

## ME5406 Deep Learning for Robotics

## Project for Part I: The Froze Lake Problem and Variations

## Project Description and Requirement

Dr. Peter C. Y. Chen

Associate Professor

Department of Mechanical Engineering

National University of Singapore

Email: mpechenp@nus.edu.sg

Project report and code due on 18 October 2020, 11:59 PM, Singapore time

## I. OBJECTIVE

This project is designed for the student to demonstrate (through independent learning):

1. Competence in implementing a set of model-free reinforcement learning techniques in a small scale problem setting, and
2. Understanding of the principles of, and implementation issues related to, this set of techniques.
2. The robot is confined within the grid.
3. The robot receives a reward of (i)  $+1$  if it reaches the frisbee, (ii)  $-1$  if it falls into a hole, and (iii)  $0$  for all other cases.
4. An episode ends when the robot reaches the frisbee or falls into a hole.

## II. PROBLEM STATEMENT

Consider a frozen lake with (four) holes covered by patches of very thin ice. Suppose that a robot is to glide on the frozen surface from one location (i.e., the top left corner) to another (bottom right corner) in order to pick up a frisbee, as is illustrated in Figure 1.

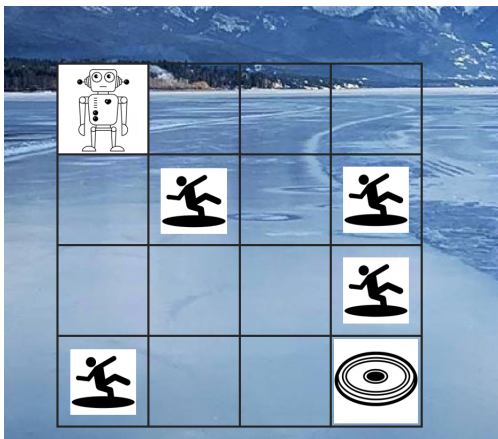


Fig. 1: A robot moving on a frozen lake.

The operation of the robot has the following characteristics:

1. At a state, the robot can move in one of four directions, left, right, up, and down.

## III. REQUIREMENT

## A. What to be done

Three tasks as described below are to be completed for this project. The percentage associated with each task indicates the mark weightage.

Task 1: Basic implementation (25%)

Write a Python program to compute an optimal policy for the Frozen Lak problem as described in Section II, using the following three tabular (i.e., not involving any use of a neural network) reinforcement learning techniques:

1. First-visit Monte Carlo control without exploring starts.
2. SARSA with an  $\epsilon$ -greedy behavior policy.
3.  $Q$ -learning with an  $\epsilon$ -greedy behavior policy.

You can set the values for all the necessary parameters, such as discount rate, learning rate, etc.

Task 2: Extended implementation (25%)

Increase the grid size to at least  $10 \times 10$  while maintaining the same proportion between the number of holes and the number of states (i.e.,  $4/16 = 25\%$ ). Distribute the holes randomly with-

out completely blocking access to the frisbee. Repeat Task 1.

### Task 3: Report (50%)

Write an individual report that describes the implementation and discusses the results. This report should be no more than 10 pages (excluding the cover page). Compare and contrast the performance of the three reinforcement learning techniques and the results that they have generated in your implementations. Discuss the difficulties encountered and describe how they were overcome during the project. Elaborate on your own initiatives (if any) to investigate and improve the efficiency of these techniques in solving the given problem.

#### B. Python programming

For setting up the “frozen lake environment”, you can use publicly available toolkits (such as OpenAI gym) or write the code yourself. The advantage of the latter option is that you will learn how to implement the “low-level” features of a reinforcement learning problem.

Your Python code must be able to run under Python 3.6, either as a Jupyter Notebook or in plain Python code, and use only standard and publicly available packages. For programming, the PyCharm integrated development environment is recommended.

Coding convention is to be observed. In particular, clear and concise comments should be included in the source code to explain various calculation steps, e.g., how the number of first visits to a state-action pair is computed, and how an exploratory action in SARSA and  $Q$ -learning is selected, etc. The explanation should be detailed and specific; brief and general comments such as “*These lines compute the value for [something]*” are not adequate.

#### C. What to submit

1. An individual report in a PDF file. The name of the PDF file must be in the format:  
 StudentNumber\_ME5406\_Project1.pdf  
 The report must contain a cover page showing:
  - a) student name
  - b) student number
  - c) student email address
  - d) name of module
  - e) project title
2. The Python code in either plain text or as a Jupyter Notebook.

#### D. How to submit

Only softcopy of the report (in PDF) and the Python code are to be submitted. Please put the report and the code file in a folder. Use your student number as the folder name. Generate a non-password-protected zipfile of this folder and upload this zipfile onto LumiNUS at:

Module: ME5406 Deep Learning for Robotics  
 [2010] 2020/2021 Semester 1  
 Folder: Submission – Project for Part 1

Note: Make sure to upload your zipfile into the correct folder as specified above.

#### E. Due date

The project report and code are due on 18 October 2020, 11:59 PM, Singapore time.

## IV. ASSESSMENT

The project will be assessed based on

1. The contents and the presentation of the report, and
2. The functionality and readability of the Python code.