Reinforcement Learning Mini-Project

Machine Learning Course

1 Introduction

This project will give you hands-on experience with Reinforcement Learning (RL) fundamentals by working with the **Panda-Gym** environment. You'll learn how to train an intelligent agent to control a robotic arm in a realistic simulation.

Panda-Gym is a collection of tasks based on the PyBullet simulator, offering a variety of robotic control challenges. We will focus on object manipulation tasks such as reaching, pushing, and pick-and-place.

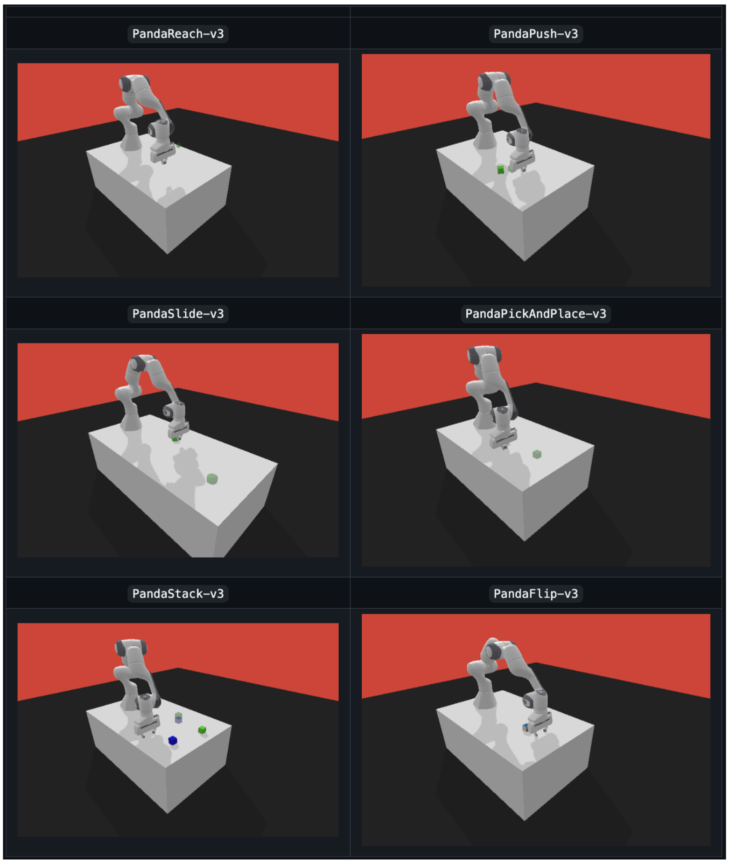

Figure 1: Environment visualization

2 Project Objectives

- Understand the RL paradigm: agent, environment, states, actions, rewards.
- Implement and train basic RL agents using standard algorithms.
- Experiment with different environments and observe policy behavior.
- Analyze the learning process using reward curves and policy visualization.

3 Setup Instructions

1. Install Anaconda or create a virtual environment.
2. Clone the repository:

```
git clone https://github.com/qgallouedec/panda-gym.git
cd panda-gym
```

3. Install dependencies:

```
pip install -r requirements.txt
pip install panda-gym
pip install stable-baselines3[extra]
```

4. Refer to this example notebook to understand how to interact with the environment: PickAndPlace.ipynb Example

4 Project Tasks

4.1 Task 1: Explore the Panda-Reach Environment

Goal: Move the robot's end-effector to a target position.

Subtasks:

- Run the `PandaReach-v3` environment.
- Observe how actions affect the robot arm.
- Modify rendering and step limits.

Deliverables:

- Screenshots of successful reach episodes.
- Brief explanation of the observation and action space.

4.2 Task 2: Train a DDPG Agent

Goal: Use the DDPG algorithm to train an agent to complete the reaching task.

Subtasks:

- Implement the training loop using Stable-Baselines3.
- Save and load the trained model.
- Plot training reward over episodes.

Deliverables:

- Python script or notebook for training.
- A graph showing reward convergence.

4.3 Task 3: Extend to PandaPush-v3

Goal: Move an object to a goal location using the robotic arm.

Subtasks:

- Replace the environment with `PandaPush-v3`.
- Re-train your agent from scratch.
- Tune hyperparameters to improve performance.

Deliverables:

- Reward graph comparison with PandaReach.
- Notes on difficulty differences and agent behavior.

4.4 Task 4 (Bonus): Evaluate with Success Metrics

Goal: Implement success criteria and log performance.

Subtasks:

- Use a success threshold to define successful episodes.
- Track and visualize success rate over time.

Deliverables:

- Success rate plots.
- Summary analysis of learning stability.

5 Report Guidelines

Your final submission should include:

- A brief project introduction.
- Descriptions of all tasks performed.
- Code snippets or notebooks.
- Plots (training reward, success rate).
- Conclusion and important facts about what you learned.

6 Resources

- Panda-Gym GitHub Repo
- Stable-Baselines3 Documentation
- OpenAI Gym: `https://gym.openai.com`
- PyBullet: `https://pybullet.org`

7 Grading Rubric

- Environment Setup and Task Execution: 25%
- Agent Training and Results: 30%
- Report Quality and Presentation: 30%
- Bonus Task (if done): 10%
- Code Readability and Comments: 15%