AML Assignment – 3
Aishvarya S
MDS202302

Deep Q-Network (DQN) Training on Acrobot-v1

Objective

The goal was to implement a Deep Q-Network (DQN) to train an agent to solve the "Acrobot-v1" environment from OpenAI's Gymnasium. The objective of the environment is to apply torques to swing the lower link of a double pendulum above a specified threshold in the shortest number of steps.

Methodology

The DQN algorithm employs a neural network to approximate the Q-values for state-action pairs. Two networks were utilized: the **policy network**, which learns during training, and the **target network**, which provides stable targets for the loss function. Key components of the implementation include:

1. Neural Network Architecture: A fully connected feedforward network with two hidden layers, each containing 256 units with ReLU activations.
2. Replay Memory: A buffer that stores transitions (state, action, reward, next state, done). A batch of 64 transitions was sampled randomly for training to decorrelate data.
3. Epsilon-Greedy Exploration: The agent initially selects actions randomly (high exploration) and transitions to a more greedy policy (low exploration) as training progresses, using a decaying epsilon schedule.
4. Reward Shaping: Rewards were adjusted by adding a scaled angular velocity term to encourage faster progress toward the goal.
5. Training Loop: The agent was trained for 1,000 episodes, updating the target network every 10 episodes and decaying epsilon over 1,000 steps. The performance was evaluated by the cumulative reward achieved in each episode.

Results

The trained agent was evaluated against a random agent to assess performance. The results are as follows:

- Trained Agent:
  - Average Reward: -201.91
  - Best Episode Reward: -107.37903186690528

- Random Agent:
  - Average Reward: -499.06

The significant improvement in the trained agent's reward demonstrates the efficacy of the DQN approach compared to random agent.

Visual Analysis

To analyse the behaviour of the trained agent, the best episode was replayed, and an animation of the episode was generated. The agent effectively achieved the goal by strategically applying torques to swing the pendulum above the threshold.

Observations and Challenges

- Epsilon Decay: Balancing exploration and exploitation required careful tuning of the epsilon decay schedule to avoid premature convergence to suboptimal policies.
- Reward Shaping: Modifying the reward signal was critical in accelerating convergence, as the original reward system provides sparse feedback.
- Training Stability: Occasional instability was observed, likely due to the non-stationary nature of the Q-value targets.