Addis Ababa Institute of Technology

School of Information Technology and Engineering

AI Assignment 5

Name: Welela Bekele

ID: UGR/4983/15

Section:1

Submission Date:14/05/2025G.C

Submitted To: Mrs.Simreteab Mekbib

Q1. What is Reinforcement Learning (RL)?

Explain in your own words.

Reinforcement Learning (RL) is a form of machine learning in which an AI or a program learns from trying out different methods. The AI isn't given step by step instructions. Instead, it works within a system where it makes decisions and either gets rewards or gets punished based on its decisions.

Let's say you create an AI that plays a racing video game. The AI starts at a point where it does not even know how to operate a car. The AI will attempt to press buttons with the hope of something good happening. At this stage, the AI can get stuck in a loop where it crashes repeatedly, but with enough practice it will learn to speed forwards.

Reinforcement Learning (RL) differs from both supervised and unsupervised learning in what it's trying to achieve and how it learns. Supervised learning has a model that is trained using known data which comes in the form of labeled pairs. The model predicts an outcome based on examples it has learned where an input is like a solved problem—it's akin to teaching kids math using the answers. With no labeled data, a model has to find structures or patterns on its own in data. That is unsupervised learning and is akin to a child sorting block without being told how to do it. RL is different because it does not have a labeled set of inputs and relies on no passive discovery of a pattern in the data. Rather, an agent does learn everything through feedback to decisions by rewards or penalties and interacting with the environment to figure out conditions necessary for a long-term reward.

Q2. Define the following key components of an RL problem:

Agent(player): The AI or system that interacts with the environment and makes decisions to maximize rewards.

Environment: Everything outside the agent that responds to its actions. It's like the game world where the agent operates and learns.

State: A representation of environment at some moment provides the agent with critical information to making the next action decision.

Action: The options available to the agent at a particular state.

Reward: The agent receives feedback considering his action. Positive rewards stimulate good decisions while negative rewards discourage poor ones.

Policy: The strategy that the agent follows to decide its actions. It maps states to actions and can evolve over time to improve performance.

Value Function: A measure of how good a particular state is for long-term rewards. It helps the agent understand which states will likely lead to better outcomes.

Q-Value (Action-Value) Function: A function that estimates the value of taking a specific action in a given state. It helps the agent choose the best action to maximize rewards.

Q3. What is the goal of an RL agent?

Describe what the agent is trying to learn.

What does "maximizing cumulative reward" mean?

The **goal of a Reinforcement Learning (RL) agent** is to develop decision-making skills to optimize cumulative reward over time. The agent is attempting to learn a method that instructs it on what to do in each condition or circumstance in order to maximize the overall reward over time.

"**Maximizing cumulative reward**" implies that the agent isn't looking for just after each action a reward whether immediate or short-term, but rather trying to string together actions which will yield the biggest payoff in the long haul. This requires trade-offs between short-term rewards and long-term consequences and often involves the agent taking actions that seem counterproductive in the short run.

Q4. What is the difference between:

Exploration and Exploitation?

Why is balancing them important in RL?

- **Exploration**: The use of novel actions by an RL agent in a bid to find new and hopefully better strategies. Like a scientist trying different solutions to problems.
- **Exploitation**: The step in the algorithm where the agent uses its pre-existing knowledge for maximizing rewards. It follows the best-known strategy instead of trying new approaches.

Why Balancing Exploration and Exploitation Matters

If the agent only **explores**, it may never settle on a good strategy—it keeps searching but never fully optimizes its actions. If the agent only **exploits**, it may get stuck in a suboptimal strategy—missing better solutions because it stopped exploring.

An ideal RL agent finds a balance: exploring enough to discover new opportunities while exploiting well enough to maximize rewards.

Q5. What is the Markov Decision Process (MDP)?

List and explain the components of an MDP.

Why is the Markov property important?

A **Markov Decision Process (MDP)** is a mathematical model utilized in Reinforcement Learning to describe decision-making processes where outcomes are probabilistic but partially determined by the decision of an agent.

Components of an MDP

An MDP consists of the following elements:

1. **States (S)**: Represents the possible situations in which the agent can find itself.
2. **Actions (A)**: The available options an agent can execute given each state.
3. **Transition Probability (P)**: The likelihood of moving from one state to another upon performing an action.
4. **Reward (R)**: Monetary compensation provided to the agent for performing certain actions while in a particular state.
5. **Policy (π)**: A strategy that defines which actions the agent should take in different states.
6. **Discount Factor (γ)**: A value between 0 and 1 that determines the importance of future rewards compared to immediate rewards.

Why is the Markov Property Important?

Markov property states that the system's future state depends on the current state and action only and not previous states. This enhances simplicity in decision-making because the agent needs to take only the current situation into account instead of the entire sequence of events.

For example, in a game of chess, a simplified RL model might only consider the **current board position** rather than the entire sequence of past moves. This makes computations more efficient and practical.

Q6. Compare and contrast:

Policy-based vs Value-based RL

Model-free vs Model-based RL

Feature	**Policy-based RL**	**Value-based RL**
What it learns	**Learns the policy directly (π = action strategy)**	**Learns the value function (V(s) or Q(s, a))**
Decision making	**Chooses actions directly using the policy**	**Chooses actions by maximizing value estimates**

Stochastic policies	**Can naturally learn stochastic (randomized) policies**	**Typically learns deterministic policies**
Use cases	**Better for high-dimensional or continuous action spaces**	**Works well with discrete action spaces**
Example algorithm	**REINFORCE, PPO**	**Q-learning, Deep Q-Network (DQN)**

Feature	**Model-free RL**	**Model-based RL**
What it uses	Learns directly from **experience** without modeling the environment	Builds or uses a **model** of the environment
Planning	No planning; learns via trial-and-error	Uses the model for **planning** and simulation
Sample efficiency	Usually **less sample-efficient**	Generally **more sample-efficient**
Computation	Simpler, but may need more data	More complex, but can use fewer real interactions
Example algorithm	Q-learning, Policy Gradient	Dyna-Q, MuZero

Q7. What are some real-world applications of Reinforcement Learning?

Give at least 3 examples and describe how RL is used in each.

1.Personalized Recommendations

- How RL is used: RL enables automated systems to adapt to user preferences by figuring out which suggestions provide the highest user interaction.

- Example: In platforms like YouTube or Netflix, RL algorithms learn which videos or shows to recommend by observing user interactions (clicks, watch time, etc.) and adjusting future recommendations to maximize user satisfaction.

2. Autonomous Vehicles (Self-Driving Cars)

- RL encourages self-driving cars to learn safe navigational skills through decision-making based on the surroundings and environment feedback loops.
- Example: RL helps autonomous vehicles optimize brake, acceleration, and lane-change maneuvers to conserve fuel while enhancing safety.

3. Finance & Trading

- RL is widely used in **algorithmic trading** to make smarter investment decisions.
- Example: Banking and other financial services build reinforcement learning models to optimize trading activities to predict market trends for profit maximization and risk aversion.