Actor Critic Methods

Devesh Nath

February 10, 2025

1 Actor Critic Algorithm

The Actor-Critic algorithm is a type of reinforcement learning algorithm that combines both value-based and policy-based methods. It consists of two main components: the actor and the critic.

1.1 Actor

The actor is responsible for selecting actions based on the current policy $\pi_\theta(a|s)$. It updates the policy parameters θ in the direction suggested by the critic to improve the policy.

1.2 Critic

The critic evaluates the action taken by the actor by computing the value function $V^\pi(s)$ or the action-value function $Q^\pi(s,a)$. It provides feedback to the actor on how good the action was, which helps in updating the policy.

1.3 Advantage Function

The advantage function $A^\pi(s,a)$ is used to determine how much better an action is compared to the average action taken from state s. It is defined as:

$$A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$$

where $Q^\pi(s,a)$ is the action-value function and $V^\pi(s)$ is the value function.

1.4 Mathematical Formulation

The policy gradient theorem states that the gradient of the expected return $J(\theta)$ with respect to the policy parameters θ is given by:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)\right]$$

The critic estimates the action-value function $Q^\pi(s,a)$ using the Bellman equation:

$$Q^\pi(s,a) = r + \gamma \mathbb{E}_{s'}\left[V^\pi(s')\right]$$

where r is the reward, γ is the discount factor, and s' is the next state.
The actor updates the policy parameters θ using the gradient ascent:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

where α is the learning rate.
The critic updates its parameters ϕ by minimizing the mean squared error between the estimated value and the target value:

$$\phi \leftarrow \phi - \beta \nabla_\phi \left(Q^\pi(s, a) - (r + \gamma V^\pi(s'))\right)^2$$

where β is the learning rate for the critic.

2 Off-Policy Actor Critic

Off-policy Actor-Critic methods allow the use of data generated from a different policy (behavior policy) than the one currently being optimized (target policy). This can improve sample efficiency by reusing past experiences.

2.1 Importance Sampling

To correct for the discrepancy between the behavior policy $\mu(a|s)$ and the target policy $\pi_\theta(a|s)$, importance sampling is used. The importance sampling ratio is defined as:

$$\rho_t = \frac{\pi_\theta(a_t|s_t)}{\mu(a_t|s_t)}$$

2.2 Off-Policy Gradient

The gradient of the expected return $J(\theta)$ with respect to the policy parameters θ in the off-policy setting is given by:

$$\nabla_\theta J(\theta) = \mathbb{E}_\mu \left[\rho_t \nabla_\theta \log \pi_\theta(a_t|s_t) A^\pi(s_t, a_t)\right]$$

2.3 Critic Update

The critic can be updated using the same Bellman equation as in the on-policy setting, but with data generated from the behavior policy:

$$Q^\pi(s, a) = r + \gamma \mathbb{E}_{s'} \left[V^\pi(s')\right]$$

2.4 Actor Update

The actor updates the policy parameters θ using the gradient ascent with the importance sampling correction:

$$\theta \leftarrow \theta + \alpha \rho_t \nabla_\theta \log \pi_\theta(a_t|s_t) A^\pi(s_t, a_t)$$

Off-policy methods can be more complex to implement due to the need for importance sampling and the potential for high variance in the gradient estimates. However, they offer the advantage of being able to leverage past experiences more effectively.