Reinforcement Learning Algorithms

Your Name

Semester Year

Q-Learning

Motivation:
Q-Learning is a model-free reinforcement learning algorithm that helps an agent learn how to act optimally in a given environment.

Q-Learning: Problem

Problem:
How does the agent learn the value of each action in a given state without a model of the environment?

Q-Learning: Intuitive Solution

Intuitive Solution:
The agent uses a Q-table to keep track of the value of actions in each state and updates this based on the rewards received after taking actions.

Q-Learning: Mathematical Formulation

Q-Value Update:
The Q-value is updated using the Bellman equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right) \quad (1)$$

Where:
- $Q(s_t, a_t)$: the current value of taking action a_t in state s_t
- r_t: reward received after taking action a_t
- α: learning rate
- γ: discount factor

Deep Q-Networks (DQN)

Motivation:
DQNs combine Q-learning with deep learning to handle high-dimensional observation spaces.

DQN: Problem

Problem:
How to apply Q-learning when the state space is large or continuous, making it impractical to maintain a Q-table?

DQN: Intuitive Solution

Intuitive Solution:
Use a neural network to approximate the Q-values for each action based on the state inputs.

DQN: Mathematical Formulation

Loss Function:
The loss function used for training the network can be defined as follows:

$$L(\theta) = \mathbb{E}\left[\left(r_t + \gamma \max_a Q(s_{t+1}, a; \theta^-) - Q(s_t, a_t; \theta)\right)^2\right] \quad (2)$$

Where θ are the parameters of the Q-network and θ^- are the parameters of the target network.

Policy Gradient Methods

Motivation:
Policy gradient methods directly optimize the policy instead of the value function, which can lead to better performance in some scenarios.

Policy Gradients: Problem

Problem:
How do we train a policy directly while ensuring stable and efficient updates?

Policy Gradients: Intuitive Solution

Intuitive Solution:
Use the gradient of expected return concerning the policy parameters to adjust the policy in the direction of increasing expected returns.

Policy Gradients: Mathematical Formulation

Policy Gradient Theorem:
The gradient of the expected return can be expressed as:

$$\nabla J(\theta) = \mathbb{E}\left[\nabla \log \pi_\theta(a|s) Q(s,a)\right] \tag{3}$$

Where $\pi_\theta(a|s)$ is the policy and $Q(s,a)$ is the action-value function.

Conclusion

Reinforcement learning encompasses various algorithms with distinct approaches to solving problems involving sequential decision-making. We explored Q-Learning, DQN, and Policy Gradients.

Questions?

Thank you for your attention! Any questions?