Gradient Policy Theory

MUKESH KUMAR

Goal of Reinforcement Learning

- In Reinforcement Learning (RL), we want an agent (like a robot or game player) to **learn the best way to act** in an environment to **maximize rewards** over time.

What is Policy Gradient?

- **Policy** = The strategy the agent uses to choose actions.

- Instead of learning values (like in Q-learning), **Policy Gradient methods directly learn the policy** — meaning, they learn the actual function that maps a state to the best action **probability**.

Key idea:

- We represent the policy with a **neural network** that outputs **probabilities** of actions.
- Example:
  - In a given state, the policy might say:
    - Move left: 20%
    - Move right: 80%

How does it learn?

- Policy Gradient methods adjust the policy so that:
- Actions that **led to high rewards become more likely** in the future.
- Actions that led to poor rewards become less likely.
- This is done using **gradient ascent** on the expected reward.

Core formula:

- The main update rule is:

$$\theta \leftarrow \theta + \alpha \cdot \nabla_\theta \log \pi_\theta(a|s) \cdot R$$

Where:

- θ = parameters of the policy (neural network weights)
- πθ(a|s) = probability of taking action a in state s
- R = reward received
- α = learning rate

Intuition:

- Imagine playing a game:
- Every time you win after doing a certain move, you **make that move more likely** next time.
- Over time, you **reinforce good behaviors**.

Advantages:

- Works well for **continuous action spaces**
- Can learn **stochastic policies** (not just deterministic ones)

Explain Gradient Policy using examples

- MouseMaze Example : refer *1Policy_Gradient_Mouse_Maze_Example.ipynb*
- SelfDrivingCar Example: refer *2.Example_Policy_Gradient_Self_Driving.ipynb*

Role of Gamma

- Refer notebook : 3.Gamma_Gradient_Policy.ipynb

Policy Gradient Theorem

- Please refer : 4.Policy Gradient Theorem.ipynb

Cart-Pole Problem

- https://gymnasium.farama.org/environments/classic_control/cart_pole/