BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES DIVISION

Deep Reinforcement Learning
Lab Assignment 2

Total Marks = 13 Marks

Intended Learning Outcome:

Students should be able to

- Understand the basic functionality of Deep Q Networks (DQN) and Actor-Critic.
- Implement the concepts of Deep Q Networks (DQN) and Actor-Critic.

Prerequisite:

(1) Students should go through the lectures CS6, CS7, CS8, CS9, CS10, CS11, CS12, CS13;
(2) Webinar demonstrations

Please note that this assignment will involve some amount of self-learning (on the part of modelling solutions appropriately + programming skills)

Submission Deadline: 13th March, 2025

Instructions:

- Read the assignment proposal carefully.
- Solve **any one** assignment problem. Submit solution file as. (Team # - TrafficOptimization OR Team # - CyberAttack)
- It is mandatory to **submit** the assignment in **PDF format only** consisting of all the outcomes with each and every iteration. Any other format will not be accepted.
- Add comments and descriptions 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.

How to reach out for any clarifications:

This assignment is administered by

(1) Pooja Harde - pooja.harde@wilp.bits-pilani.ac.in
(2) Divya K - divyak@wilp.bits-pilani.ac.in
(3) Dincy R Arikkat - dincyrarikkat@wilp.bits-pilani.ac.in

Any request for clarification must be addressed through email (official email only) to all the three instructors listed.

Messaging in TEAMS is discouraged. This is to ensure we maintain track of all the transactions. If we find any clarifications to be shared across all the students, we will share this using discussion forums.

-

PROBLEM STATEMENT - #1

Cyber-Attack Simulation Using Deep Reinforcement Learning

Reference Paper: electronics-13-00555.pdf

1. Problem Statement:

Develop a Deep Reinforcement Learning (DRL) agent to simulate cyber-attacks in a controlled environment based on the MITRE ATT&CK framework. The agent must learn optimal attack strategies to infiltrate a simulated network while overcoming security measures and maximizing penetration success.

2. Objective:

The agent must learn the optimal attack policy π* using Deep Reinforcement Learning to maximize cumulative rewards while navigating and attacking the simulated network.

3. Environment Details:

a. **State Space:** The state of the environment at any given time is represented by a vector (or a dictionary in code) that characterizes each target PC. The following attributes are considered *(Ref. Table 1 Pg. No. 6 of the paper)*:

1. Num_Ports (Integer): Number of open ports on the target PC. Typical range would be 0-10 ports.
2. Is_Admin_Compromised (Binary): 0 if the agent doesn't have admin access, 1 if the agent does.
3. Keyboard_Security_Enabled (Binary): 0 if keyboard security (e.g., preventing keylogging) is disabled, 1 if it's enabled.
4. Web_Creds_Present (Binary): 0 if no stored web credentials are found, 1 if they are present.

b. **Action Space:** The agent can choose from the following discrete actions *(Ref. Table 2 Pg. No. 7 of the paper)*:

0: "Open Port Attack" (attempt to exploit vulnerabilities through open ports)

1: "Spoofing" (attempt to impersonate an authorized PC via IP address spoofing)
2: "Keylogging" (attempt to capture credentials via keylogging, only effective if Keyboard_Security_Enabled is 0)
3: "Access Web Credential" (attempt to extract credentials stored in web browsers, only effective if Web_Creds_Present is 1)

c. **Rewards** *(Ref. Pg. No. 7, Section 3.2.3 of the paper)***:**
+1.0: If the action successfully acquires administrative privileges on a target PC that wasn't previously compromised.
-1.0: If the action fails to acquire credentials or is invalid in the current state (e.g., attempting keylogging when Keyboard_Security_Enabled is 1).

Synthetic Dataset Creation:
To make the agent learn on the cyber-attack simulated network environment, it is required to create a synthetic dataset containing of all the possible information about the network and possible actions to perform along with the reward and next_state. The agent's objective is to launch an attack on the PCs within the network, which consists of five systems labeled as PC1, PC2, PC3, PC4, and PC5. These PCs will be considered as the target PCs *(Ref. Pg. No. 8 of the paper)*.

The dataset will contain **6 columns**: target PC, current_state, action, reward, next_state, and done. Each row in the dataset can be seen as a tuple : `(state, action, reward, next_state, done)` for every target PC.

The **state column contains a vector of 4 values** with each value randomly generated: [Num_Ports, Is_Admin_Compromised, Keyboard_Security_Enabled, Web_Creds_Present]. The possible values for each state value are discussed in the above state space section.

