Assignment 2: Energy Consumption Optimization (15Marks)

Instructions:

- Read the assignment proposal carefully.
- If any of the requirements are missed in the final code submission, the respective marks will be deducted.
- It is mandatory to submit the assignment in the PDF format only consisting of all the outcomes with each and every iteration. Any other format will not be accepted.
- Add comments and description to every function you are creating or operation you are performing. If not found, then 1 mark will be deducted. There are many assignments that need to be evaluated. By providing the comments and description it will help the evaluator to understand your code quickly and clearly.
- Maintain the same naming conventions for the PDF files to be submitted as that of ipynb files.
- Submit 2 different PDFs. One forActor-Critic and One for DQN & DDQN.
- Late submissions will lead to a deduction of 1 Mark.

Problem Statement: The objective of the problem is to implement an Actor-Critic reinforcement learning algorithm to optimize energy consumption in a building. The agent should learn to adjust the temperature settings dynamically to minimize energy usage while maintaining comfortable indoor conditions.

Dataset: https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction

This dataset contains energy consumption data for a residential building, along with various environmental and operational factors.

Data Dictionary:

o Appliances: Energy use in Wh
o lights: Energy use of light fixtures in the house in Wh
o T1 - T9: Temperatures in various rooms and outside
o RH_1 - RH_9: Humidity measurements in various rooms and outside
o Visibility: Visibility in km
o Tdewpoint: Dew point temperature
o Press_mm_hg: Pressure in mm Hg
o Windspeed: Wind speed in m/s

State Space:

The state space consists of various features from the dataset that impact energy consumption and comfort levels.

• Current Temperature (T1 to T9): Temperatures in various rooms and outside.
• Current Humidity (RH_1 to RH_9): Humidity measurements in different

locations.

- Visibility (Visibility): Visibility in km.
- Dew Point (Tdewpoint): Dew point temperature.
- Pressure (Press_mm_hg): Atmospheric pressure in mm Hg.
- Windspeed (Windspeed): Wind speed in m/s.

Total State Vector Dimension: Number of features = 9 (temperature) + 9 (humidity) + 1 (visibility) + 1 (dew point) + 1 (pressure) + 1 (windspeed) = 21 features

Target Variable: Appliances (energy consumption in Wh).

Action Space:

The action space consists of discrete temperature adjustments:

- Action 0: Decrease temperature by 1°C
- Action 1: Maintain current temperature
- Action 2: Increase temperature by 1°C

Adjustments are clamped within the defined temperature limits (-10°C to 30°C).

If the action is to decrease the temperature by 1°C, you'll adjust each temperature feature (T1 to T9) down by 1°C. If the action is to increase the temperature by 1°C, you'll adjust each temperature feature (T1 to T9) up by 1°C. Other features remain unchanged.

The action space is limited to discrete temperature adjustments (±1°C) within a defined range (-10°C to 30°C).

Policy (Actor): A neural network that outputs a probability distribution over possible temperature adjustment.

Value function (Critic): A neural network that estimates the expected cumulative reward (energy savings) from a given state.

Reward function:

The reward function should reflect the overall comfort and energy efficiency based on all temperature readings. i.e., balance between minimising temperature deviations and minimizing energy consumption.

- Calculate the penalty based on the deviation of each temperature from the target temperature and then aggregate these penalties.
- Measure the change in energy consumption before and after applying the RL action.
- Combine the comfort penalty and energy savings to get the final reward.

The RL framework integrates these adjustments by modifying the temperature features in the state vector, computing rewards based on energy savings and comfort penalties, and training the Actor-Critic model to find an optimal policy.

Example:

Target temperature=22°C

Initial Temperatures: T1=23, T2=22, T3=21, T4=23, T5=22, T6=21,

T7=24, T8=22, T9=23 Action Taken: Decrease temperature by 1°C for each

room

Resulting Temperatures: T1 = 22, T2 = 21, T3 = 20, T4 = 22, T5 = 21, T6 = 20, T7 =

23, T8 = 21, T9 = 22 Energy Consumption: 50 Wh (before RL adjustment) and 48 Wh

(after RL adjustment)

- Energy Before (50 Wh): Use the energy consumption from the dataset at the current time step.
- Energy After (48 Wh): Use the energy consumption from the dataset at the next time step (if available).

Deviation = abs ($T_i - T_{Target}$)

Deviations=[abs(22−22), abs(21−22), abs(20−22), abs(22−22), abs(21−22), abs(20−22), abs(23−22), abs(21−22), abs(22−22)]

Deviations = [0, 1, 2, 0, 1, 2, 1, 1, 0], Sum of deviations = 8

Energy Savings = Energy Before−Energy After = 50 – 48 = 2Wh

Reward= −Sum of Deviations + Energy Savings = -8+6 = -2

Requirements:

- Pre-process the dataset to handle any missing values and create training and testing sets.

Actor-Critic Problem: (7 Marks)

- Design the Actor-Critic algorithm using TensorFlow. (1 M)
- Design Reward Function. (0.5 M)
- Environment Solution (0.5 M)
- Train the model over 500 episodes to minimize energy consumption while maintaining an indoor temperature of 22°C. (2 M)
- Evaluate the performance of the model on test set to measure its performance (0.5 M)
- Provide graphs showing the convergence of the Actor and Critic losses. (1 M)
- Plot the learned policy by showing the action probabilities across different state values (e.g., temperature settings). (1 M)
- Provide an analysis on a comparison of the energy consumption before and after applying the reinforcement learning algorithm. (0.5 M)

DQN & DDQN: (8 Marks)

- Implement the DQN and DDQN algorithm
- Design an EnergyConsumption Environment. (1 M)
    - o Print the state space and action space (0.5 M)
    - o Clearly define the parameters used for training an AI agent. (1 M)
        - Number of episodes
        - Max capacity of replay memory
        - Batch size
        - Period of Q target network updates
        - Discount factor for future rewards
        - Initial value for epsilon of the e-greedy
        - Final value for epsilon of the e-greedy
        - Learning rate of ADAM optimizer, and etc.
    - o Define the functions for DecreaseTemperature, IncreaseTemperature and MaintainCurrentTemperature actions. (1.5 M)
    - o Implement a replay buffer for storing the experiences. (0.5 M)
    - o Design DQN
        - Network Design (0.5 M)
        - Model Training (0.5 M)
    - o Design DDQN
        - Network Design (0.5 M)
        - Model Training (0.5 M)
- Plot the graph for agents for decreasing, increasing, and maintaining the temperature for DQN & DDQN together. (0.5)
- Conclude your assignment with your analysis consisting of at least 200 words by summarizing your findings for agent's behaviour using Actor-Critic, DQN and DDQN techniques for optimizing the energy consumption. (1 M)