Policy Gradients and Actor Critic methods - A summary

April 3, 2024

1 Introduction

This short article will shed light on one of the most fundamental algorithms of Reinforcement Learning (RL), the Policy Gradient and its derived methods, the Actor-Critic. I will present them intuitively and analytically, how I understood them best!

2 Policy Gradient Foundation

Foundational RL algorithms focus on learning the state value function

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}R_{t+k+1}|S_t = s\right]$$

and the action value function

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}R_{t+k+1}|S_t = s, A_t = a\right]$$

measuring the discounted reward of following policy π in state s for $V^{\pi}(s)$ or the discounted reward of taking action a in state s and thereafter follow the policy π for $Q^{\pi}(s,a)$ in a tabular approach. We either use Monte-Carlo methods with Dynamic Programming or Temporal Difference algorithms to experience all state and action combinations and then update the current estimated value functions with the newly seen rewards. We result in a table where we can map each state or state-action pair to a specific value which then lets us choose the optimal policy by selecting a combination that yields maximum value.

However, as the environment gets intricate the possible states start to explode. For instance, take the game Breakout in Atari where the state s consists of the current frame of the game, i.e. an image of dimension (210,160,3). It should be straightforward that, let alone, all possible states of this image are almost infinite. Taking the actions additionally into account, the state-action combinations even further increase.

The basic reasoning behind the Policy Gradient is the idea that instead of using a tabular approach we parametrize our policy using θ and adjust these parameters such that given a state s the policy suggests a distribution over actions which maximizes some performance measure $\mathbf{J}(\theta)$. Sutton and Barto (2020) opt to use the expected return of future rewards, which we denote with $v_{\pi(\theta)}(s_0)$, i.e. the true value function, as this measure. The policy Gradient Theorem shows that we can describe the gradient of $\mathbf{J}(\theta)$ as:

$$\nabla \mathbf{J}(\theta) = \nabla v_{\pi(\theta)} \propto \sum(s)\mu(s)\sum_a \pi(a|s)q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_a \pi(a|s)q^\pi(s,a)\right]$$

Hence, the Policy Gradient is proportional to the expected value of the action value function weighted by the probability of taking the action under the current policy π (Refer to pages 325ff of Sutton and Bartos book in chapter 13.2 for more details).
The cool part is, that we only need basic calculus rules for the proof. I will not go in detail, because the proof is presented very detailed by Sutton and Barto on page 325 in chapter 13.2.
If we now only consider the actual action a_t taken in this state (instead of summing over all possible actions) as suggested by the policy π and remember that $q^\pi(s,a) = G_t$, which is the discounted sum of rewards and simplify further we get:

$$\nabla \mathbf{J}(\theta) \propto \mathbb{E}_\pi\left[\frac{\nabla\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)}G_t\right] = \mathbb{E}_\pi\left[\nabla \ln \pi(a_t|s_t,\theta)G_t\right]$$

Remember that we can plug in the ln since $\frac{\nabla x}{x} = \nabla \ln(x)$.

Using Monte Carlo sampling for the episodic case and applying stochastic gradient ascent we get the basic REINFORCE algorithm as described by Sutton and Barto.
In line 8 of algorithm 1 we can see the update rule. The parameters θ are adjusted in a way, such that the policy pi increases the probability of actions a_t that yield high discounted returns and reduces the probability of actions that do yield low discounted returns.

This section presented the core idea of the Policy Gradient why it was introduced and how it works. The research area of RL used this foundation to build new, optimized algorithms to further increase the performance of Policy Gradient methods. We will explore these in the next chapters.

In summary, the inherent need to invent the Policy Gradient method was shown. Further, the derivation of the Policy Gradient was examined resulting in our fi-

Algorithm 1 REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π^*

```
1: Input: a differentiable policy parameterization $\pi(a|s,\theta)$
2: Algorithm parameter: step size $\alpha > 0$
3: Initialize policy parameter $\theta \in \mathbb{R}^{d_0}$ (e.g., to 0)
4: loop (for each episode)
5:     Generate an episode $s_0, a_0, r_1, ..., s_{T-1}, a_{T-1}, r_T$, following a$\pi(\cdot|\cdot,\theta)$
6:     for each step of the episode $t = 0, 1, ..., T-1$ do
7:         Calculate return $G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_k$
8:         Update $\theta \leftarrow \theta + \alpha\gamma^t G_t \nabla_\theta \ln \pi(a_t|s_t,\theta)$
9:     end for
10: end loop
```

nal algorithm which successfully incorporated the Policy Gradient in the Monte-Carlo sampling method to construct a parametrized policy $\pi(\theta)$ that maximizes the discounted future rewards.

3 Baseline and Advantage methods