Task 2.1: Explain the exploration vs. exploitation problem in RL

Exploration: Trying out different actions to gain more information about the environment. This helps in discovering potentially better actions that might lead to higher rewards in the long term. Exploration is essential for learning and improving the policy.

Exploitation: Leveraging current knowledge to select actions that are believed to be the best based on the learned policy. Exploitation aims to maximize immediate rewards based on existing information.

The exploration vs. exploitation problem in reinforcement learning (RL) refers to the dilemma of deciding whether to explore new actions or exploit the current knowledge to maximize rewards.

Task 2.2: Explain the credit-assignment problem in RL

The credit assignment problem (CAP) is a fundamental challenge in reinforcement learning. It arises when an agent receives a reward for a particular action, but the agent must determine which of its previous actions led to the reward. In other words, The credit assignment problem refers to the problem of measuring the influence and impact of an action taken by an agent on future rewards.

Task 2.3: What is the Markov property?

A stochastic process has the Markov property if the conditional probability distribution of future states of the process (conditional on both past and present values) depends only upon the present state; that is, given the present, the future does not depend on the past.

Task 2.4: Motivate whether or not chess satisfies the Markov property

Chess does not satisfy the Markov property because the outcome of a game state (i.e., future states) depends not only on the current board configuration but also on the history of moves leading to that state. The effectiveness of certain moves, positions, or strategies in chess can be influenced by previous game states and sequences of actions, which violates the condition of state independence in the Markov property.

Task 2.5: Difference between Q-learning and Deep Q-learning

Q-learning is tabular and requires discrete state and action spaces, while Deep Q-learning can handle continuous and high-dimensional state spaces using neural networks.

Deep Q-learning can generalize across similar states and learn complex state-action mappings, whereas Q-learning is limited to explicit state-action pairs.

Deep Q-learning is more scalable to complex environments but requires additional training and may suffer from stability issues (e.g., divergence) without proper techniques like experience replay and target networks.