Deep Reinforcement Learning

Homework #4

Due: 2024/05/14 (Tue.) 23:59

Problem Description

1. Learn to Move: Walk Around (“`<Student_ID>_hw4_<train|test>.py`”)
   This is a NeurIPS 2019 challenge. In this challenge, your task is to develop a controller for a physiologically plausible 3D human model to move (walk or run) following velocity commands with minimum effort.

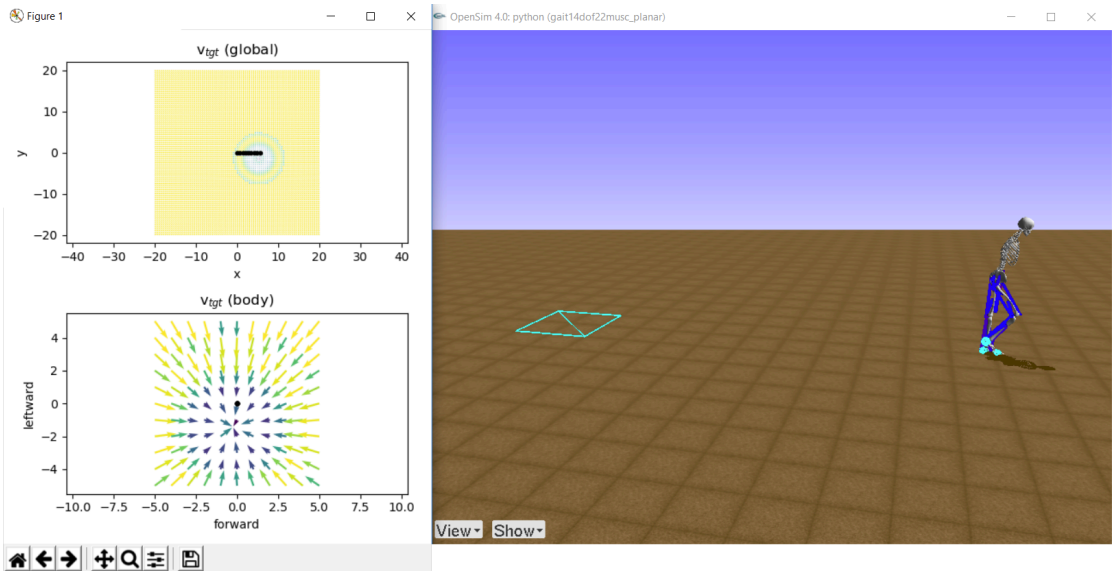

Figure1. Demonstration of Learn to Move: Walk Around Environment

You need to train your agent on “`L2M2019Env`”.

a. Please refer the environmental setup to the OSIM_RL.
b. The observation consists of a velocity map **V** and a body state **S**.
c. The action space is comprised of the muscle activations of the 22 muscles, which is defined as $[0, 1]^{22}$.
d. The reward signal is defined and described in osim_rl document.
e. You may implement your policy with any RL model.
f. You must write all your training and testing code by yourself.
g. You may store your learned results in an external file **“`./<Student_ID>_hw4_data`”** (read-only, max 50 MB), and access it with your program (during testing).
h. After the deadline, TAs will compare your agent with your classmate and grade it.

Detailed Rules for “Learn to Move: Walk Around”

1. You should implement an “`act(observation)`” function in your testing code. See “`random_agent.py`” for an example.
2. Please make sure you pass the checkings provided by *checker.py*.
3. If your program outputs invalid moves, you lose and the game ends immediately.
4. Time limit for each move is **1 second**, and the memory limit is **4 GB**. (Note that the 1-second duration may vary depending on different processors. If you only conduct a single forward pass and doesn’t perform additional calculations during

inference, you don't need to worry about the time limit.)

