Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Human-Centered Interaction in Robotics

HCIR Assignment-4

Shrikar Nakhye
Trushar Ghanekar
Keerthan Shekarappa
Mohammed Ibrahim Memon

May 2024

Contents

1

TASK 1 - Multimodal Machine Learning

1.1 Think of a human-robot interaction scenario. Describe the scenario and argue where single modalities are sufficient and where multiple modalities are needed.

Scenario: Hospital Service Robot which is assiting nurses to deliver first aid medics

1.1.1 Single Modalities:

Navigation

- Vision - it is sufficient for navigation. Robot can use cameras and imagery recognition algorithms to navigate to the hospitals and generate pathways and follow the markers if provided by the deployer of the robot. It is very important to note that vision based navigation is only reliable in proper lit environments.

Delivery

- Audiotory - while delivering any medicine to a patient, robot can use speech recognition and synthesis to announce its use, strength, type and frequency. Audio announcements are effective as they do not require any visual attention from the patient.

1.1.2 Multiple Modalities:

Patient Interaction

- Vision + Auditory + Touch - interacting with patients may require multiple modalities to ensure effective and comfortable communication.
  - Vision - Recognition of faces and identification of their emotion
  - Auditory - communication in verbal sense and answering basic questions
  - Touch - gently give a haptic feedback to a patient as a affirmation or comforting gesture

1.2 Explain any two challenges in the field of multimodal machine learning with the help of examples. Describe the categories of approaches that have been developed to deal with these challenges.

1.2.0 Challenge 1: Data Fusion

Description: Data fusion integrates information from multiple modalities to make decisions or predictions. Each modality has different properties, noise levels, and temporal characteristics, making effective combination challenging.

Example: In healthcare, a multimodal system integrates data from MRI scans (high-dimensional spatial data), patient history (unstructured text), and ECG signals (time-series data).

Approaches:

- **Feature-level Fusion:**
  - **Description:** Combine features from different modalities into a single vector before model input.
  - **Example:** Concatenate features from MRI, text, and ECG data for a classifier.
  - **Advantages:** Simplifies the model and leverages traditional ML techniques.
  - **Challenges:** May suffer from the curse of dimensionality; requires careful preprocessing.
- **Decision-level Fusion:**
  - **Description:** Independently process each modality and combine predictions later.
  - **Example:** Separate models analyze MRI, text, and ECG, with combined predictions using voting or averaging.
  - **Advantages:** Allows modality-specific preprocessing and modeling.
  - **Challenges:** Requires effective decision combination methods; may lose inter-modality interactions.
- **Hybrid Fusion:**
  - **Description:** Combines feature-level and decision-level fusion techniques.
  - **Example:** Combine features from MRI and ECG in a joint model, with text data processed separately.
  - **Advantages:** Balances strengths of both fusion approaches.
  - **Challenges:** Increased complexity and computational cost.

1.2.0 Challenge 2: Handling Missing Data

Description: Multimodal systems often encounter missing data due to sensor failure, occlusions, or data corruption. Handling incomplete data without degrading performance is a significant challenge.

Example: In autonomous driving, if a camera is obscured by fog, the system must rely on LiDAR and radar data to make decisions.

Approaches:

- **Data Imputation:**
  - **Description:** Fill missing data using statistical methods or ML models.
  - **Example:** Estimate missing camera data based on LiDAR and radar using interpolation or deep learning.
  - **Advantages:** Maintains a complete dataset for training and inference.
  - **Challenges:** Imputed data may introduce biases and inaccuracies.
- **Modality-Agnostic Models:**
  - **Description:** Develop models that operate with varying input modalities, adjusting dynamically.
  - **Example:** Autonomous driving models weighing data sources based on reliability.
  - **Advantages:** Robust to missing data and adaptable to scenarios.
  - **Challenges:** Complex design requiring careful balancing of modality contributions.
- **Ensemble Methods:**
  - **Description:** Use an ensemble of models trained on different modality subsets, combining outputs.
  - **Example:** Combine models trained on camera, LiDAR, and radar data to handle missing modalities.
  - **Advantages:** Increases robustness and handles missing data gracefully.
  - **Challenges:** Higher computational cost and complexity in managing multiple models.

