

# LunarLander using Actor-Critic Method

Professor Mudassar Liaq

Session Fall 2024

**Deadline 11:50pm on 3rd August 2025**

## 1 Objective

The goal of this assignment is to implement and understand the Actor-Critic reinforcement learning algorithm to solve the LunarLander-v3 environment from OpenAI Gymnasium.

## 2 Implementation

### 2.1 Setup and Environment Exploration

- Install the necessary libraries: gymnasium, numpy, pytorch on Jupyter notebook.
- Explore the LunarLander-v3 environment to understand the state and action spaces, and reward structure as shown in Fig. 1.

### 2.2 Design the Actor and Critic Networks

- Implement two separate neural networks: one for the actor (policy network) and one for the critic (value network).
- Define the architecture of these networks (input layer, hidden layers, and output layer).

### 2.3 Policy and Value Estimation

- Implement the forward pass for both actor and critic networks.
- Ensure the actor network outputs a probability distribution over actions.
- Ensure the critic network outputs a scalar value representing the state value.

### 2.4 Training Loop

- Implement the training loop for the Actor-Critic algorithm.
- At each step, interact with the environment to obtain the state, action, reward, and next state.
- Compute the advantage estimate using the critic's value function.
- Update the actor network using the policy gradient method.
- Update the critic network by minimizing the mean squared error of the value estimates.

### 2.5 Evaluation and Testing

- Evaluate the trained model by running several episodes and recording the performance.
- Plot the learning curve to show the improvement in cumulative rewards over time.
- Compare the performance with a baseline random policy.

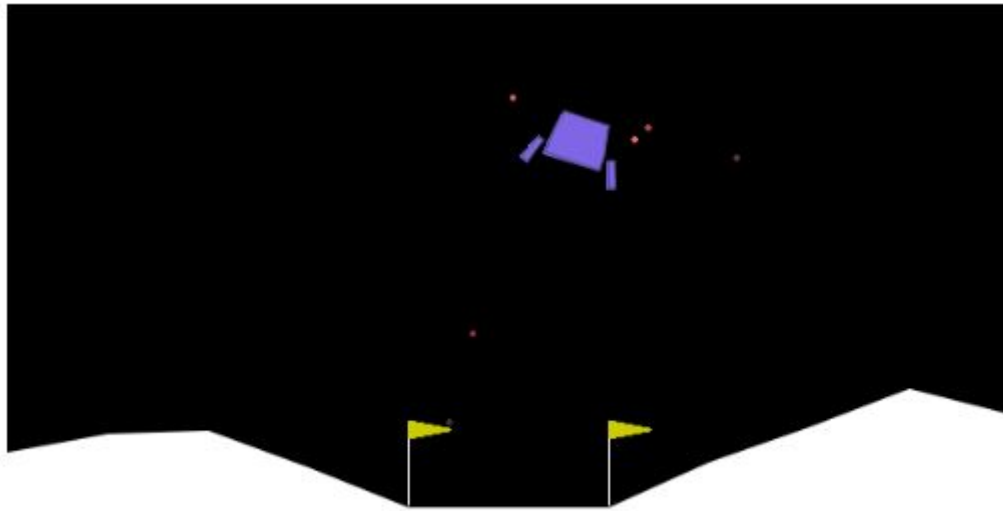


Figure 1: LunarLander-v3 environment from OpenAI Gymnasium.

### 3 Submission

- Write a report detailing your implementation, including the architecture of your networks, training process, and results.
- Discuss any challenges faced during the implementation and how you overcame them.
- Provide suggestions for further improvement or exploration.
- You should submit a single PDF document containing your report and a single archive containing all your source code.
- Please organize your source code so that it is easy to follow and apparent how to run your solutions to the assignment's questions.
- You have to work on this assignment using jupyter notebooks. Any work done on Google Collab will be given 0 credit.
- You strictly have to use forward view implementation of these algorithms. Any implementations of the backward view (i.e. with eligibility traces) will result in 0 credits.