5. Please use OpenSim 4.0, since the simulation of previous versions may vary.
6. You are allowed to access an external file for loading your learned policy. You can read the file at the following path: **"./<Student_ID>_hw4_data"**.
7. You are allowed to use the following Python package:
   a. numpy, scipy, gym, pandas, **tensorflow**, **pytorch** and the packages mentioned in the environment's repo.
   b. You are allowed to use Python default installed packages. (e.g., sys, time, pickle, random, etc.)
   c. If you need to use other packages, state your reasons and post on eeclass discussion forum.
8. You are not allowed to use the following Python package:
   a. `stable_baselines`

Program Submission

1. For this assignment, please use Python to implement the a source files.
2. Your files must be named as:
   a. "<Student_ID>_hw4_train.py"
   b. "<Student_ID>_hw4_test.py"
   c. "<Student_ID>_hw4_data"
   d. Please make sure that all characters of the filename are in lower case. For example, if your student id is `108062000`, the name of your program file should be `108062000_hw4_train.py` and so on.
3. Your program will be run in a GNU/Linux environment with Python 3.6.1:
   `python <Student_ID>_hw4_test.py`
4. 0 points will be given to Plagiarism. NEVER SHOW YOUR CODE to others and you must write your code by yourself. If the codes are similar to other people and you can't explain your code properly, you will be identified as plagiarism.
5. 0 points will be given if you violate the rules above.
6. If you use modularized / OOP code and want to use multiple files to keep your code structured, please upload it along with the 3 files above.
7. Submit your work to eeclass before the deadline.

Leaderboard Submission

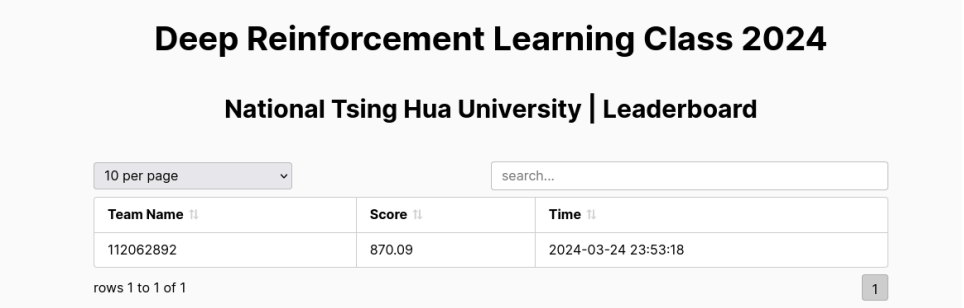

Figure2. Leaderboard Overview

A leaderboard will be provided for comparison, significantly influencing your overall score based on your ranking.

Submit your agent to the leaderboard as instructed in the repository. Ensure your agent is set to **CPU mode** for leaderboard submission. The leaderboard will evaluate your agent across 10 epochs, with a maximum epoch limit of 120 seconds. Scores are calculated based on total_reward/10..
Submission to leaderboard is **compulsory**.

Report

1. Please elaborate on how you design your agent. What advanced techniques do you used? The report is graded directly. So, make sure you have included enough details and figures to help TAs grade your report.
2. TAs will not refer to your code when grading your report, so make sure you have taken a screenshot of the important code snippets.
3. The report filename must be **"**`<Student_ID>_hw4_report.pdf`**"**. Please make sure that all characters of the filename are in "**lower case**".

The Grading Policy

1. The project accounts for 15 points (tentative) of your total grade.
2. You must submit both your source code and report. Please remember the submission rules mentioned above, or you will be penalized on your grade. Late submission rules are specified in the Lecture 1 Slides.
3. Please compress all your files directly (do not compress the folder containing your files) and upload to the eeclass before the deadline. (Total 4 files)
4. The baseline agent will be released soon.

- L2M2019Env
  - Round 1 evaluation set (model='3D', difficulty=2, project=True, obs_as_dict=True)
    - Percentage of your reward comparing to the baseline result across all 10 episodes, and your implementation / code structure. (**60 %**)
  - Your performance rank among the class members (**20%**)
  - Your report (including the discussion and analysis) (**20%**)