1.2 Conclusion

Multimodal machine learning faces significant challenges in data fusion and handling missing data. Strategies such as feature-level fusion, decision-level fusion, hybrid fusion, data imputation, modality-agnostic models, and ensemble methods help improve robustness and accuracy in various applications.

2

TASK 2 - Human-Centered Reinforcement Learning

2.1 Name at least two characteristics of each of the three algorithms: TAMER, SABL and COACH.

2.1.0 TAMER

- Human feedback is mapped to a numeric value, allowing the agent to learn from the reward signals provided by humans directly
- Credit assignment is a key feature, enabling the agent to attribute feedback to specific actions, improving learning efficiency

2.1.0 SABL

- Human feedback is mapped to categorical strategies, such as positive reward, negative reward, positive punishment, and negative punishment
- Bayesian inference is used to update the policy based on the categorical probability of the given human feedback.

2.1.0 COACH

- Human feedback is mapped directly to the agent's policy, meaning the feedback influences the selection of actions
- The feedback given for correct actions decreases progressively as the agent learns, reflecting the human trainer's strategy of reducing feedback over time

2.2 For each of the three algorithms, give an example of a human-robot interaction scenario where you would apply this algorithm

2.2.0 TAMER (Training an Agent Manually via Evaluative Reinforcement)

Scenario: Teaching a Domestic Robot to Clean a Room

Description: A user can provide positive and negative feedback to a domestic robot as it learns to clean a room. For instance, the robot might receive positive feedback when it picks up trash and negative feedback when it misses a spot.

Why TAMER? TAMER is designed to leverage human-generated reward signals to shape the robot's behavior directly. This allows the robot to quickly learn the preferences and requirements of the user through simple and intuitive feedback mechanisms.

2.2.0 SABL (Strategy-Aware Bayesian Learning)

Scenario: Training a Robot to Assist in Assembly Line Tasks

Description: On an assembly line, a worker can give categorical feedback (e.g., correct, incorrect, needs improvement) to a robot helping with assembly tasks. The feedback can be strategy-aware, meaning the robot can recognize patterns in feedback over time and adjust its actions accordingly.

Why SABL? SABL uses categorical feedback to inform the robot's learning process. This is particularly useful in structured environments like assembly lines where feedback can be systematically categorized and used to improve the robot's performance in a more nuanced way than simple reward or punishment.

2.2.0 COACH (Convergent Actor-Critic by Humans)

Scenario: Teaching a Service Robot to Interact with Customers

Description: In a customer service setting, a human supervisor can give feedback on the actions of a service robot interacting with customers. For example, the supervisor can indicate whether the robot's responses were appropriate, polite, and helpful.

Why COACH? COACH incorporates human feedback directly into the policy update mechanism of the robot. This makes it effective for scenarios requiring a high degree of social interaction and adaptability, as the robot can quickly learn from the supervisor's guidance to improve its interactions with customers.

3

Task 3

3.1 Modify the code base so that the program can take human evaluative feedback via keyboard with keys'W' or 'w' for reward (+1), 'A' or 'a' for punishment (-1), while the agent is training. (5%)

```
def read_feedback(self):
    """
    Get human input. 'W' key for positive, 'A' key for negative.
    Returns: scalar reward (1 for positive, -1 for negative)
    """
    reward = 0
    self.lock.acquire()
    area = self.screen.fill((128, 128, 128),
            (0, 0, self.screen.get_width(), 200))
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_w:
                reward = 1
            elif event.key == pygame.K_a:
                reward = -1

    pygame.display.update(area)
    self.reward = reward
    self.lock.release()
```

