

ME5406 Deep Learning for Robotics

Project 1: The Frozen Lake Problem and Variations

- **Name:** Luo Zhonghao
- **Student ID:** A0308832A
- **Email:** E1404383@u.nus.edu
- **Course:** ME5406 Deep Learning for Robotics

Table of Contents

- [Project Description](#)
- [Package Dependencies](#)
- [Project Structure](#)
- [How to Run](#)

Project Description

This project aims to implement and evaluate three model-free reinforcement learning algorithms, First-visit Monte Carlo, SARSA, and Q-learning, to solve the classic Frozen Lake problem as well as its variations.

Package Dependencies

- Python 3.6.13
- Required Packages (install via `pip install -r requirements.txt`):

```
numpy==1.19.5
matplotlib==3.3.4
jupyter==1.0.0
notebook==6.4.12
```

Project Structure

```
Project_Folder/
|
├─ project1new.ipynb      # All code in Jupyter Notebook
├─ task1.1.py             # Monte Carlo implementation (4x4 grid)
├─ task1.2.py             # SARSA implementation (4x4 grid)
├─ task1.3.py             # Q-Learning implementation (4x4 grid)
├─ task2.1.py             # Monte Carlo implementation (10x10 grid)
├─ task2.2.py             # SARSA implementation (10x10 grid)
├─ task2.3.py             # Q-Learning implementation (10x10 grid)
├─ requirements.txt       # Required packages list
└─ README.md              # Project documentation
```

How to Run

1. Create a Virtual Environment (Optional)

It is recommended to create a virtual environment to avoid conflicts with system packages:

```
python3 -m venv venv
source venv/bin/activate # Linux/Mac
venv\Scripts\activate   # Windows
```

2. Install Required Packages

Ensure Python 3.6+ is installed along with the necessary scientific computing and reinforcement learning libraries.

```
pip install -r requirements.txt
```

3.1. Run in Jupyter Notebook

1. Ensure Jupyter Notebook is installed:

```
pip install notebook
```

2. Launch Jupyter Notebook:

```
jupyter notebook project1.ipynb
```

3. Execute each cell in the notebook to observe the implementation and results.

3.2. Run via Command Line

1. To execute specific tasks individually, run the following commands:

```
python task1.1.py # Monte Carlo implementation (4x4 grid)
python task1.2.py # SARSA implementation (4x4 grid)
python task1.3.py # Q-Learning implementation (4x4 grid)
python task2.1.py # Monte Carlo implementation (10x10 grid)
python task2.2.py # SARSA implementation (10x10 grid)
python task2.3.py # Q-Learning implementation (10x10 grid)
```