Deciding the reward for each state and action value can be generated as follows:
Probability for the success of each action:

a. **Open Port Attack:** This action is only possible if the target PC has any open ports. If open ports exist, there's a 40% chance of success, resulting in administrative compromise and a reward of +1.0. Otherwise, it fails with a reward of -1.0. If there are no open ports to begin with, this action automatically fails, earning the agent a reward of -1.0.
b. **Spoofing:** This action cannot be performed if the target PC is already compromised. Otherwise, there is a 10% chance of success, resulting in administrative compromise and a reward of +1.0. On a failure, the reward is -1.0.
c. **Keylogging:** Keylogging will only work if Keyboard Security is disabled on the target PC. If keyboard security is disabled, the agent has an 80% chance of success, achieving administrative compromise and a reward of +1.0. This attempt

automatically fails if keyboard security is enabled, earning the agent a reward of -1.0.

d. **Access Web Credentials:** Attempting to access web credentials is only possible if such credentials exist on the target PC. If credentials exist, the agent has a 60% chance of success, achieving administrative compromise and earning a reward of +1.0. This attempt automatically fails if web credentials are not present, earning the agent a reward of -1.0.

Sample Dataset Structure **(Create a dataset of 2000 rows)**:

target PC	current_state	action	reward	next_state	done
PC2	[2, 0, 1, 1]	keylogging	-1.0	[2, 0, 1, 1]	False
PC2	[2, 1, 1, 1]	keylogging	-1.0	[2, 1, 1, 1]	False
PC2	[2, 0, 0, 1]	keylogging	1.0	[2, 1, 0, 1]	False

Example scenario for creating a dataset:

Input: (target PC, state, action)

target PC	current_state	action	reward	next_state	done
PC2	[2, 0, 1, 1]	keylogging			

Target PC: PC2
State Vector (Randomly Generated): [2, 0, 1, 1] (Num_Ports = 2
Is_Admin_Compromised = 0 (Not yet compromised)
Keyboard_Security_Enabled = 1 (Keyboard security is enabled)
Web_Creds_Present = 1 (Web credentials are present))

Action (Randomly Generated): Action 2 = "Keylogging"

Then follow the below logic for deciding the reward:

```
if action == 2: #Keylogging
  if state[target_pc]["is_admin_compromised"] == 1:
    reward = -1.0

  elif state[target_pc]["keyboard_security_enabled"] == 0: #security enabled
    if random_number < 0.8:  #Probability condition
      new_state[target_pc]["is_admin_compromised"] = 1
      reward = 1.0
    else:
        reward = -1.0 #Returns back
    else:
        reward = -1.0
```

Explanation: The motive of this action by agent is to compromise the target PC. If the PC is already comparomised and the agent is still selecting an action of “Keylogging” then the reward will be -1.0. Otherwise if the PC is not compromised and “Keylogging” is disabled, then there is a possibility of 0.8 that the agent is successful in “Keylogging” and the compromised_PC flag is set to 1. Else if the agent is not successful then the reward will be -1.0.

Hence, next_state will be: [2, 0, 1, 1] -> unchanged

Output: (reward, next state, done)

Target PC	state	action	reward	next_state	done
PC2	[2, 0, 1, 1]	keylogging	-1.0	[2, 0, 1, 1]	False

Stopping condition for Agent:

1. Done = True (when all the PCs are compromised then the program will get terminated)
   OR
2. Maximum iterations reached (e.g. max_iteration = 3000)

Implementation:

1. Create a synthetic dataset for the cyber-attack simulation task as mentioned in the Dataset Sample. (1 Mark)
2. You are required to implement DQN and Actor-Critic on the developed dataset.
3. Create a CyberAttack Environment. (0.5 Marks)
4. DQN algorithm:
   a. Parameters: (1 Mark)
      i. Number of episodes
      ii. Max capacity of replay memory
      iii. Batch size
      iv. Period of Q target network updates
      v. Discount factor for future rewards
      vi. The initial value for epsilon of the e-greedy
      vii. Final value for epsilon of the e-greedy
      viii. Learning rate of ADAM optimizer, and etc..
   b. Implement a replay buffer for storing the experiences. (0.5 Marks)
   c. Design the Main Network (0.5 Marks)
   d. Target Network (0.5 Marks)
   e. Training Implementation for DQN (1 Mark)
   f. Print all the iterations (1 Mark)

