Proximal Policy Optimization

Presented by :
Atmani Hanan

University mohammed vi polytechnic

July 23

Presentation plan

Introduction

- Update Gradient asciente:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta J(\theta^t)$$

- **Unstable update**
  - Step size is very important (If step size is very small, learning process is slow)
  - Next batch is generated from current bad policy $\rightarrow$ Collect bad samples.
  - Bad sample $\rightarrow$ worse policy
- **Data Inefficiency**
  - On policy method: for each new policy we need to generate a completely new trajectory
  - the data is throw out after just one gradient update

Efficient Data

- If we uses the advantage expression of gradient:

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t A^{\pi_\theta}(s, a) \nabla \log \pi_\theta(a \mid s) \right]$$

- Can we estimate an expectation of one distribution without taking samples from it ?
- Estimate one distribution by sampling from another distribution:

$$\begin{aligned} \mathbb{E}_{x \sim p}(f(x)) &= \int f(x) p(x)\, dx, \\ &= \int f(x) \frac{p(x)}{q(x)} q(x)\, dx, \\ &= \mathbb{E}_{x \sim q}\left(f(x) \frac{p(x)}{q(x)} \right) \approx \frac{1}{N} \sum_{i=1, x_i \sim q}^{N} \left(f(x^i) \frac{p(x^i)}{q(x^i)} \right) \end{aligned}$$

Efficient Data

- $$\begin{aligned}\nabla J(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta}\left[A^{\pi_\theta}(s,a) \nabla \log \pi_\theta(a \mid s)\right] \\ &= \mathbb{E}_{\tau \sim \pi_{\theta \text{old}}}\left[\frac{\pi_\theta(s_t, a_t)}{\pi_{\theta \text{old}}(s_t, a_t)} A^{\pi_\theta}(s,a) \nabla \log \pi_\theta(a \mid s)\right]\end{aligned}$$
- Then the surrogate objective function:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta \text{old}}}\left[\frac{\pi_\theta(s_t, a_t)}{\pi_{\theta \text{old}}(s_t, a_t)} A^{\pi_\theta}(s,a)\right]$$

- Two expectations are same, but we are using sampling method to estimate them $\rightarrow$ variance is also important

Efficient Data

- We have: $\mathrm{Var}_{x\sim p}(f(x)) = \mathbb{E}_{x\sim p}(f(x)^2) - (\mathbb{E}_{x\sim p}(f(x)))^2$ and $\mathbb{E}_{x\sim p}(f(x)) = \mathbb{E}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right)$

Then

$$\mathrm{Var}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right) = \mathbb{E}_{x\sim q}\left(\left(f(x)\frac{p(x)}{q(x)}\right)^2\right) - \left(\mathbb{E}_{x\sim q}\left(f(x)\frac{p(x)}{q(x)}\right)\right)^2$$
$$= \mathbb{E}_{x\sim p}\left(f(x)^2\frac{p(x)}{q(x)}\right) - (\mathbb{E}_{x\sim p}(f(x)))^2$$

- We may need to sample more data if $\frac{p(x)}{q(x)}$ is far away from 1

UM6P College of Computing

Stable Update

- Make confident update:
  - Adaptive learning rate
  - limit the policy update range
- Can we measure the distance between two distributions ?
- **KL Divergence:** Measure the distance of two distributions:

$$D_{KL}(p||q) = \sum_{x} p(x) \log \left(\frac{p(x)}{q(x)} \right)$$

- KL divergence of two policies:

$$D_{KL}(\pi_1||\pi_2)[s] = \sum_{a \in \mathcal{A}} \pi_1(a|s) \log \left(\frac{\pi_1(a|s)}{\pi_2(a|s)} \right)$$

Trust Region Policy optimization (TRPO)

- The TRPO method involves solving the problem by linearizing the objective function and transforming the constraints into quadratic form.

$$\mathrm{max}_\theta \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\mathrm{old}}(a_t|s_t)} \hat{A}_t \right)$$

