Reinforcement Learning

Core Terminology

- **Agent**: The autonomous decision-maker that learns from experience.
- **Environment**: The external system the agent interacts with.
- **State (s)**: A representation of the current situation.
- **Action (a)**: A decision or movement the agent makes.
- **Reward (r)**: Immediate numerical feedback from the environment.
- **Policy (π)**: The strategy that maps states to actions.
- **Trajectory (Episode)**: A sequence of states, actions, and rewards.

1 Markov Decision Process (MDP)

Reinforcement learning problems are often modeled as a Markov Decision Process (MDP), which provides the mathematical framework.

Definition

An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$:

- $\mathcal{S}$: A set of states.
- $\mathcal{A}$: A set of actions.
- $P(s'|s, a)$: Transition probability of moving to state s' from s using action a.
- $R(s, a)$: Expected reward received after taking action a in state s.
- $\gamma \in [0, 1]$: Discount factor for future rewards.

Markov Property

The Markov property assumes that the future is independent of the past given the present:

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots) = P(s_{t+1}|s_t, a_t)$$

This implies that the current state captures all relevant information needed for decision-making.

2 Types of Reinforcement Learning

2.1 Model-Based vs. Model-Free RL

Model-Based RL	Model-Free RL
• Agents build a model of environment dynamics. • Enable planning via simulations. • More sample-efficient. • Sensitive to model inaccuracies.	• Learn directly from experience. • No internal model of the environment. • Easier to apply in complex scenarios. • Requires more data.

Healthcare Examples:

- **Model-Based:** Radiotherapy treatment planning, where the model predicts outcomes of different radiation doses.
- **Model-Free:** Adaptive insulin dosing for diabetes using real-time glucose monitoring.

2.2 On-Policy vs. Off-Policy RL

- **On-Policy** (e.g., SARSA):
  - Learns from the same policy it uses for action selection.
  - Safer in sensitive environments.
  - Example: Clinical support tools during live surgeries.
- **Off-Policy** (e.g., Q-Learning):
  - Learns from different behavior than the policy being improved.
  - Learns from historical or simulated data.
  - Example: Treatment planning from historical electronic health records (EHRs).

2.3 Value-Based vs. Policy-Based

- **Value-Based**:
  - Learns value functions like $Q(s, a)$.
  - Derives policy by maximizing values.
  - Examples: Q-Learning, DQN.
- **Policy-Based**:
  - Directly optimizes the policy $\pi(a|s)$.
  - Better for continuous action spaces.
  - Examples: REINFORCE, PPO.

3 Bellman Equations

3.1 Bellman Expectation Equation

The Bellman Expectation Equation expresses the value of a state (or action) under a policy π as the expected return starting from that state and following π thereafter.

- **State-Value Function:**

$$V^\pi(s) = \mathbb{E}_\pi \left[r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s \right]$$

- **Action-Value Function:**

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a \right]$$

Interpretation: The value of a state (or state-action pair) is the immediate reward plus the discounted value of the next state under policy π.

3.2 Bellman Optimality Equation

When seeking the optimal policy π^*, the Bellman Optimality Equation defines the best possible value function.

- **Optimal State-Value:**

$$V^*(s) = \max_a \mathbb{E} \left[r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a \right]$$

- **Optimal Action-Value:**

$$Q^*(s, a) = \mathbb{E} \left[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right]$$

Key Idea: The optimal value is obtained by choosing the action that leads to the best future value.

Healthcare Example

In treatment planning:

- s: current patient condition (e.g., lab values, vitals).
- a: treatment decision (e.g., drug dosage, imaging scan).
- r: immediate health outcome (e.g., symptom relief, complication risk).

The Bellman Equation models how the present decision influences long-term health outcomes, helping build policies for optimal care trajectories.

4 Key Reinforcement Learning Algorithms

4.1 Q-Learning (Off-Policy, Value-Based)

- Update rule:

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

- **Healthcare Example:** Optimizing sepsis treatment decisions (fluid resuscitation, vasopressors).

Algorithm 1 Q-Learning

Initialize $Q(s,a)$ arbitrarily
for each episode **do**
  Initialize state s
  repeat
    Choose a using ϵ-greedy policy
    Execute a, observe r and s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
    $s \leftarrow s'$
  until s is terminal
end for

4.2 SARSA (On-Policy, Value-Based)

- Update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s',a') - Q(s,a) \right]$$

- **Healthcare Example:** Robotic surgery systems that prioritize safety by learning only from actions taken.

Algorithm 2 SARSA

Initialize $Q(s, a)$ arbitrarily
for each episode **do**
    Initialize s, choose a from s
    repeat
        Take action a, observe r, s'
        Choose a' from s'
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$
        $s \leftarrow s'$, $a \leftarrow a'$
    until s is terminal
end for

4.3 Monte Carlo Methods

- Learn from complete episodes.
- Two main variants:
  - **First-Visit:** Update only first time (s, a) appears.
  - **Every-Visit:** Update all visits of (s, a).
- Update rule:
$$Q(s, a) \leftarrow Q(s, a) + \alpha[G_t - Q(s, a)]$$
- **Healthcare Example:** Treatment planning from full patient history.

Algorithm 3 First-Visit Monte Carlo Control

Initialize $Q(s, a)$, policy π
for each episode **do**
    Generate episode: $s_0, a_0, r_1, \ldots, s_T$
    $G \leftarrow 0$
    for $t = T - 1$ to 0 **do**
        $G \leftarrow \gamma G + r_{t+1}$
        if first visit of (s_t, a_t) **then**
            $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[G - Q(s_t, a_t)]$
            Update π to be ϵ-greedy
        end if
    end for
end for

4.4 Deep Q-Networks (DQN)

- Combines Q-learning with deep neural networks.
- Key techniques:

- **Experience Replay**: Breaks correlation between samples.
- **Target Network**: Stabilizes learning with delayed updates.

- Loss function:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[\left(r + \gamma \max_{a'} Q_{\text{target}}(s', a') - Q(s, a; \theta)\right)^2\right]$$

- Optimized using stochastic gradient descent or Adam.

DQN Architecture:

- **Input Layer:** Encodes state s (e.g., vital signs, image).
- **Hidden Layers:** Convolutional or fully connected layers.
- **Output Layer:** Q-values for each possible action.

Variants:

- **Double DQN:** Reduces overestimation of Q-values.
- **Dueling DQN:** Separates value and advantage estimations.
- **Prioritized Experience Replay:** Focuses on important transitions.

Healthcare Examples:

- **Chronic Disease Management:** Learning long-term treatment plans.
- **Personalized Drug Dosing:** Recommending dosages based on patient trajectory.
- **Radiology:** Optimal scan protocols, anomaly detection from image data.