Page 9

In math form, old A3C objective could be written as $J_\theta = \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)A_t]$. The new objective proposed by the PPO is $J_\theta = \mathbb{E}_t[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}A_t]$. But if we just start to blindly maximize this value, it will lead to very large update to the policy weights. To limit the update, the clipped objective is used. If we write the ratio between the new and the old policy as $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, the clipped objective could be written as this

$$J_\theta^{clip} = \mathbb{E}_t[\min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)]$$

This objective limits the ratio between the old and the new policy to be in the interval $[1-\epsilon, 1+\epsilon]$, so, by varying ϵ we can limit the size of the update.

Another difference from the A3C method, is the way we estimate the advantage. In the A3C paper, the advantage obtained from the finite-horizon estimation of T steps in the form:

$$A_t = -V(s_t) + r_t + \gamma r_{t+1} + \ldots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T)$$

In the PPO paper, the authors used more general estimation in the form of

$$A_t = \sigma_t + (\gamma\lambda)\sigma_{t+1} + (\gamma\lambda)^2\sigma_{t+2} + \ldots + (\gamma\lambda)^{T-t+1}\sigma_{T-1}$$

where $\sigma_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. The original A3C estimation, is a special case of the proposed method with $\lambda = 1$.

Page 11

the advantage estimation (tracked in last_gae variable) is calculated as the sum of deltas with discount factor $\gamma\lambda$.

Page 12

In the next step, we calculate the logarithm of probability of the actions taken. This value will be used as $\pi_{\theta_{old}}$ in the surrogate objective of PPO. Additionally, we normalize the advantages mean and variance to improve the training stability.

Page 13

In the actor training, we minimize the negated clipped objective,

$$\mathbb{E}_t[\min(r_t(\theta)A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)]$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. The lines below is straightforward implementation of this formula.

Page 15, Trust Region Policy Optimisation

As the first step, TRPO method defines the discounted visitation frequencies of the state: $\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots$. In this equation, $P(s_i = s)$ equals to the sampled probability of state s to be met at position i of the sampled trajectories. Then, TRPO defines the optimisation objective as $L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a)$, where $\eta(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t)]$ is the expected discounted reward of the policy and $\tilde{\pi} = \arg\max_a A_\pi(s,a)$ defines the deterministic policy. To address the issue with large policy updates, TRPO

defines the additional constraint on the policy update, expressed as a maximum Kullback-Leibler divergence between the old and the new policies, which could be written as $\bar{D}_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$.

Page 21, Soft Actor-Critic Method

The central idea in SAC method is entropy regularization, which adds at each timestamp a bonus reward that is proportional to the entropy of the policy at this timestamp. In mathematical notation, the policy were looking for is the following:

$$\pi^* = \arg\max_a \mathbb{E}_{\tau\sim\pi} \sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t, s_{t+1}) + \alpha H(\pi(\cdot|s_t)))$$

Where $H(P) = \mathbb{E}_{x\sim P}[-\log P(x)]$ is an entropy of distribution P.

So, in total, we train four networks: policy $\pi(s)$, value $V(s, a)$ and two Q-networks $Q_{1,2}(s, a)$. For the value network $V(s, a)$, the target network is used. So, in total, Soft Actor-Critic training looks like following:

- Q-networks are trained using the MSE objective by doing Bellman approximation using target value network: $y_q(r, s') = r + \gamma V_{tgt}(s')$ (for non-terminating steps).
- The V-network is trained using the MSE objective with the following target: $y_v(s) = \min_{i=1,2} Q_i(s, \tilde{a}) - \alpha \log \pi_\theta(\tilde{a}|s)$, where $\tilde{a}$ is sampled from policy $\pi_\theta(\cdot|s)$.
- The policy network π_θ is being trained in the DDPG style by maximizing the following objective: $Q_1(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s)|s)$ where $\tilde{a}_\theta$ is a sample from $\pi_\theta(\cdot|s)$.