Subject to

$$\hat{\mathbb{E}}_t \left(KL(\pi_{\theta_{\mathrm{old}}}, \pi_\theta(.|s_t)) \right) \leq \delta$$

- TRPO uses conjugate gradient descent to solve the optimization problem. The Hessian matrix is computationally and memory expensive

PPO with Adaptive KL Penalty

- The Constraint helps in the training process. However, maybe the constraint is not a strict constraint. Does it matter if we only break the constraint just a few times ?
- What if we treat it as a " soft" constraint? add proximal value to objective function?
- PPO with Adaptive KL Penalty:

$$L^{KLpen}(\theta) = \hat{\mathbb{E}}_t\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}\hat{A}_t - \beta D_{KL}(\pi_{\theta_{\text{old}}}(.|s_t), \pi_\theta(.|s_t))\right)$$

  - Hard to pick β value $\rightarrow$ Use adaptive
- Compute $d = \hat{\mathbb{E}}_t\left(D_{KL}(\pi_{\theta_{\text{old}}}(.|s_t), \pi_\theta(.|s_t))\right)$
  - If $d < d_{targ}/1.5$, $\beta \leftarrow \frac{\beta}{2}$ (more data)
  - If $d > d_{targ} \times 1.5$ (more penalty)
- Still need to setup a KL divergence target value...

Algorithm PPO with adaptive KL penalty

- Input: Initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
- for $k = 0, 1, 2, ...$ do
  - Collect set of partial trajectories on policy $\pi_k = \pi_{\theta_k}$
  - Estimate advantage $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm
  - Compute policy update:

$$\theta_{k+1} = \text{argmax}_\theta \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(a_t|s_t)}{\pi_\text{k}(a_t|s_t)} \hat{A}_t^{\pi_k} - \beta_k D_{KL}(\pi_{\theta_k}(.|s_t), \pi_\theta(.|s_t)) \right)$$

  - if $\hat{\mathbb{E}}_t \left(D_{KL}(\pi_{\theta_k}(.|s_t), \pi_{\theta_{k+1}}(.|s_t))) \right) \geq 1.5\delta$, Then:
    - $\beta_{k+1} = 2\beta_k$
  - Else if $\hat{\mathbb{E}}_t \left(D_{KL}(\pi_{\theta_k(.|s_t), \pi_{\theta k+1}}(.|s_t)) \right) \leq \frac{1.5}{\delta}$, then
    - $\beta_{k+1} = \beta_k / 2$
- end for

PPO with Clipped objective

-$max_\theta \hat{\mathbb{E}}_t \left(\frac{\pi_\theta(s_t/a_t)}{\pi_{\theta_{old}}(s_t/a_t)} \hat{A}_t \right)$.

-Denote the probability ratio: $r_t(\theta) = \frac{\pi_\theta(s_t/a_t)}{\pi_{\theta_{old}}(s_t/a_t)}$

Invariance happens when r changes too quickly $\rightarrow$ limit r within a range?

- Input :initial policy parameters θ_0, clipping parameter ϵ
- For k=0,1,2,... do
  - Collect of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi_{\theta_k}$
  - Estmate advantage $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm.
  - Compute policy update:

$$\theta_{k+1} = \text{argmax}_\theta L_{\theta_k}^{CLIP}(\theta)$$

  - and for

- Where

$$L_{\theta_k}^{CLIP}(\theta) = \hat{\mathbb{E}}_{\tau \sim \pi_k} \left(\sum_{t=0}^{T} \left[min \left(r_t(\theta) \hat{A}_t^{\pi_k}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t^{\pi_k} \right) \right] \right.$$

and

$$clip(x; 1-\epsilon, 1+\epsilon) = \begin{cases} 1-\epsilon & x \leq 1-\epsilon \\ 1+\epsilon & x \geq 1+\epsilon . \pi_\theta, \\ x & \text{else} \end{cases}$$