Lecture13: Baseline and Actor-Critic

> Notes taken by squarezhong
> Repo address: squarezhong/SDM5008-Lecture-Notes

Reinforce

According to policy gradient

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)G_t\right]$$

Pseudo code:

Algorithm 1 REINFORCE

```
Initialize policy network π_θ(a|s)
for each episode do
    Generate an episode s_0, a_0, r_0, ..., s_T, a_T, r_T, following π_θ(a|s)
    for step t = 0, 1, ..., T do
        G ← Σ_{t'=t}^{T} γ^{t'-t} r_{t'}
        θ ← θ + αG∇ ln π_θ(a_t|s_t)
    end for
end for
```

For detailed python code, you can find in many RL tutorials.

Reinforce with Baseline

A result of the EGLP lemma:

> For any function b that depends solely on the state, we have
>
> $$\mathbf{E}_{a_t\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)b(s_i)\right] = 0$$

Then we have **policy gradient with baseline**

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)\left(G_t - b(s_t)\right)\right]$$

- Any function b is called a baseline.
- In general the baseline doesn't change the expected value, but has a large effect on its variance.

- The most common choice of baseline is the **value function** $V(s_t)$
- In practice, $V(s_t)$ is usually approximated by a neural network $V_\phi(s_t)$, which is updated concurrently with the policy.

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(G_t - V(s_t) \right) \right]$$

Pseudo code:

Algorithm 1 REINFORCE with Baseline

Initialize policy network $\pi_\theta(a|s)$ and state value network $V_\phi(s)$
for each episode **do**
  Generate an episode $s_0, a_0, r_0, ..., s_T, a_T, r_T$, following $\pi_\theta(a|s)$
  for step $t = 0, 1, ..., T$ **do**
    $G \leftarrow \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
    $\delta \leftarrow G - V_\phi(s_t)$
    $\phi \leftarrow \phi + \alpha^\phi \delta \nabla V_\phi(s_t)$
    $\theta \leftarrow \theta + \alpha^\theta \delta \nabla \ln \pi_\theta(a_t|s_t)$
  end for
end for

Attention:

- parameters in value network should not attend the backward process. `value.item()` may be used.

Actor-Critic

More general policy gradient form:

$$\nabla_\theta J(\pi_\theta) = \mathbf{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) f_t \right]$$

f_t can take take on various forms

$$1. \quad \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'}$$

$$2. \quad \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'} - b(s_t)$$

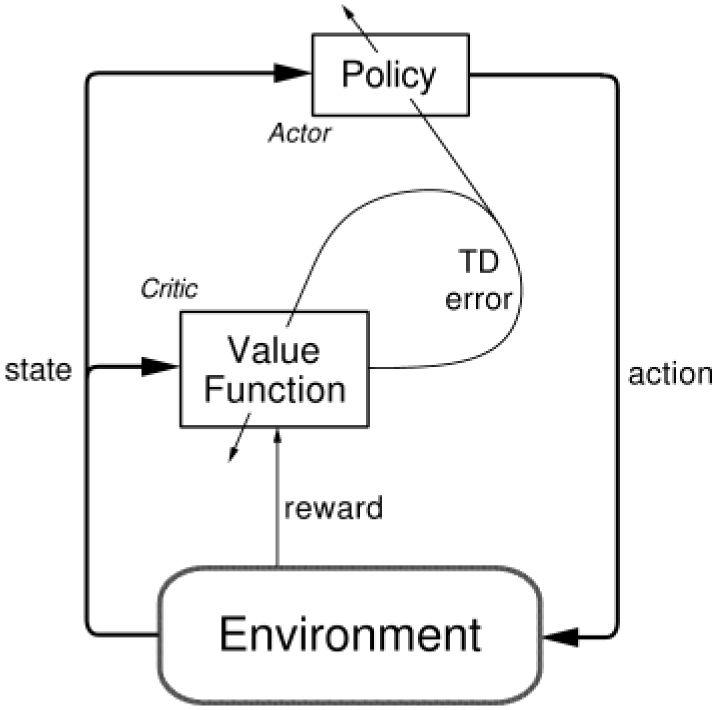
$$3. \quad Q(s_t, a_t)$$

$$4. \quad A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

$$5. \quad R_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)$$

The latter three f_t directly evaluate the action, which can be used in actor-critic.

The following graph describes the basic process of actor-critic



TD Actor-Critic

Here we only discusses TD Actor-Critic methods with one-step return:

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\left(G_t - V^\pi(s_t)\right)\right] \\
&= \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\left(R_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)\right)\right] \\
&= \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\delta\right]
\end{aligned}
$$

Pseudo code:

Algorithm 1 TD Actor-Crtic

Initialize policy network $\pi_\theta(a|s)$ and state value network $V_\phi(s)$
for each step t in episode **do**
  Generate action a_t, following $\pi_\theta(a|s)$
  $\delta \leftarrow r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$
  $\phi \leftarrow \phi + \alpha^\phi \delta \nabla V_\phi(S_t)$
  $\theta \leftarrow \theta + \alpha^\theta \delta \nabla \ln \pi_\theta(A_t|S_t)$
end for

- actor: policy network
- critic: value network

Detach

Value network in the calculation of δ and **actor loss** is just a numerical value. It does not attend the backward, so we need to use `.detach()` in the code.

Importance Sampling

For a group of data, we may need to iterate many epochs to make the loss converge. However, after updating the policy, we can not use data sampled by old policy $\pi_{\theta_{\text{old}}}$ to update the parameters of new policy π_θ. So we use **importance sampling** here.

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta \log\pi_\theta(a_t|s_t)\,\delta\right] \\
&= \mathbf{E}_{\tau\sim\pi_{\theta_{\text{old}}}}\left[\sum_{t=0}^{T}\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\nabla_\theta \log\pi_\theta(a_t|s_t)\,\delta\right]
\end{aligned}
$$