**Assignment-3**

|  |
| --- |
| Markov Decision Process |

**Q1. What is Reinforcement Learning? Draw suitable diagram and describe elements of Reinforcement Learning?**

* **Reinforcement** learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.
* Reinforcement learning differs from supervised learning in not needing labelled input/output pairs to be presented, and in not needing sub-optimal actions to be explicitly corrected.
* Instead, the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).
* Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error.
* It uses algorithms that learn from outcomes and decide which action to take next.
* It performs actions with the aim of maximizing rewards, or to achieve the best outcomes.
* After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect.
* **Elements of Reinforcement Learning:**
* **Agent:** The agent is the learning algorithm or decision-making entity that interacts with the environment. The agent takes actions in the environment and receives feedback in the form of rewards or penalties.
* **Environment:** The environment is the external system with which the agent interacts. The environment provides the agent with information about the current state of the system and the set of possible actions it can take.
* **Reward:** The reward is the feedback that the agent receives from the environment in response to its actions. The reward signal is a scalar value that indicates the quality of the agent's performance at a particular time step.
* **Actions:** The actions are the decisions made by the agent in the environment. The set of possible actions depends on the specific problem being solved. For example, in a game, the set of actions might be the different moves that a player can make.
* **Policy:** The policy is the strategy or rule that the agent uses to determine its actions in the environment. The policy maps states to actions and can be deterministic or stochastic. A deterministic policy always chooses the same action for a given state, whereas a stochastic policy chooses actions probabilistically based on the state.

**Q2. Explain Exploration vs Exploitation dilemma?**

* **Exploitation and exploration are the key concepts in Reinforcement Learning, which help the agent to build online decision making in a better way.**
* **Exploitation in Reinforcement Learning:** Exploitation is defined as a greedy approach in which agents try to get more rewards by using estimated value but not the actual value. So, in this technique, agents make the best decision based on current information.
* **Exploration in Reinforcement Learning:** Unlike exploitation, in exploration techniques, agents primarily focus on improving their knowledge about each action instead of getting more rewards so that they can get long-term benefits. So, in this technique, agents work on gathering more information to make the best overall decision.
* Reinforcement learning is a machine learning method in which an intelligent agent (computer program) learns to interact with the environment and take actions to maximize rewards in a specific situation. This ML method is currently being used in so many industries such as automobile, healthcare, medicine, education, etc.
* As in Reinforcement learning, the agent is not aware of the different states, actions for each state, associate Rewards, and transition to the next state, but it learns it by exploring the environment. However, the knowledge of an agent about the state, actions, rewards, and resulting states is partial, and this results in **Exploration-Exploitation Dilemma**.
* In reinforcement learning, whenever agents get a situation in which they have to make a difficult choice between whether to continue the same work or explore something new at a specific time, then, this situation results in Exploration-Exploitation Dilemma because the knowledge of an agent about the state, actions, rewards and resulting states is always partial.
* **There are a examples of Exploitation and Exploration in Machine Learning as follows:**
* **Example 1:** Let's say we have a scenario of online restaurant selection for food orders, where you have two options to select the restaurant. In the first option, you can choose your favorite restaurant from where you ordered food in the past; this is called **exploitation** because here, you only know information about a specific restaurant. And for other options, you can try a new restaurant to explore new varieties and tastes of food, and it is called exploration. However, food quality might be better in the first option, but it is also possible that it is more delicious in another restaurant.
* **Example 2:** Suppose there is a game-playing platform where you can play chess with robots. To win this game, you have two choices either play the move that you believe is best, and for the other choice, you can play an experimental move. However, you are playing the best possible move, but who knows new move might be more strategic to win this game. Here, the first choice is called exploitation, where you know about your game strategy, and the second choice is called exploration, where you are exploring your knowledge and playing a new move to win the game.

**Q3. Describe Epsilon Greedy Algorithm?**

|  |
| --- |
| q learning epsilon greedy 1 |

* The epsilon-greedy algorithm is a simple but effective exploration-exploitation strategy that is widely used in reinforcement learning. At each time step, the agent has a choice between exploiting the best-known action or exploring a new action. Exploitation refers to selecting the action with the highest expected reward based on the agent's current knowledge, while exploration refers to selecting a new action at random.
* The epsilon-greedy algorithm takes a parameter **ε**, which determines the probability of choosing a random action. When ε is set to 0, the agent will always choose the action with the highest expected reward, meaning it will always exploit. When ε is set to 1, the agent will always choose a random action, meaning it will always explore. A value of ε between 0 and 1 allows the agent to balance its exploration and exploitation, with higher values of ε encouraging more exploration and lower values encouraging more exploitation.
* **For example,** let's say the agent is faced with two actions, A and B, and has learned that A has a higher expected reward than B. If ε=0.1, the agent will choose action A with a probability of 0.9 (1-ε) and choose action B (or another random action) with a probability of 0.1 (ε). This allows the agent to mostly exploit the best-known action, while still occasionally exploring other actions.
* The epsilon-greedy algorithm is simple and easy to implement, but it does have some limitations. One limitation is that it does not take into account the uncertainty of the agent's estimates of the expected rewards. In some cases, the agent may be overly confident in its estimates and miss out on better rewards. Another limitation is that the choice of ε can have a significant impact on the agent's performance. Choosing the right value of ε requires knowledge of the problem being solved and the properties of the environment.
* Despite these limitations, epsilon-greedy is a widely used exploration-exploitation strategy in reinforcement learning. It provides a simple and effective way for the agent to balance its exploration and exploitation, and can be easily combined with other reinforcement learning algorithms.

