Weekly Report: Oudarja Barman Tanmoy Alpha AI

Week-05 (April 21 - April 25)

1. **Machine Learning specialization :**
    a. **Course 03 : Unsupervised Learning, Recommenders, Reinforcement Learning**
        i. Gained a strong foundation in reinforcement learning concepts, including return, state, action, and policy.
        ii. Developed an understanding of Bellman equations and their role in reinforcement learning models.
        iii. Explored the state-action value function and its use in dynamic decision-making environments.
        iv. Addressed the complexities of working with continuous state spaces.
        v. Successfully implemented a Deep Q-Learning Network (DQN) through hands-on assignments.
2. **Mathematics for Machine Learning: Multivariate Calculus:**
    a) **Week 3:**
        1. Utilized the multivariate chain rule to differentiate complex, nested functions.

2. Examined the architecture and operational principles of neural networks.

3. Applied multivariate calculus techniques to understand how network parameters influence outputs.
   Implemented the backpropagation algorithm on a simplified neural network to deepen understanding of learning dynamics.

b) **Week 4:**

1. Analyzed function approximations using power series.

2. Investigated the behavior of power series when applied to irregular or non-smooth functions.

3. Explored the use of linearization for practical function approximation.

4. Evaluated and applied suitable methods for approximating multivariate functions based on function characteristics and approximation goals.

3) **OOP Unit Testing with Pytest :**

a) Developed comprehensive unit tests for Object-Oriented Programming concepts using `pytest`, covering the following areas:

1. **Classes and Objects:** Tested object initialization, attribute handling, and method behavior.

2. **Inheritance:** Validated correct inheritance and method overriding from base classes.

3. **Polymorphism:** Assessed dynamic method binding and function overloading.

4. **Abstraction:** Verified implementation and correctness of abstract base classes.

5. **Encapsulation:** Ensured effective data hiding and controlled access to internal attributes.

6. **Decorators:** Tested the behavior and impact of method decorators.

7. **Properties:** Confirmed proper functionality of getters and setters using Python's `@property` decorators.