3.2 Train the mountain car to reach the goal state when credit assignment is set to ‘False’ and share a screenshot of how far your car reached. (15%)

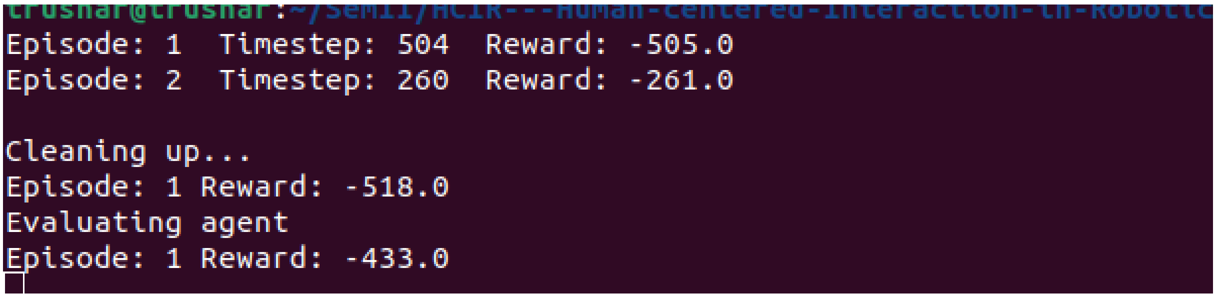

Figure 3.1: Screenshot of terminal when CA was assigned ’False’

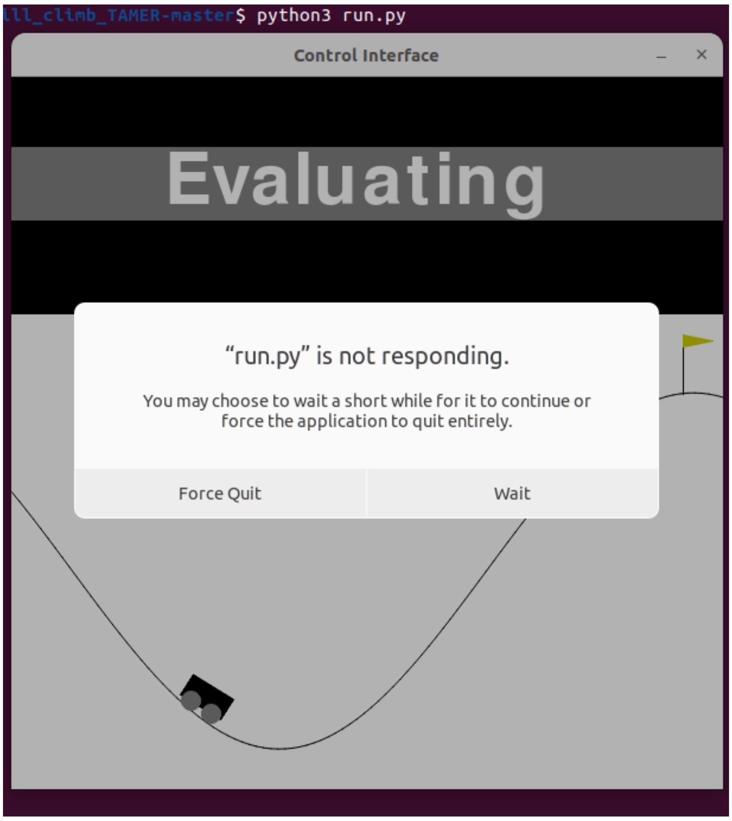

Figure 3.2: Run.py file was not responding several time while Evaluating

3.3 Explain the strategy you used for giving rewards and punishments while training the mountain car. (5%)

Technique for Applying Penalties and Rewards:

- **Rewards**
  - **Uphill Movement:** A reward (W or w) is awarded when the car moves uphill in the direction of the goal. This motivates the agent to carry out other acts that lead to mountain climbing.
  - **Speed Increase:** As the car accelerates, particularly toward the objective, a reward is given to encourage the momentum-building acts that are essential to reaching the top.
- **Punishments**
  - **Sliding Back:** A penalty (A or a) is applied when the vehicle starts to slide back down the hill. This teaches the agent to stay away from acts that cause them to regress.
  - **Stalling:** To promote exploration and the development of more effective climbing techniques, the car is punished if it is not moving forward significantly, such as when it becomes stalled at the base of a hill.

