Assignment 1: Reinforcement Learning and Deep Learning

Question 1: Define Reinforcement Learning with Its Features

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment. The goal of the agent is to maximize cumulative rewards through trial-and-error learning. In RL, the agent receives feedback in the form of rewards or penalties based on its actions, which helps it learn optimal strategies over time.

Features of Reinforcement Learning:

1. **Agent-Environment Interaction**: RL involves an agent that takes actions in an environment to receive rewards or penalties based on those actions.
2. **Exploration vs. Exploitation**: The agent must balance exploring new actions to discover their effects and exploiting known actions that yield high rewards.
3. **Delayed Rewards**: Unlike supervised learning, RL can involve delayed rewards where the consequences of an action are not immediately apparent.
4. **Policy**: The strategy that the agent employs to determine its actions based on the current state of the environment. Policies can be deterministic or stochastic.
5. **Value Function**: A function that estimates the expected return or future rewards from a particular state or state-action pair, helping the agent evaluate its actions.
6. **Temporal Difference Learning**: A method used in RL to update the value function based on the difference between predicted and actual rewards over time.

Question 2: Explain Q-Learning, Justify with an Example

Q-Learning is a model-free reinforcement learning algorithm used to learn the value of an action in a particular state. The goal of Q-learning is to learn a policy that maximizes the total reward for an agent over time. It does so by updating Q-values, which represent the expected utility of taking a given action in a given state.

Q-Learning Algorithm Steps:

1. **Initialize** the Q-table with arbitrary values (often zero).
2. For each episode:
   - Initialize the starting state.
   - For each step in the episode:
     - Choose an action using an exploration strategy (e.g., ε-greedy).
     - Take the action and observe the reward and next state.

    - Update the Q-value using the formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) (1) \text{where:} \begin{aligned} \alpha &= \text{learning rate} \\ r &= \text{reward} \\ \gamma &= \text{discount factor} \\ s &= \text{current state} \\ a &= \text{action taken} \\ s' &= \text{next state} \end{aligned}$$

3. **Repeat** until convergence or a maximum number of episodes is reached.

Example:

Consider a simple grid world where an agent moves in a 5×5 grid, with the goal of reaching a target cell. The agent receives a reward of +10 for reaching the target and -1 for each move. Initially, the Q-values for each state-action pair are zero.

- The agent starts in the bottom-left corner (state (0,0)).
- It takes actions (up, down, left, right) based on its exploration strategy.
- After several episodes of exploration and updating the Q-values, the agent learns that moving right and then up yields the highest rewards. Eventually, the Q-table reflects that moving from (0,0) to (1,1) (the target) is optimal.

Question 3: Compare and Contrast Dueling DQN and Prioritized Experience Replay

Feature	Dueling DQN	Prioritized Experience Replay
Architecture	Separates the estimation of state values and advantages into two streams, allowing better value estimation.	Focuses on replaying important experiences more frequently to improve learning efficiency.
Q-Value Representation	Estimates the value of a state and the advantage of each action separately.	Does not alter the Q-value representation but prioritizes the sampling of experiences.
Learning Efficiency	Improves learning efficiency by providing more accurate Q-value estimates in complex environments.	Enhances learning by focusing on experiences that have a higher impact on learning (higher TD error).
Implementation Complexity	More complex due to the dual network architecture.	Requires additional mechanisms to compute priorities and adjust sampling probabilities.
Use Case	Beneficial in environments where the value of being in a state is more significant than the actions taken.	Useful when some experiences are significantly more informative than others.

Conclusion

Both Dueling DQN and Prioritized Experience Replay aim to improve the efficiency and effectiveness of deep reinforcement learning. While Dueling DQN focuses on enhancing the representation of Q-values by separating value and advantage, Prioritized Experience Replay aims to optimize the learning process by focusing on more important experiences. Combining these two techniques can potentially lead to even better performance in reinforcement learning tasks.

Feel free to customize the content and add your insights as needed!