Introduction to RL Algorithm:TRPO and derivatives

Zhang Tingyu

16th March 2025

1 Introduction

For the prevailing LLM deepseek R1, it uses Reinforcement Learning(RL) method to do supervised fine-tuning(SFT). Here is the optimization formula of Deepseek R1(GRPO):

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)\right] =$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_t|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}$$

After the lecture, I hope you will feel that this formula is not that terrifying.

2 Preliminaries

There are some notation to build the model

Markov reward process($\langle \mathcal{S}, \mathcal{P}, r, \gamma \rangle$) $\rightarrow$ Markov decision process($\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$).

$\mathbb{E}[R_t|S_t = s] = r(s)$: reward for getting into the state s.

Policy : $\pi(a|s) = \mathbb{P}(A_t = a|S_t = s)$ $\begin{cases} \text{deterministic policy : one action each time .} \\ \text{stochastic policy : distribution of an action .} \end{cases}$

In a Markov Reward Process, the sum of discounted rewards from state S_t at time t until the terminal state is called the **Return(r.v.)**. The formula is as follows(R_{t+k} is the reward at time $t+k$, $\gamma \in (0, 1]$.):

$$G_t = \sum_{k=0}^{\infty}\gamma^k R_{t+k}. \tag{1}$$

We use $V^\pi(s)$ to denote the **state-value function** for a policy πin a Markov Decision Process (MDP). It is defined as the expected return when starting from state s and following policy π. Mathematically, it is expressed as:

$$V^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]. \tag{2}$$

The transition matrix P^{π} for state-action pairs under a stationary policy π. Specifically:

$$P^{\pi}_{(s,a),(s',a')} = P(s'|s,a)\pi(a'|s'). \tag{3}$$

Remark 1 It is always important to ask yourself which notation is a number, and which is a r.v.

3 Build the model

Define the optimal function of the parameter θ:

$$\begin{aligned}
J(\theta) &= \mathbb{E}_{s_0}[V^{\pi_\theta}(S_0)](\text{Start from arbitrary state}) \\
&= \mathbb{E}_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t V^{\pi_\theta}(s_t) - \sum_{t=1}^{\infty}\gamma^t V^{\pi_\theta}(s_t)\right](\text{For the unified form}) \\
&= -\mathbb{E}_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t(\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t))\right](\text{Absolutely convergence})
\end{aligned}$$

Based on the above equations, we can derive the difference between the objective functions of the old and new policies:

$$\begin{aligned}
J(\theta') - J(\theta) &= \mathbb{E}_{s_0}[V^{\pi_{\theta'}}(s_0)] - \mathbb{E}_{s_0}[V^{\pi_\theta}(s_0)] \\
&= \mathbb{E}_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] + E_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t(\gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t))\right] \\
&= \mathbb{E}_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t[r(s_t,a_t) + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)]\right] \\
&= \mathbb{E}_{\pi_{\theta'}}\left[\sum_{t=0}^{\infty}\gamma^t A^{\pi_\theta}(s_t,a_t)\right] \\
&= \sum_{t=0}^{\infty}\gamma^t \mathbb{E}_{s_t\sim P_t^{\pi_{\theta'}}}\mathbb{E}_{a_t\sim\pi_{\theta'}(\cdot|s_t)}[A^{\pi_\theta}(s_t,a_t)] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim\mathcal{V}^{\pi_{\theta'}}}\mathbb{E}_{a\sim\pi_{\theta'}(\cdot|s)}[A^{\pi_\theta}(s,a)].
\end{aligned}$$

Set $A^{\pi_\theta}(s_t,a_t) \triangleq r(s_t,a_t) + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t)$.(In RL, it is Time Difference)

The second-to-last equation:Rewrite the s_t sampling from a distribution determined by $\pi_{\theta'}$.

