Deep Deterministic Policy Gradient (DDPG)

Hadelin de Ponteves

February 10, 2024

Abstract

Deep Deterministic Policy Gradient (DDPG) is a seminal algorithm in the field of reinforcement learning that combines the benefits of Deep Learning with the classical Deterministic Policy Gradient approach. Bridging the gap between policy-based and value-based methods, DDPG is capable of handling high-dimensional, continuous action spaces. This document aims to provide a comprehensive overview of DDPG, elucidating its principles, mathematical framework, and practical applications, making it accessible for learners and practitioners alike.

Contents

1 Introduction

DDPG is an actor-critic, model-free algorithm that learns a deterministic policy in continuous action spaces. It adapts the insights from Deep Q-Learning (DQN) to the policy gradient methods, enabling efficient learning in environments with a high degree of complexity.

2 Background

2.1 Policy Gradient Methods

Policy gradient methods optimize the policy directly by estimating the gradient of the expected return with respect to policy parameters, facilitating learning in continuous domains.

2.2 Actor-Critic Framework

Actor-critic methods utilize two models: the actor, which suggests actions given the current state, and the critic, which evaluates the proposed action by estimating the Q-value function.

3 The DDPG Algorithm

DDPG combines ideas from DQN and Deterministic Policy Gradient (DPG) to learn policies in high-dimensional, continuous action spaces efficiently.

3.1 Key Features

- **Deterministic Policy:** DDPG learns a deterministic policy that maps states to specific actions, reducing the variance in policy gradient estimation.
- **Off-Policy Learning:** It uses experience replay and target networks to stabilize training, similar to DQN.
- **Continuous Action Spaces:** DDPG can operate in environments with continuous action spaces, making it suitable for a wide range of applications.

3.2 Mathematical Formulation

The objective of DDPG is to maximize the expected return from the start distribution $J = \mathbb{E}_{s_0,a_0,\ldots}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$, where γ is the discount factor.

3.2.1 Critic (Value) Network

The critic network estimates the Q-value function. The loss function for updating the critic is:

$$L(\theta^Q) = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}\sim\mathcal{D}}[(Q(s_t, a_t|\theta^Q) - y_t)^2], \tag{1}$$

where $y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1})|\theta^Q)$, and $\mathcal{D}$ is a replay buffer.

The critic network is updated by minimizing the mean squared error loss between the predicted Q-values and the target Q-values.

3.2.2 Actor (Policy) Network

The actor network deterministically maps states to actions. The objective function for the actor is given by:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}_{s_t \sim \rho^\beta}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_t}], \tag{2}$$

where θ^μ are the parameters of the actor network, and ρ^β is the state distribution under behavior policy β.

The actor network is updated by moving the actor's parameters in the direction that maximizes the critic's Q-value estimation, effectively improving the policy.

3.2.3 Target Networks

To further stabilize training, DDPG employs target networks for both the actor and the critic.

The target networks Q' and μ' are updated using a soft update strategy to slowly track the learned networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \quad \theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}, \tag{3}$$

where τ is a small constant (e.g., 0.001), blending the parameters of the learned networks into the target networks to ensure training stability.

4 Practical Considerations

4.1 Experience Replay

DDPG uses a replay buffer to store transitions, which are sampled randomly to break the correlation between consecutive samples, enhancing training stability.

4.2 Exploration vs. Exploitation

In continuous spaces, exploration is achieved by adding noise to the policy's actions, commonly using Ornstein-Uhlenbeck process for temporally correlated noise.

4.3 Hyperparameter Tuning

Key hyperparameters include the learning rates for the actor and critic networks, the discount factor γ, and the soft update coefficient τ for the target networks.

5 Applications

DDPG has been successfully applied in domains such as robotics for tasks that require precise control in continuous action spaces.

6 Conclusion

DDPG represents a significant advancement in reinforcement learning, enabling efficient policy learning in continuous action spaces. Its actor-critic architecture, combined with techniques like experience replay and target networks, offers a powerful framework for solving complex problems.