_\theta(a_t \mid s_t)p(s_{t+1} \mid s_t, a_t) \\
\log p_\theta(\tau) &= \log p(s_0) + \sum_t \log \pi_\theta(a_t \mid s_t) + \sum_t \log p(s_{t+1} \mid s_t, a_t) \\
\nabla_\theta \log p_\theta(\tau) &= \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)
\end{aligned}
$$

Then:

$$
\nabla_\theta J(\theta) = \mathbb{E}\left[\sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) G_t\right]
$$

We can make rest a bias and define the advantage $A(s_t, a_t) := Q(s_t, a_t) - V(s_t)$, resulting in:

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

TRPO

TRPO maximizes this problem:

$$
\begin{aligned}
\max_\theta \quad & \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] \\
\text{subject to} \quad & \hat{\mathbb{E}}_t[KL[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta
\end{aligned}
$$

with a hard constraint over the restriction.

Theorically we could use the restriction on the function with a coefficient β:

$$
\max_\theta \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta KL[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]
$$

this would make a lower bound of the value.

The problem here is that β shouldn't be fixed and with additional modfications to Stochastic Gradient Descent.

Innovation: CLIP, Adaptative KL

We define the ratio $r_t(\theta) := \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$.

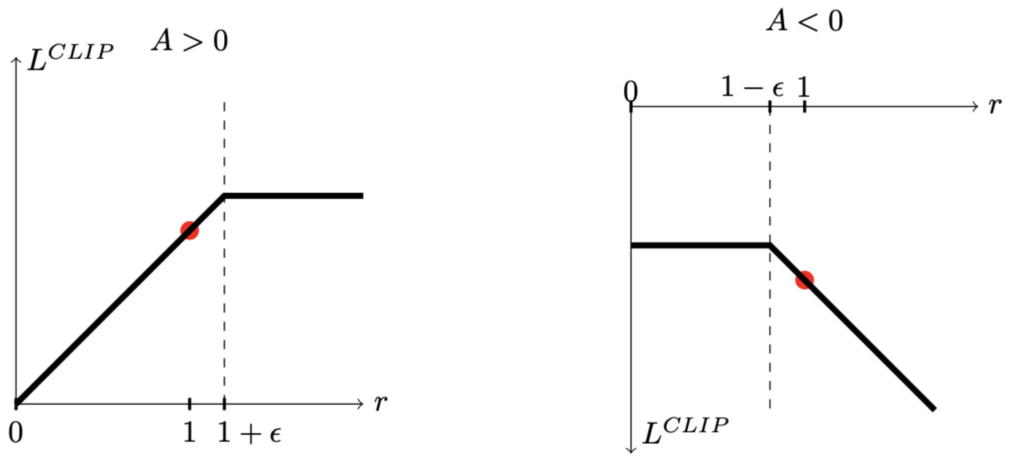
TRPO maximizes a surrogate objective:

$$
L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t[r_t(\theta)\hat{A}_t]
$$

where CPI refers to *Conservative Policy Iteration*. But the gradient could be very large. So we clipped it.

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

This allows to move the gradient in no more than the range $[1-\varepsilon, 1+\varepsilon]$, and as we take the minnimum we create a lower bound.


L^{CLIP}
$A > 0$
0
1
$1 + \epsilon$
r
$A < 0$
0
$1 - \epsilon$
1
r
L^{CLIP}

3 Methodology

Modeling

Assumptions

Architecture

4 Experiments

Environments or datasets

Baselines

Settings

New Scenarios

5 Results

¿En qué escenarios funciona mejor / peor?

6 Discusión Crítica

Fortalezas

Debilidades

Supuestos cuestionables

Qué no queda claro

7 Conclusiones

Comentario Personal

¿Es una buena contribución?

¿Lo usaría en mi investigación?