UNIT 5: Reinforcement Learning

- Passive reinforcement learning
- Direct utility estimation
- Adaptive dynamic programming
- Temporal difference learning
- Active reinforcement learning- Q learning

2 Marks Questions

- What is rule based learning?
- Define temporal difference learning.
- Q-learning algorithm in reinforcement learning
- Temporal difference learning
- Passive reinforcement learning
- Adaptive dynamic programming
- What is Q-learning algorithm in reinforcement learning?
- What is adaptive dynamic programming?
- What is reinforcement learning?

5 Marks Questions

- Explain adaptive dynamic programming with suitable example.
- Explain Passive reinforcement learning in detail.
- Discuss the Q-learning algorithm in reinforcement learning.

10 Marks Questions

- What do you mean by Reinforcement Learning? Explain practical applications of RL.
- Explain adaptive dynamic programming and active reinforcement learning in detail with appropriate examples.
- Describe the importance of Q-learning algorithm in reinforcement learning with the help of suitable illustrations.
- Differentiate between active reinforcement learning and passive reinforcement learning.
- Explain Temporal Difference Learning and A* algorithm.

Reinforcement Learning (RL)

- Reinforcement Learning (RL) is a type of **machine learning** where an agent learns to make decisions by interacting with an environment.
- The agent performs actions, receives feedback in the form of rewards or penalties, and adjusts its strategy to maximize cumulative rewards over time.
- It learns through trial and error, making decisions based on rewards and penalties.

Key Components of Reinforcement Learning

- Agent: The entity (AI or model) that takes actions in an environment.
- Environment: The external system with which the agent interacts.
- State (s): A representation of the environment's condition at a given time.
- Action (a): The choices available to the agent at a given state.
- Reward (r): A numerical value given as feedback for the agent’s action.
- Policy (π): The strategy that the agent follows to decide which action to take in a given state.
- Value Function (V): Estimates the expected long-term reward from a state.
- Q-Function (Q): Estimates the expected long-term reward for a given state-action pair.

How Reinforcement Learning Works?

- Observation: The agent perceives the current state of the environment.
- Action Selection: Based on the state, the agent selects an action using a policy.
- Reward Assignment: The environment provides a reward or penalty based on the action.
- State Transition: The agent moves to a new state based on the action.
- Policy Update: The agent updates its policy using learning algorithms like Q-learning.
- Repeat: Steps 1-5 are repeated until the agent finds an optimal strategy.

Real World Example: Self-Driving Cars 🚗

Scenario: A self-driving car must learn how to drive safely and reach a destination while obeying traffic rules.

- **Agent:** The AI system controlling the car.
- **Environment:** The roads, traffic lights, pedestrians, and other vehicles.
- **State:** The car's speed, position, lane, distance to other objects, etc.
- **Actions:** Accelerate, brake, turn left, turn right, change lanes, etc.
- **Reward:**

- +10 points for following traffic signals.
- -50 points for breaking a red light.
- +100 points for safely reaching the destination.
- -100 points for an accident.

➢ The self-driving car learns by continuously interacting with the environment. Over time, it improves its policy by maximizing rewards (safe driving) and minimizing penalties (violations or accidents).

Practical Applications of RL

- Robotics 🤖
- Gaming 🎮
- Self-Driving Cars 🚗
- Finance & Trading 📈
- Healthcare & Medicine 🏥
- Natural Language Processing (NLP) 🗣️
- Manufacturing & Industrial Automation 🏭

Rule-Based Learning

- Rule-based learning is a traditional AI approach where decisions are made based on predefined rules or expert knowledge.
- It follows an "if-then" structure, making it less adaptable to dynamic environments compared to machine learning methods like reinforcement learning.

Example of Rule-Based Learning: Chatbots

A basic customer service chatbot that answers FAQs is an example of rule-based learning.

🔹 **Rule Example:**

- **IF** the user says: "What are your business hours?"
- **THEN** the chatbot replies: "Our business hours are 9 AM to 5 PM, Monday to Friday."

📌 **Limitations:**

- If the user asks, "When do you open?" (a different phrasing), the chatbot might not understand because it follows strict rules.
- It cannot **learn** new responses like a machine learning-based chatbot.

Comparison: Rule-Based vs. Machine Learning

Feature	Rule-Based Learning	Machine Learning
Decision Making	Based on fixed rules	Learns from data
Adaptability	Cannot handle new situations	Adapts to new data
Example	Simple chatbots	AI-powered assistants (e.g., Siri, Alexa)

Passive and Active RL

- **Passive Reinforcement Learning** is a type of **Reinforcement Learning (RL)** where an agent **follows a fixed policy** (predefined set of actions) and learns the **value of states** without actively exploring new actions.
- Unlike **Active RL**, where the agent chooses actions to maximize rewards, in **Passive RL**, the agent only **observes** and learns **how good or bad different states are** while following a fixed path.

Key Characteristics of Passive RL

- **Fixed Policy:** The agent does not choose its actions; it follows a given path.
- **Learns State Values:** The agent estimates how good each state is (its utility).
- **No Exploration:** It does not try new actions to find better rewards.