5. Actor-Critic algorithm:
    a. Parameters: (1 Mark)
        i. Number of episodes
        ii. Batch size
        iii. Learning Rate
        iv. Optimizer, etc.
    b. Design the Actor Network (0.5 Marks)
    c. Design the Critic Network (0.5 Marks)
    d. Training Implementation for Actor-Critic (1 Mark)
    e. Print all the iterations (1 Mark)
6. Plot the graph for Average Reward Obtained by all RL Algorithms. (1 Mark)
7. Plot the graph for Average Success Rate Obtained by all RL Algorithms. (1 Mark)
8. Conclude your assignment with your analysis consisting of at least 200 words by summarizing your findings of the assignment. (1 Mark)

Colab Template: Cyber-Attack Simulation.ipynb (*Make a copy of the template. Do not send the edit access request.*)

PROBLEM STATEMENT - #2

Adaptive Traffic Flow Optimization

Objective:

Develop and compare Reinforcement Learning agents (**DQN, and Actor-Critic**) to optimize traffic flow and vehicle speed regulation. Implement RL-based strategies to enhance traffic efficiency, reduce congestion, and improve safety by adaptively controlling the speed and lane changes of a selected vehicle within the simulated environment.

Dataset:

https://drive.google.com/file/d/1pyExKzpKVRhFr2Ltfp6Ts8yF8OdWZNgH/view?usp=drive_link

- **Time Step:** The dataset provides vehicle trajectory data at a frequency of **10 Hz**, meaning each frame represents a **0.1-second interval**.

State Space :

The **state** represents the **current traffic conditions and vehicle status**:

1. **Vehicle Speed (v_Vel)** (m/s)
2. **Vehicle Acceleration (v_Acc)** (m/s^2)
3. **Lane Position (Lane_ID})**
4. **Distance to Preceding Vehicle (Space_Headway)** (m)
5. **Time Gap to Preceding Vehicle (Time_Headway)** (s)
6. **Vehicle Class (v_Class)**
7. **Global X (Global_X)**
8. **Global Y (Global_Y)**

Total State Vector Dimension: 8 features

Action Space :

Action	Description	Conditions	Change Applied
0	Maintain current speed	No change required	0 m/s adjustment
1	Increase speed	If Space_Headway≥15m	+2 m/s
2	Decrease speed	If Space_Headway<10m	−2 m/s
3	Change to left lane	If **left lane exists and is not occupied** and Space_Headway≥15m	Move left
4	Change to right lane	If **right lane exists and is not occupied** and Space_Headway≥15m	Move right

---

Traffic Safety and Target Speed:

- **Safe Following Distance: At least 15 meters** from the preceding vehicle (Space_Headway≥15m).
- **Collision Risk: Less than 5 meters** gap is **unsafe** (Space_Headway<5m).
- **Optimal Target Speed: 27 m/s** (approximately **60 mph**, highway recommended speed).

Reward Function:

$$R = (10 - |V_t - V_{optimal}|) - P_{collision}$$

Where:

- V_t = Current vehicle speed (m/s).
- $V_{optimal}$ = 27 m/s (highway optimal speed)
- $P_{collision}$ = {

  20 if SpaceHeadway < 5m (high collision risk)

  0 otherwise

  }

Requirements and Deliverables:

1. Load and Preprocess data. **(1 Mark)**
   a. Convert relevant columns (e.g., v_Vel, v_Acc, Space_Headway, Time_Headway, etc.) to numerical data types.
   b. Normalize or standardize the numerical data to improve training stability.
2. Develop the Traffic Control Environment. **(0.5 Mark)**
3. Define Action Functions: MaintainSpeed, IncreaseSpeed, DecreaseSpeed, ChangeLaneLeft, ChangeLaneRight. **(2.5 Mark)**
4. Implement the Reward Function **(1 Mark)**
5. Implement a Replay Buffer for experience storage in DQN. **(1 Mark)**
6. Design and Train DQN: **(2.5 Mark)**
   a. Neural Network Structure.
   b. Model Training.
7. Design and train Actor-Critic Algorithm **(2.5 Mark)**
8. Plot the graph for Average Reward Obtained by DQN and actor critic. **(1 Mark)**
9. Compare and summarize Traffic Flow Outcomes for DQN and Actor Critic consisting of at least 200 words. **(1 Mark)**

Colab Template: Traffic Flow Optimization.ipynb (*Make a copy of the template. Do not send the edit access request.*)