3.4 Repeat steps 'b' and 'c' with credit assignment now set to 'True'. (20%)

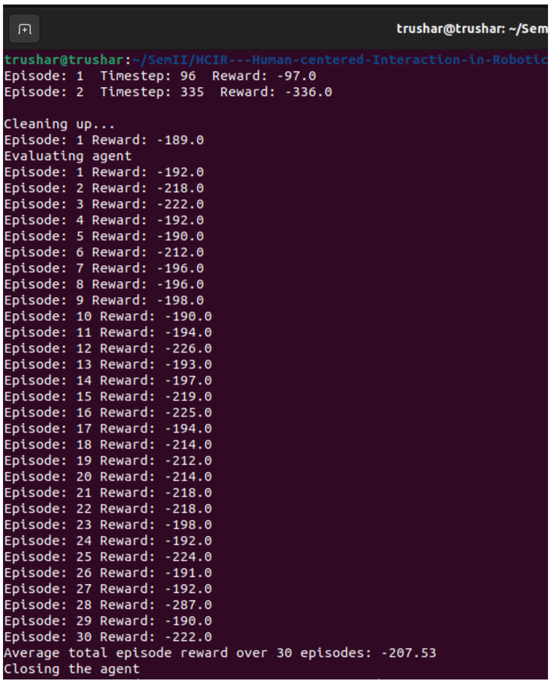

Figure 3.3: Screenshot of terminal when CA was assigned 'True'

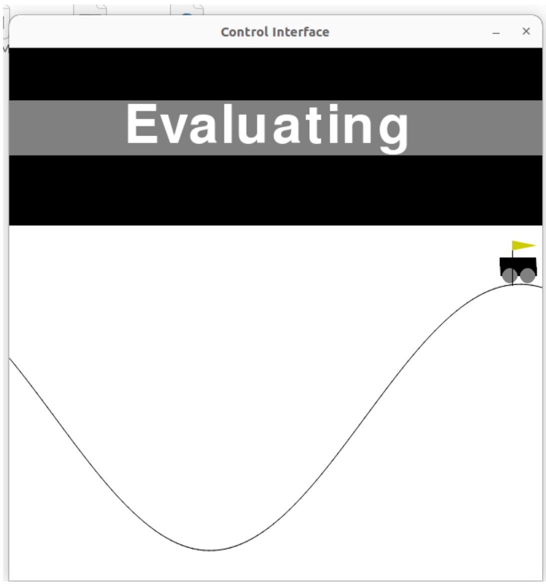

Figure 3.4: The car reached the goal

3.5 Compare your results and reason about the role of credit assignment. (5%)

- **Without Credit Assignment:**
  - **Results:** The car showed some learning and managed to climb parts of the hill, but it often got stuck or slid back frequently. The overall progress was slower, and it required more feedback to make incremental improvements.
  - **Reasoning:** Without credit assignment, the agent lacks the ability to properly attribute delayed effects of actions.
- **With Credit Assignment:**
  - **Results:** The car reached the goal state more consistently and quickly. The learning curve was steeper, showing more rapid improvement in performance.
  - **Reasoning:** Credit assignment allows the agent to understand the delayed consequences of its actions. This means that even if a particular action's reward or punishment is not immediate, the agent can learn the value of sequences of actions that lead to a successful outcome. This enhances the agent's ability to develop effective strategies and understand the long-term benefits of certain actions.

All things considered, credit assignment is essential in assisting the agent in gaining a greater understanding of the surroundings and the long-term effects of its activities, which improves performance and speeds up learning.