**Q4. Draw suitable diagram and describe Markov Decision Process (MDP)?**

* The MDP model operates by using key elements such as the agent, states, actions, rewards, and optimal policies. The agent refers to a system responsible for making decisions and performing actions. It operates in an environment that details the various states that the agent is in while it transitions from one state to another.
* MDP defines the mechanism of how certain states and an agent’s actions lead to the other states. Moreover, the agent receives rewards depending on the action it performs and the state it attains (current state). The policy for the MDP model reveals the agent’s following action depending on its current state.
* **The MDP framework has the following key components:**
* S: states (s ∈ S)
* A: Actions (a ∈ A)
* P (St+1|st.at): Transition probabilities
* R (s): Reward
* **The graphical representation of the MDP model is as follows:**

|  |
| --- |
| MDP model |

* The MDP model uses the Markov Property, which states that the future can be determined only from the present state that encapsulates all the necessary information from the past. The Markov Property can be evaluated by using this equation:

**P[St+1|St] = P[St+1 |S1,S2,S3……St]**

* According to this equation, the probability of the next state (P[St+1]) given the present state (St) is given by the next state’s probability (P[St+1]) considering all the previous states (S1,S2,S3……St). This implies that MDP uses only the present/current state to evaluate the next actions without any dependencies on previous states or actions..

|  |
| --- |
| MDP schema Wikipedia |

* **The process of selecting an action from a given state**, transitioning to a new state, and receiving a reward happens sequentially over and over again.
* Which creates something called a trajectory that shows the sequence of states, actions, and rewards.
* Throughout this process, it is the agent's goal to maximize the total amount of rewards that it receives from taking actions in given states.
* This means that the agent wants to maximize not just the immediate reward, but the cumulative rewards it receives over time.
* **WORKING:**
* In order to understand how the MDP works, first the Markov Property must be defined.
* The Markov Property states that the future is independent of the past given the present.
* In other words, only the present is needed to determine the future, since the present contains all necessary information from the past.
* The Markov Property can be described as:
* A state St is Markov if and only if **P[St+1|St] = P[St+1|S1,S2,S3,S4,…,St]**
* The probability of the next state given the current state is equal to the probability of the next state given all previous states.

**Q5. Derive necessary equation and Brief Q Learning Algorithm?**

* In Q-learning, the agent learns an optimal action-value function Q(s,a) that gives the expected cumulative reward for taking action a in state s and following the optimal policy thereafter. The optimal action-value function satisfies the following Bellman equation:
* **Q\*(s,a) = E[R\_{t+1} + γ max\_{a'} Q\*(s\_{t+1},a') | s\_t = s, a\_t = a]**,
* where R\_{t+1} is the reward obtained by taking action a in state s and transitioning to state s\_{t+1}, and γ is the discount factor.
* The Q-learning algorithm updates the estimate of Q(s,a) based on the observed reward and the next state. The update rule is as follows:
* **Q(s\_t,a\_t) ← Q(s\_t,a\_t) + α [R\_{t+1} + γ max\_{a'} Q(s\_{t+1},a') - Q(s\_t,a\_t)]**,

where α is the learning rate that determines how much weight is given to the new observation, and max\_{a'} Q(s\_{t+1},a') is the maximum expected cumulative reward over all possible actions in the next state.

* **The Q-learning algorithm can be summarized as follows:**
* Initialize the Q-values for all state-action pairs arbitrarily.
* Repeat for each episode:

1. Observe the current state s\_t.
2. Choose an action a\_t using an exploration/exploitation strategy, such as ε-greedy.
3. Take action a\_t and observe the next state s\_{t+1} and the reward R\_{t+1}.
4. Update the Q-value for the current state-action pair using the Q-learning update rule.
5. Set s\_t+1 as the new current state.

* Return the learned Q-values.

Q-learning is a reinforcement learning algorithm that learns an optimal action-value function by iteratively updating the estimates based on the observed rewards and the next state. The Q-learning update rule uses the Bellman equation to update the estimate of the Q-value for the current state-action pair. The Q-learning algorithm is a widely used and effective method for solving reinforcement learning problems.

|  |
| --- |
| https://miro.medium.com/v2/resize:fit:700/1*Wim9wr-jYJtZyZ_4ZJRHNg.jpeg |

**(PPT Give you more understanding than PDF)** The material for the PDF has been compiled from various sources such as books, tutorials (offline and online), lecture notes, several resources available on Internet. The information contained in this PDF is for general information and education purpose only. While we endeavor to keep the information up to date and correct, **we make no representation of any kind about the completeness and accuracy of the material.** The information shared through this PDF material should be used for educational purpose only.

**Thank you…**