How Passive RL Works?

- The agent starts in a given environment.
- It follows a fixed policy (predefined actions).
- It observes rewards received for reaching different states.
- Over time, it learns the value of states (how beneficial a state is).

Main approaches to Passive RL

- **Direct Utility Estimation –** The agent calculates the average reward of each state based on past experiences.
- **Adaptive Dynamic Programming (ADP) –** Uses a model of the environment to calculate state utilities using the **Bellman Equation**.
- **Temporal Difference (TD) Learning –** Updates state values using the difference between estimated and actual rewards.

Example of Passive RL: A Robot 🤖 on a Fixed Path

- Imagine a **robot vacuum cleaner** moving in a house. It follows a **fixed path** (policy) and does **not decide** where to go. It only **learns** which areas are more beneficial (cleaner) based on rewards.
- **States:** Different locations in the house (e.g., kitchen, bedroom, living room).
- **Fixed Policy:** Always moves in the same direction.
- **Rewards:**

- +10 points for cleaning a dirty spot.
- -5 points for bumping into furniture.
- 0 points for moving normally.

What the Robot Learns

- The robot **does not change its path** but learns that some areas (like under the table) have **higher rewards** (more dirt collected).
- It updates its **understanding of state values** without making new decisions.

Active Reinforcement Learning

- Active Reinforcement Learning (Active RL) is a type of **Reinforcement Learning (RL)** where an **agent actively explores** different actions to **find the best policy** (the best way to behave).
- Unlike **Passive RL**, where the agent follows a fixed path, **Active RL allows the agent to make decisions** to maximize long-term rewards.

Key Characteristics of Active RL

- **Learns an Optimal Policy:** The agent does not follow a fixed path but instead learns the best actions to take.
- **Explores and Experiments:** The agent **tries different actions** to discover the most rewarding ones.
- **Maximizes Future Rewards:** The goal is to take actions that lead to the highest total reward over time.

How Active RL Works?

- The agent starts in an **environment** with **unknown rewards**.
- It **chooses actions** and observes the rewards received.
- It **explores** different actions to find the best way to behave.
- Over time, it **learns the best policy** (optimal set of actions).
- 👉 **Active RL often uses Q-learning**, a method where the agent stores and updates values for **state-action pairs** to determine the best action in each state.

Example of Active RL: A Self-Driving Car 🚙

Imagine a **self-driving car** learning to navigate a city.

States: The car's location on the road.
Actions: Move forward, turn left, turn right, stop.
Rewards:

- +10 points for reaching the destination safely.
- -5 points for taking a longer route.
- -100 points for crashing.

How Active RL Works in This Case:

- At first, the car **tries random routes** (exploration).
- It **learns from experience** which paths are safe and efficient.
- Over time, it **chooses the best route** to reach its goal quickly and safely.

Difference Between Active and Passive RL

Feature	Passive RL	Active RL
Policy	Fixed	Learns optimal policy
Exploration	No	Yes
Goal	Learn state values	Learn optimal actions

Temporal Difference (TD) Learning

- TD learning is an RL method that updates state values based on the difference between estimated and observed rewards.
- It combines ideas from Monte Carlo methods and Dynamic Programming to learn without needing a model of the environment.

Temporal Difference (TD) Learning

Formula:

$$U(s) \leftarrow U(s) + \alpha(r + \gamma U(s') - U(s))$$

Where:

- $U(s)$ = Utility of state s
- α = Learning rate
- γ = Discount factor
- r = Immediate reward
- $U(s')$ = Utility of the next state

Comparison of TD Learning and A* Algorithm

- TD Learning is used in reinforcement learning, focusing on estimating future rewards dynamically.
- A Algorithm* is used in pathfinding, finding the shortest path using heuristics.

Adaptive Dynamic Programming (ADP)

- ADP is an approach in RL that uses a model of the environment (transition probabilities and rewards) to compute state values using Bellman equations.
- Example of Adaptive Dynamic Programming
- • Chess AI: The model learns the value of board positions using past game data and optimizes strategies based on probability models.

Q-Learning Algorithm

- Q-learning is a model-free RL algorithm that learns the value of actions in each state to determine the optimal policy.

Q-Learning Algorithm

Q-Learning Formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

Where:

- $Q(s,a)$ = Quality of action a in state s
- r = Immediate reward
- γ = Discount factor (future reward weight)
- α = Learning rate
- $\max_{a'} Q(s',a')$ = Best possible future reward

Importance of Q-Learning

- It learns optimal policies even when the environment model is unknown.
- It is widely used in robotics, game AI, and self-driving cars.

Illustration of Q-Learning Process

1. The agent selects an action based on the Q-table.
2. It receives a reward and updates the Q-value.
3. It explores different actions and refines its policy over time.

Conclusion

- Reinforcement learning provides powerful methods for solving complex decision-making problems.
- Q-learning and Temporal Difference Learning are key techniques that allow agents to learn optimal strategies through trial and error.
- ADP further enhances learning by incorporating environment models.
- These concepts are fundamental to developing AI systems in robotics, gaming, and autonomous control.