And this actually is :$\mathbb{E}_{s_t\sim P_t^{\pi_{\theta^*}}}(s_t) \triangleq \sum_{s_t\in\mathcal{S}} s_t P_t(s_t|s_{t-1},a_{t-1})\pi_{\theta^*}(a_t|s_t)$.

The last equality uses the definition of **discounted state visitation distribution** :

$$\mathcal{V}^{\pi}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_t^{\pi}(s). \tag{4}$$

$\mathbb{E}_{s\sim\mathcal{V}^{\pi_{\theta'}}}$: Average over states s, weighted by how often s is visited under $\pi_{\theta'}$.

$\mathbb{E}_{a\sim\pi_{\theta'}(\cdot|s)}$: Average over actions sampled from $\pi_{\theta'}$ in state s.

If the propose is $J(\theta')-J(\theta)\geqslant 0$, it is to make $\mathbb{E}_{s\sim\mathcal{V}^{\pi_{\theta'}}}\mathbb{E}_{a\sim\pi_{\theta'}(\cdot|s)}[A^{\pi_\theta}(s,a)]\geqslant 0$.

3.1 Is it solvable?

However, directly solving this equation is very difficult because $\pi_{\theta'}$is the policy we need to solve for, but we also need to use it to collect samples.

Do $s\sim\mathcal{V}^{\pi_{\theta'}}\rightarrow s\sim\mathcal{V}^{\pi_\theta}$, approximately get:

$$\begin{aligned} J(\theta')\approx L_\theta(\theta') &= J(\theta)+\frac{1}{1-\gamma}\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_{\theta'}(\cdot|s)}[A^{\pi_\theta}(s,a)] \\ &= J(\theta)+\frac{1}{1-\gamma}\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\frac{\pi_{\theta'}(a|s)}{\pi_\theta(a|s)}A^{\pi_\theta}(s,a)\right]. \text{Importance sampling} \end{aligned}$$

Then the optimization problem is:

$$\max_{\theta'} L_\theta(\theta')$$

$$s.t.\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}[D_{\mathrm{KL}}(\pi_\theta(\cdot|s),\pi_{\theta'}(\cdot|s))]\leqslant\delta.(\text{Trust region}).$$

Remark 2 Under the "Bregman Divergence" sense, the relationship between negative Shannon Entropy and KL-Divergence is just like norm and metric.

Bregman Divergence is defined as:

$$B_\varphi(x,y)=\varphi(x)-\varphi(y)-<\nabla\varphi(y),x-y>. \tag{5}$$

Question 1 Why not $D_{\mathrm{KL}}(\pi_{\theta'}(\cdot|s),\pi_\theta(\cdot|s))$?

4 Solve the model: TRPO

We use Talor$'$ s expansion to approximate the expected advantage under the new policy :

$$J(\theta')-J(\theta)\approx g^T(\theta'-\theta)$$

$$\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}[D_{\mathrm{KL}}(\pi_\theta(\cdot|s),\pi_{\theta'}(\cdot|s))]\approx\frac{1}{2}(\theta'-\theta)^TH((\theta'-\theta)),$$

where $g\triangleq\nabla_{\theta'}\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\frac{\pi_{\theta'}(a|s)}{\pi_\theta(a|s)}A^{\pi_\theta}(s,a)\right]$, $H\triangleq\mathbf{H}_{\theta'}\left[\mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}[D_{\mathrm{KL}}(\pi_\theta(\cdot|s),\pi_{\theta'}(\cdot|s))]\right]$.

Question 2 Check whether if H is a positive-defined matrix.

Then the optimization problem can be approximately expressed:

$$\theta_{k+1}=\operatorname*{argmax}_{\theta'}(g^T(\theta'-\theta)) \tag{6}$$

$$s.t.\frac{1}{2}(\theta'-\theta)^TH(\theta'-\theta)\leqslant\delta.$$

Remark 3 The maximum of hyperplane on the elliptic region can always expect to have solution.

