Reinforcement Learning Algorithm

The goal of reinforcement learning algorithms is to learn how to take actions that maximize cumulative rewards through interaction with the environment.

In reinforcement learning, the agent learns by observing the state of the environment, choosing actions, and receiving rewards. It does not require explicit labels but rather optimizes strategies by trying different actions to maximize long-term cumulative rewards. Reinforcement learning is widely applied in areas such as automatic control, gaming, robotics, autonomous driving, where agents gradually enhance their performance by interacting with the environment.

1，Q-Learning

Q-Learning is a fundamental algorithm in reinforcement learning used to address the problem of a single agent in a Markov Decision Process (MDP). The primary goal of Q-Learning is to learn a value function (known as the Q-value function), which represents the expected return for taking each action in every state. By learning this value function, an agent can select the optimal actions in the environment to maximize cumulative rewards.

Basic Principles

1. State and Action: Q-Learning is based on a Markov Decision Process, which involves a set of states and a set of available actions. The agent selects an action based on the current state, then observes the reward and the next state, continuously making decisions in this manner.
2. Q-Value Function: At the core of Q-Learning is the learning of a Q-value function (also known as the action-value function), typically denoted as Q(s, a). This function represents the expected return for taking action **a** in state **s**
3. Bellman Equation: The Q-value function satisfies the Bellman equation, which expresses the relationship between the Q-values of a state and the next state.
4. Policy: The agent selects actions based on the Q-value function. Typically, an ε-greedy policy is used, where, with a certain probability, it chooses the action with the highest Q-value to exploit the environment, while occasionally exploring other actions.

Q-Learning is a powerful reinforcement learning algorithm that can be applied to solve various single-agent problems, including robot control, game strategy optimization, and automated decision-making. It serves as a fundamental cornerstone in the field of reinforcement learning and has been extended and improved to accommodate more complex problems and multi-agent scenarios.

2，Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) is an approach that combines deep learning and reinforcement learning to address complex decision-making problems. The fundamental principle of DRL involves using deep neural networks to estimate and optimize the agent's strategy in an environment to maximize cumulative rewards through action selection.

Basic Principles

1. Agent and Environment: Problems in DRL are typically modeled as the interaction between an agent and an environment. The agent observes states within the environment, takes actions, and receives rewards.
2. State: States represent the environmental information observed by the agent. They can be partially observable or completely observable, depending on the specific problem.
3. Action: Actions are the operations the agent can choose within a state. They can be discrete (e.g., moving a chess piece) or continuous (e.g., controlling a robot's speed).
4. Reward: The reward is the numerical feedback received by the agent from the environment after executing an action, indicating the quality of the action. The agent's objective is to maximize cumulative rewards.
5. Policy: The policy defines how the agent selects actions based on observed states. It can be a deterministic policy or a stochastic policy.
6. Value Function: The value function estimates the expected value of cumulative rewards in the current state or state-action pairs. It includes the state-value function (V-value function) and the action-value function (Q-value function).

Policy Gradient Methods: DRL utilizes policy gradient methods to optimize the policy network, aiming to maximize the expected cumulative reward. This involves computing the policy gradient and using gradient ascent methods to update the policy network parameters.

Deep Neural Networks: In DRL, deep neural networks are commonly used to estimate Q-values, policies, or value functions. These deep neural networks can be convolutional neural networks (CNN) or recurrent neural networks (RNN), depending on the nature of the problem.

DRL methods such as Deep Q Networks (DQN), Deep Deterministic Policy Gradients (DDPG), A3C (Asynchronous Advantage Actor-Critic), and others have shown significant success in various domains, including gaming, robot control, autonomous driving, natural language processing, and more. By combining deep learning and reinforcement learning, these methods enable agents to autonomously learn and optimize policies in complex environments.