Proximal Policy Optimization

RL FOR EFFICIENT HIGH DIMENSIONAL CONTROL

By Luke Ditria

github.com/LukeDitria
youtube.com/@LukeDitria
in/lukeditria

MOTIVATION….

PROBLEMS WITH "VANILLA" ACTOR CRITIC METHODS

- Value estimate used for calculating Advantage can introduce bias.
  - Slow to converge
  - May not converge at all
- Can't perform many update steps between rollouts
  - Overfit Value function
  - Change action probabilities too much

Basic Actor Critic

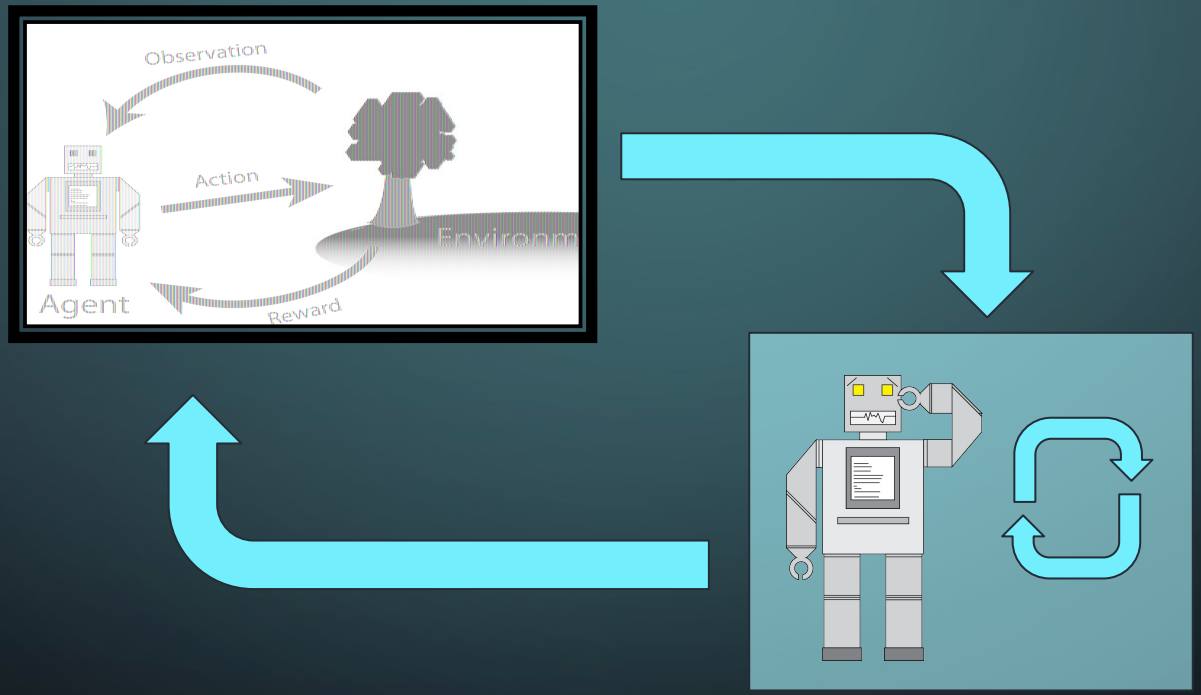
$$A_t = R_t - V_t$$

$$L_\theta^{PG} = \mathbb{E}_\pi \left[log(\pi_\theta(s_t, a_t)) A_t \right]$$

IMPROVEMENTS TO MAKE…

- Trust Region Policy Optimisation (TRPO)
  - Limit how much our current policy can differ from the old policy during training step
- Generalized Advantage Estimation (GAE)
  - Balancing noise and bias when calculating Advantage
- Proximal Policy Optimization (PPO)
  - Easy to implement approximation of TRPO

Perform rollouts and collect data with “old” policy



Use data collected to train “new” policy

TRPO

- *Trust Region Policy Optimization. J.Schulman et. al*

- Previous work shows that it's theoretically possible to ensure monotonic improvement by maximising a certain objective.
- However for "real-world" problems we have estimations of certain values.
- For parameterised policies we need to know "how much" of a change we can make to our policy

TRPO

- TRPO introduces a “Trust Region”
  - Only maximize the objective IF the constraint can be satisfied
    - AKA change the size of your steps to ensure that your policy does not change too much!

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

GAE

- *High-dimensional continuous control using generalized advantage estimation,* J.Schulman et. al

- GAE creates an estimate of the Advantage that is less noisy be combining Advantage you get by using the returns approximation from TD learning and "real" returns update.

$$\hat{\mathbf{A}}_t^{GAE(\gamma,\tau)} = \sum_{l=0}^{\infty} (\gamma\tau)^l \delta_{t+1}^V$$

GAE

$$\hat{\mathbf{A}}_t^{GAE(\gamma,\tau)} = \sum_{l=0}^{\infty} (\gamma\tau)^l \delta_{t+1}^V$$

$$\delta_{t+1}^V = r_t + \gamma \mathbf{V}(s_{t+1}) - \mathbf{V}(s_t)$$

$$\hat{\mathbf{A}}_t^{GAE(\gamma,0)} = r_t + \gamma \mathbf{V}(s_{t+1}) - \mathbf{V}(s_t)$$

$$\hat{\mathbf{A}}_t^{GAE(\gamma,1)} = \sum_{l=0}^{\infty} \gamma^l r_{t+l} - \mathbf{V}(s_t)$$

Trust region for Value function

$$\underset{\phi}{\text{minimize}} \quad \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2$$

$$\text{subject to} \quad \frac{1}{N} \sum_{n=1}^{N} \frac{\|V_\phi(s_n) - V_{\phi_{\text{old}}}(s_n)\|^2}{2\sigma^2} \leq \epsilon.$$

PPO

- *Proximal Policy Optimization Algorithms. J.Schulman et. al*

- TRPO “Trust region” is computationally expensive and hard to implement.
    - PPO offers an easy to implement alternative that in some cases performs even better! (the paper actually gives a few alternatives but we’ll focus on the best/most popular)

PPO

- We use the fact that the ratio of the new/old policy is an indication of how far the new policy has moved from the old.
- Force an explicit constraint of the objective if the ratio moves outside of a given range.

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

TRPO

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

PPO

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

PPO