By KKT-condition:

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g \tag{7}$$

4.1 Numerical improvement:Conjugate Gradient method

Solving $H^{-1}g$ is expensive . Set $x = H^{-1}g$:

$$\theta_{k+1} = \theta_k + \sqrt{\frac{2\delta}{x^T H x}} x \tag{8}$$

Then the problem reduces to solve $Hx = g$.

Solve $xs.t.Hx = g \Leftrightarrow \min_x \frac{1}{2}x^T Hx - gx$. Set $\varphi(x) \triangleq \frac{1}{2}x^T Hx - gx$, since $\nabla\varphi(x) = Hx - g$, we can solve :

$$\min_{\alpha} \varphi(x - \alpha \nabla \varphi(x)) \tag{9}$$

to get the best step based on the information now . But the flaw is too focus on the local information .

This is the so called Steepest gradient descent.

For CGset $r_0 = p_0 = -\nabla\varphi(x_0) = b - Ax_0, \alpha_k = \operatorname{argmin}_{\alpha} \varphi(x_k + \alpha p_k), x_{k+1} = x_k + \alpha_k p_k,$

$r_{k+1} = r_k - \alpha_k A(x_{k+1} - x_k)$, if not converge: $\begin{cases} \beta_{k+1} = \operatorname{argmin}_{\beta} \varphi(x_{k+1} + \alpha_{k+1}(r_{k+1} + \beta p_k)) \\ p_{k+1} = r_{k+1} + \beta_{k+1} p_k. \end{cases}$

it updates like:

Algorithm Conjugate Gradient Algorithm

Input: Symmetric positivedefinited matrix H, vector g, initial point x_0, tolerance ϵ;

Initialize $r_0 = g - Hx_0$, $p_0 = r_0$, $x_0 = 0$;

1. Repeat

   (a) For $k = 0$ to N do

      i. Compute step size:

$$\alpha_k = \frac{r_k^T r_k}{p_k^T H p_k};$$

      ii. Update solution:

$$x_{k+1} = x_k + \alpha_k p_k;$$

      iii. Update residual:

$$r_{k+1} = r_k - \alpha_k H p_k;$$

iv. If $r_{k+1}^T r_{k+1} < \epsilon$, exit loop;

v. Compute conjugate direction:

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k};$$

vi. Update search direction:

$$p_{k+1} = r_{k+1} + \beta_k p_k;$$

2. Until convergence;

Output: Solution vector x.

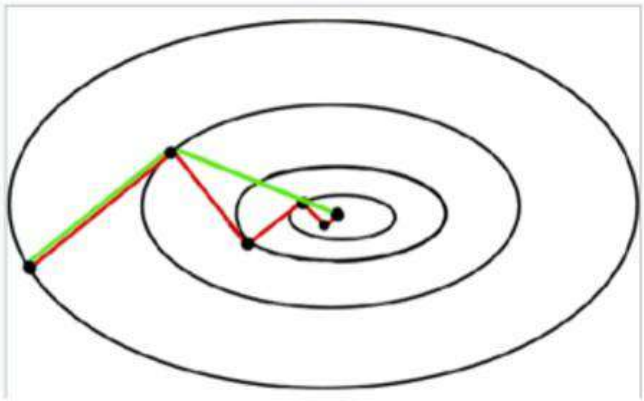

Figure 1: Steepest Descent Method(green) and CG Method(red)

4.2 Linear search

Since the solving process bases on linear approximation, here we need to make sure the newly update parameter is better. To make the step more precise, the TRPO updates the formula to get θ_{k+1} to be :

$$\theta_{k+1} = \theta_k + \alpha^i \sqrt{\frac{2\delta}{x^T H x}} x, \alpha^i \in (0,1), i \in \mathbb{N}. \tag{10}$$

i is the smallest $non-negative$ number $s.t. J(\theta_{k+1}) \geqslant J(\theta_k)$.

