PRM with off-policy RL method

Saito Karuha

November 2024

1 Pseudo Code

Algorithm 1 DDPG For PRM Traning

```
**Initialize:** CriticNet: $Q_\omega(s,a)$ , ActionNet: $\mu_\theta(s)$
**Initialize:** $Q_{\omega^-}(s,a) \leftarrow Q_\omega(s,a)$ , $\mu_{\theta^-}(s) \leftarrow \mu_\theta(s)$
**Buffer Initialize:** $B \leftarrow \emptyset$
**for** $e = 0$ **to** $E$ **do**
  Initialize initial state $s_1$ (Randomly pick a question from UCB-Math)
  **for** $t = 1$ **to** $T$ **and** $done == True$ **do**
    Choose an action $a_t \sim \mu_\theta(s_t)$
    $r_t \leftarrow Env(s_t, a_t)$ ; $s_{t+1} \leftarrow [s_t, a_t]$
    $B \leftarrow (s_t, a_t, r_t, s_{t+1})$
    **if** Buffer is big enough **then**
      Randomly pick $N$ touples $\{(s_i, a_i, r_i, s_{i+1})\}_{i=1,...,N}$
      Sample $K$ actions: $a^m_{i+1} \sim \mu_{\theta^-}(s_{i+1})$ , $(m = 1, ..., k)$
      Calculate for every tuples:

        $$y_i = r_i + \gamma * \max_{m \in \{1,...,k\}} Q_{\omega^-}(s_{i+1}, a^m_{i+1}) \qquad (1)$$

        $$A_i = r_i + \gamma * Random_j Q_{\omega^-}(s_{i+1}, a^j_{i+1}) - Q_\omega(s_i, a_i) \qquad (2)$$

      Compute loss for CriticNet($L$) and ActorNet($J$) respectively:
      $L = \frac{1}{N} \sum_{i=1}^N y_i - Q_\omega(s_i, a_i)$
      $J = \frac{1}{N} \sum_{i=1}^N min(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta^-}(a_i|s_i)} A_i, clip(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta^-}(a_i|s_i)}, 1-\epsilon, 1+\epsilon) A_i)$
      **Update for Critic and Actor Network**
      **Soft update target Network**
      $\omega^- \leftarrow \tau\omega + (1-\tau)\omega$ , $\theta^- \leftarrow \tau\theta + (1-\tau)\theta$
    **end if**
  **end for**
**end for**
```

2 Problems