***Machine Learning Lecture 2***

Overview and History:

* Reinforcement learning (RL) is machine learning paradigm where an agent learns by interacting with its environment
* It has roots in the 1950s with applications in game playing and recent breakthroughs like DeepMind’s Atari-playing system and AlphaGo have showcased its potential

The RL Agent and Policy:

* An RL agent is a software program that perceives its environment through sensors, takes actions according to a policy and learns from rewards (or penalites)
* The policy is the decision-making algorithm the agent uses, which can be deterministic or stochastic
* Policy search involves finding the best parameters for the policy using methods like brute-force search genetic algorithms or gradient based optimization

Exploration vs Exploitation

* A key challenge in RL is the balance between explotation (choosing actions known to yield high rewards) and exploration (trying new cations to discover potentially better rewards)
* Techniques such as the ε-greedy strategy help manage this trade-off by sometimes choosing a random action

The Credit Assignment Problem:

* In RL, rewards can be delayed making it difficult to know which actions contributed to a final outcome
* This is addressed by computing the action return (the discounted sum of future rewards using a discount factory ) and the action advantage which normalizes how much better or worse an action is compared to the average

Markov Decision Processes (MDP’s):

* MDPs extend Markov chains by incorporating actions and rewards. In an MDP the agent selects an action in each state and the system transition to a new state with an associated reward
* Optimal state Values (V\*(s)) represent the maximum expected discounted reward starting from state ‘s and methods like value iteration are used to compute these values

Q Learning and Temporal Difference Learning:

* Q-Learning is an off-policy Temporal Difference learning algorithm that estimates the Q-value (expected cumulative reward) for each state-action pair, even when transition probabilities and rewards are unknown
* It updates Q-values incrementally while the agent explores the environment gradually refining its policy
* The algorithm uses a learning rate (α) that is typically reduced over time for convergence

Neural Network Policies:

* Modern RL can employ neural networks to approximate policies where the network takes observations as inputs and outputs a probability distribution over actions
* This approach is particularly useful when dealing with high- dimensional or complex environment’s

***Quiz Questions and Answers***

1. Q: What is Reinforcement Learning?  
   A: Reinforcement Learning is a machine learning paradigm in which an agent learns to make decision by interacting with its environment receiving rewards or penalties and updating its policy to maximize cumulative rewards
2. Q: What is historical breakthrough’s have contributed to the popularity of RL?  
   A: Early RL was applied to game playing in the 1950’s. More recently DeepMind’s system in 2013 that learned to play Atari games from scratch and AlphaGo which beat world champions in Go (2016-2017) have been significant breakthroughs
3. Q: Define an RL agent and explain its role  
   A: As an RL agent is a software program that interacts with its environment by perceiving data through sensors selecting actions based on a policy and receiving rewards or penalties to learn optimal behavior over time
4. Q: What is a policy in the context of Reinforcement Learning?  
   A: A policy is the strategy or algorithm that the agent uses to decide which action to take in a given state. It can be deterministic or stochastic and is often refined via policy search methods
5. Q: How does the exploration vs exploitation dilemma affect an RL agent?  
   A: The dilemma involves balancing the need to exploit actions that yield known rewards with the need to explore new actions that might result in even better rewards. Techniques like the ε-greedy strategy help maintain the balance
6. Q:What is the credit assignment problem and how is it addressed in RL?  
   A: The credit assignment problem refers to the difficulty or determining which actions are responsible for received rewards especially when rewards are delayed. It is addressed by computing the action return the discounted sum of future rewards and by evaluating the action advantage
7. Q: What is Markov Decision Process (MDP) and what are its key components?  
   A: An MDP is a framework for modeling decision-making in situations where outcomes are partly random and partly controlled by an agent. Its key component’s include states actions transition probabilities and rewards. The goal is to maximize the cumulative discounted reward
8. Q: What is the purpose of the value iteration algorithm in MDP’s  
   A: The value iteration algorithm is used to compute the optimal state values by iteratively updating them based on expected discounted rewards, eventually converging to the optimal policy for the MDP.
9. Q: Explain Q-Learning and its significance in Reinforcement Learning?  
   A: Q-Learning is an off-policy Temporal Difference Learning algorithm that estimates the Q values for state action pairs. It allows an agent to learn an optimal policy through trial and error updating its estimates as it interacts with the environment even when the transition dynamics are unknown.
10. Q: What does it mean that Q-Learning is an off-policy algorithm?  
    A: Being off-policy means that the Q Learning algorithm learns the optimal policy independently of the agent’s current exploration strategy. This allows the agent to explore the environment using one strategy while learning the value of the best possible actions.
11. Q: How are neural network used in modern RL?  
    A: Neural Network policies take observations as input and output a probability distribution over possible actions enabling the agent to choose actions probabilistically. This method is effective in handling high-dimensional or complex environments where traditional approaches may struggle