Introduction to Reinforcement Learning

- A type of machine learning where an **agent** learns by interacting with the environment.
- Uses a system of **rewards and punishments** to guide decision-making.
- Inspired by **behavioral psychology**.

Key Elements of RL

1. **Agent** – Learner or decision-maker.
2. **Environment** – The world where the agent interacts.
3. **State (S)** – A representation of the environment.
4. **Action (A)** – Possible moves the agent can take.
5. **Reward (R)** – Feedback from the environment.
6. **Policy (π)** – Strategy to choose actions based on the state.
7. **Value Function (V)** – Expected long-term reward for being in a state.
8. **Q-Value (Q)** – Expected reward for taking a specific action in a given state.

Types of RL

1. Active RL

- The agent actively explores and takes actions to learn optimal policies.
- Uses **trial and error** to maximize rewards.
- Examples: **Q-learning, Deep Q Networks (DQN)**.
- **Example Scenario**: A robot trying different paths in a maze to maximize rewards.

2. Passive RL

- The agent **follows a fixed policy** and evaluates it.
- Learns the value of states under the given policy without exploring alternatives.
- Example Algorithms: **Policy Evaluation, Monte Carlo Methods**.
- **Example Scenario**: A self-driving car following a preset route while estimating travel time.

3. Model-Free RL

- The agent learns directly from interactions without a model of the environment.
- Examples: **Q-learning, SARSA**.
- **Use Case**: Video game AI improving through trial and error.

4. Model-Based RL

- The agent builds a model of the environment and plans actions based on it.
- Examples: **Dynamic Programming, Policy Iteration**.
- **Use Case**: A robot simulating possible moves before taking action.

5. Online RL

- The agent updates its knowledge continuously as it interacts with the environment.
- Example: **Stock trading AI learning in real time**.

6. Offline RL

- The agent learns from a pre-collected dataset without real-time interactions.
- Example: **Training autonomous cars on past driving data**.

Q-learning

- A **model-free, off-policy** reinforcement learning algorithm.
- Updates Q-values based on the highest possible future reward.
- Uses the Bellman equation:
- $\mathbf{Q(s, a) \leftarrow Q(s, a) + \alpha\, [r + \gamma \max Q(s', a') - Q(s, a)]}$
- where:
  - **α** = learning rate
  - **γ** = discount factor
  - **r** = reward
  - **max Q(s', a')** = maximum future reward estimate
- **Example Use Case**: Game AI improving performance by learning optimal moves.

SARSA (State-Action-Reward-State-Action)

- A **model-free, on-policy** RL algorithm.
- Learns action-value function based on the agent's own policy.
- Update rule:
- **$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma Q(s', a') - Q(s, a)]$**
- Difference from Q-learning: Uses the next action **a'** from the same policy instead of the max Q-value.
- **Example Use Case**: Learning in a **safe driving AI**, where it avoids risky actions based on its current policy.

Feature	Active RL	Passive RL
Policy	Learns optimal policy	Follows a fixed policy
Exploration	Yes	No
Decision-Making	Learns from actions taken	Evaluates given actions
Example	Q-learning	Policy Evaluation

Applications of RL

- **Gaming**: AlphaGo, Dota 2 AI.
- **Robotics**: Autonomous robots learning to walk.
- **Finance**: Stock trading strategies.
- **Healthcare**: Personalized treatment recommendations.
- **Self-driving Cars**: Learning to navigate safely.