Comprehensive Notes on Policy Gradient Methods

Claude AI

1 Introduction to Policy-Based Reinforcement Learning

Policy-based reinforcement learning (RL) is an approach where we directly optimize the policy $\pi(a|s)$ that maps states to actions, instead of learning a value function and deriving a policy from it. This method has several advantages:

- It can learn stochastic policies
- It can handle continuous action spaces naturally
- It can solve problems with perceptual aliasing

2 Policy Objective Functions

In policy-based RL, we define an objective function $J(\theta)$ that we aim to maximize. Common choices include:

1. Start value: $J_1(\theta) = V^\pi(s_1)$
2. Average value: $J_{avV}(\theta) = \sum_s d^\pi(s) V^\pi(s)$
3. Average reward per time-step: $J_{avR}(\theta) = \sum_s d^\pi(s) \sum_a \pi(a|s) R(s,a)$

where $d^\pi(s)$ is the stationary distribution of Markov chain for π.

3 Policy Gradient Theorem

The policy gradient theorem states that for any differentiable policy $\pi(\theta)$ and for any policy objective function $J(\theta)$, the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s) Q^\pi(s,a)] \tag{1}$$

Proof:

We'll prove this for the start state objective $J_1(\theta) = V^\pi(s_1)$. Let's start with the definition of the value function:

$$V^\pi(s) = \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...|s_t = s] \tag{2}$$

Taking the gradient with respect to θ:

$$\begin{aligned}
\nabla_\theta V^\pi(s) &= \nabla_\theta \mathbb{E}_\pi[r_t + \gamma V^\pi(s_{t+1})|s_t = s] && (3)\\
&= \sum_a \nabla_\theta \pi(a|s) Q^\pi(s,a) + \sum_a \pi(a|s) \sum_{s'} P(s'|s,a) \nabla_\theta V^\pi(s') && (4)
\end{aligned}$$

Let $x(s) = \nabla_\theta V^\pi(s)$. Then we can write:

$$x(s) = \sum_a \nabla_\theta \pi(a|s) Q^\pi(s,a) + \gamma \sum_a \pi(a|s) \sum_{s'} P(s'|s,a) x(s') \tag{5}$$

This is a linear system of equations. We can solve it as:

$$x = b + \gamma P x \tag{6}$$

where $b(s) = \sum_a \nabla_\theta \pi(a|s) Q^\pi(s,a)$ and P is the transition matrix under π. The solution to this system is:

$$x = (I - \gamma P)^{-1} b \tag{7}$$

Now, let $d^\pi(s)$ be the stationary distribution of P. Multiplying both sides by $d^\pi(s)^T$:

$$d^\pi(s)^T x = d^\pi(s)^T (I - \gamma P)^{-1} b \tag{8}$$

The left side is what we want: $\nabla_\theta J(\theta)$. For the right side:

$$\begin{aligned}
d^\pi(s)^T (I - \gamma P)^{-1} &= d^\pi(s)^T (I + \gamma P + \gamma^2 P^2 + ...) && (9)\\
&= d^\pi(s)^T + \gamma d^\pi(s)^T + \gamma^2 d^\pi(s)^T + ... && (10)\\
&= \frac{1}{1-\gamma} d^\pi(s)^T && (11)
\end{aligned}$$

Therefore:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \frac{1}{1-\gamma} d^\pi(s)^T b && (12)\\
&= \frac{1}{1-\gamma} \sum_s d^\pi(s) \sum_a \nabla_\theta \pi(a|s) Q^\pi(s,a) && (13)\\
&= \frac{1}{1-\gamma} \sum_s d^\pi(s) \sum_a \pi(a|s) \frac{\nabla_\theta \pi(a|s)}{\pi(a|s)} Q^\pi(s,a) && (14)\\
&= \frac{1}{1-\gamma} \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s) Q^\pi(s,a)] && (15)
\end{aligned}$$

This completes the proof of the policy gradient theorem.

4 REINFORCE Algorithm

The REINFORCE algorithm is a Monte Carlo policy gradient method:

1. Generate an episode $S_0, A_0, R_1, ..., S_{T-1}, A_{T-1}, R_T$ following π
2. For each step $t = 0, ..., T-1$:
    - $G_t \leftarrow$ return from step t
    - $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(A_t|S_t) G_t$

5 Reducing Variance with a Baseline

We can reduce the variance of policy gradient estimates by subtracting a baseline:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)(Q^\pi(s,a) - b(s))] \tag{16}$$

A good choice for the baseline is the state value function $V^\pi(s)$.

6 Actor-Critic Methods

Actor-Critic methods combine policy-based and value-based learning. The actor (policy) is updated according to the critic's (value function) evaluation.

A simple actor-critic algorithm:

1. Initialize s, θ, w
2. Sample $a \sim \pi(a|s)$
3. Take action a, observe r, s'
4. $\delta = r + \gamma V_w(s') - V_w(s)$
5. $w \leftarrow w + \beta \delta \nabla_w V_w(s)$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi(a|s) \delta$
7. $s \leftarrow s'$
8. Go to 2

7 Advantage Actor-Critic (A2C)

A2C uses the advantage function $A(s,a) = Q(s,a) - V(s)$ instead of just $Q(s,a)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s) A^\pi(s,a)] \tag{17}$$

We can estimate the advantage function using TD error:

$$A(s_t, a_t) \approx r_t + \gamma V(s_{t+1}) - V(s_t) \tag{18}$$

8 Conclusion

Policy gradient methods offer a powerful approach to reinforcement learning, especially in domains with continuous action spaces or partial observability. While they can suffer from high variance, techniques like baselines and actor-critic methods help mitigate this issue. Advanced algorithms like A2C, which we've discussed, form the foundation for many state-of-the-art RL algorithms.