Algorithm: Policy Optimization with Value Network

1. **Initialize**:

   - Policy network parameters: θ
   - Value network parameters: ω

2. **For** episode $e = 1$ to E **do**:

(a) **Sample trajectory** using current policy π_θ:

$$\{s_1, a_1, r_1, s_2, a_2, r_2, \ldots\}$$

(b) **Estimate advantages** $A(s_t, a_t)$ for each state-action pair using the value network.

(c) **Compute gradient** g of the policy objective function.

(d) **Compute** $x = H^{-1}g$ using conjugate gradient method, where H is the Hessian matrix.

(e) **Perform line search** to find a step size α^i that improves the policy while satisfying the KL divergence constraint:

$$\theta_{k+1} = \theta_k + \alpha^i \sqrt{\frac{2\delta}{x^T H x}} x$$

where $i \in \{1, 2, \ldots, K\}$ is the smallest integer that satisfies the constraints.

(f) **Update value network parameters** using the same method as in Actor-Critic algorithms.

3. **End For**

5 Improvement:PPO

If do not use the linear approximation, we still have ways to solve the initial optimization problem:

$$\max_{\theta'} L_\theta(\theta') \tag{11}$$
$$s.t. \mathbb{E}_{s\sim\mathcal{V}^{\pi_\theta}}[D_{\mathrm{KL}}(\pi_\theta(\cdot|s), \pi_{\theta'}(\cdot|s))] \leqslant \delta. (\text{Trust region}).$$

5.1 PPO-Penalty:

$$\theta_{k+1} = \arg\max_\theta \mathbb{E}_{s\sim\mathcal{V}^{\pi_{\theta_k}}} \mathbb{E}_{a\sim\pi_{\theta_k}(\cdot|s)} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a) - \beta D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s), \pi_\theta(\cdot|s)) \right] \tag{12}$$

Set $d_k = D_{\mathrm{KL}}^{\mathcal{V}^{\pi_{\theta_k}}}(\pi_{\theta_k}, \pi_\theta)$, β has the updated rule :

1.If $d_k < \frac{\delta}{1.5} \Rightarrow \beta_{k+1} = \frac{\beta_k}{2}$.

2. If $d_k > 1.5\delta \Rightarrow \beta_{k+1} = 2\beta_k$.

3. Else $\beta_{k+1} = \beta_k$.

Remark 4 Purely empirically.

5.2 PPO-Clip:

$$\underset{\theta}{\operatorname{argmax}}\mathbb{E}_{s\sim\mathcal{V}^{\pi_{\theta_k}}}\mathbb{E}_{a\sim\pi_{\theta_k}(\cdot|s)}\left[\min\left\{\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A^{\pi_{\theta_k}}(s,a),\operatorname{chip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)};1-\varepsilon,1+\varepsilon\right)A^{\pi_{\theta_k}}(s,a)\right\}\right] \tag{13}$$

where $\operatorname{clip}(x;l,r)\triangleq\max\{\min\{x,r\},l\}$, *i.e.* restricts x in $[l,r]$.

Question 3 Can Clip method totally replaces the Penalty method?

One result for Question 3 is PPO-Clip gives out a rigorious restrict, while PPO-Penalty only gives out a quite wide region, so numerically Clip is sometimes better than Penalty but Penalty has more tolerance.

6 Experiment

Here is the result from the author Zhang Weinan:

- 7个连续控制的环境
- 3个random seed
- 每个算法跑100个episode，跑21遍，做平均值计算
- 最佳score归一化为1

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Figure 2: Experiment result from the video of Professor Zhang Weinan

7 Epilogue

With the numerical and theoretical foundations established earlier, I believe it is now much easier to understand what GRPO has achieved. However, this is not the end of the algorithm's journey—it is merely the beginning. As more talented individuals with strong mathematical backgrounds delve into the fields of Reinforcement Learning and Large Language Models (LLMs), the clearer our understanding of this world will become. Together, we can unlock new possibilities and push the boundaries of what these technologies can achieve.