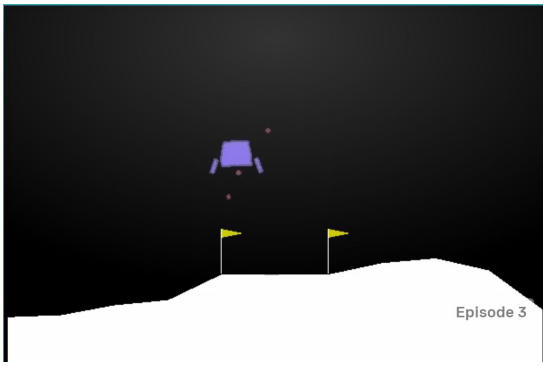
BIU - Reinforcement Learning - Final Project

As part of this course, students must complete a project instead of a final exam.
The goal of this final project is to summarize the main topics that we have discussed in the course using some practice and theory.
Especially the Deep RL lectures and its variants.

In this project you will solve the LunarLanderContinuous-v2 from the openai-gym (https://gym.openai.com/envs/LunarLanderContinuous-v2/)



Your goal is to land the space-ship between the flags smoothly.

The ship has 3 throttles in it. One throttle points downward and the other 2 points in the left and right direction. With the help of these, you have to control the Ship.

Observation Space: [Position X, Position Y, Velocity X, Velocity Y, Angle, Angular Velocity, Is left leg touching the ground: 0 OR 1, Is right leg touching the ground: 0 OR 1]

Continuous Action Space: Two floats [main engine, left-right engines].

Main engine: -1..0 off, 0..+1 throttle from 50% to 100% power. Engine can't work with less than 50% power.

Left-right: -1.0..-0.5 fire left engine, +0.5..+1.0 fire right engine, -0.5..0.5 off

Please note that there are 2 different Lunar Lander Environments in OpenAIGym. One has discrete action space and the other has continuous action space.
In this project you will solve the continuous one.

To output discrete action space you will have to quantize the action into a finite number of states. Please use more actions than the discrete case in the openAI gym. Use at least X2 states than in the discrete case.

Solving the LunarLanderContinuous-v2 means getting an average reward of 200 over 100 consecutive trials.

Link that demonstrate how to use/render the game (with just random actions):

https://colab.research.google.com/drive/1R5BwSTau9zuEj8r4Yh6gB3Nn7NXOm-Fx?usp=sharing

Your goals in this project are

1) To solve the environment

2) As fast as you can(small number of episodes until solving the problem, i.e. you want a small number of crashes until learning the task) - this is a competition part.

3) With a comparison/referring between different variants of what we have learnt in the course (e.g. DQN, target network, network architecture, double-DQN, dueling DDQN, experience replay, prioritized experience replay, TD(lambda), discount factor effects, epsilon-greedy, tabular methods, quantizing effects., use terms from the course)

Reading more papers, Using some advanced policy gradients and actor-critic methods are more than welcome.
The goal of this part is to see your knowledge and theoretical/practical understanding of the different algorithms and hyperparameters.

4) Then, refer to the Lunar-lander with uncertainty. (explanation: observations in the real physical world are sometimes noisy). Specifically, In you need to add a zero-mean Gaussian noise with mean=0 and std = 0.05 to PositionX and PositionY observation of the **location** of the lander.

Use Python & standard DL platform: Keras, TensorFlow, PyTorch

Due Date

Final Project submission is due March 07th at 11:59pm.

Submission:

Final submission
Detailed written Report, graphs, Source code links (to google collabs) , relevant movies and README.
The report should be in the style of a conference paper, including introduction, motivation, related work, etc.
An example: `https://www.overleaf.com/read/mpjvgnhxvxsg`

All writing should be your own -- all quotes must be clearly attributed.

Upload a zipped version to the LEMIDA-BIU
You can upload a link to a **private** github repository to hold the source code, executable and a README file that provides a brief guide to run your code. In this case, you just need to provide the github link within your final report.

Be very clear about what code you've used from other sources, if any. Clear citations are essential. Failure to credit ideas and code from external sources is cheating.
Make sure you evaluate both the good and bad points of your approach.
Even if you didn't accomplish your goal, evaluate what you did.
Do not forget to include the project title, your name and ID in this file.
Max number of pages: 8 (but you don’t have to use them all).

Team size

The project will be performed in groups of 2 students.

Academic Integrity

Team/student may not copy code from other team/students. Copying answers or code from other students for a project is a violation of the university’s honor code and will be treated as such. All suspicious activity will be reported to the Department Head and the university authorities.

Giving code to another student is also considered a violation. Students are responsible for protecting their work from copying.

If you build some of your code on existing work and utilize existing code (your own or code found on the web), you must give proper attribution to all existing work that you build on and make clear what your new contribution is. Any unattributed or uncited work that you use will be considered a breach of academic honesty and dealt with according to the course policy in the syllabus.