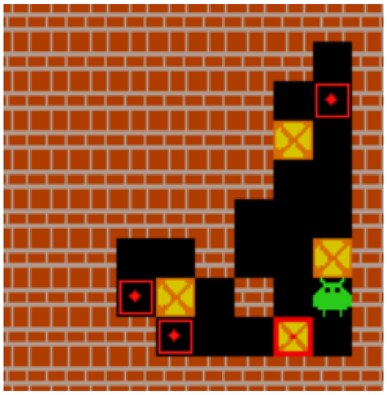
IDC – Reinforcement Learning - Final Project

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.

In this project you will solve the SOKOBAN env. The Environment: https://github.com/mpSchrader/gym-sokoban



Sokoban is Japanese for warehouse keeper and a traditional video game. The game is a transportation puzzle, where the player has to push all boxes in the room on the storage locations/ targets. The possibility of making irreversible mistakes makes these puzzles so challenging especially for Reinforcement Learning algorithms, which mostly lack the ability to think ahead.

The room generation is random and therefore, will allow to train Deep Neural Networks without overfitting on a set of predefined rooms.

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

https://colab.research.google.com/drive/1wY5BdjyJutg6qsOPV26M0denc-dBmksb?usp=sharing

Read the env. documentation carefully and make sure you understand what is the environment, the observation-space, the action-space, and how are the rewards defined. Also, make sure you understand what's the difference between the versions.

Your goals in this project are

1) Solve the following environments:

   a) Easy mode "push&pull": PushAndPull-Sokoban-v0
   b) Sokoban-v1
   c) TwoPlayer-Sokoban-v1

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

Even if you did not solve an environment - Supply a graph describing the average rewards on 100 episodes (after finishing the training phase). Note: stop an episode after 500 iterations (if it was not finished)

3) In your final report, compare different approaches, different hyper-params, etc.

4) Use all knowledge that you got in your course (and more) studies and more papers to solve the challenge.

5) Submit a final report.

6) You need to supply your code as google colab notebooks. Such that the course team can run it.

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

Due Date

Final Project submission is due July 22th at 11:59pm.

Submission:

Final submission

Detailed written Report, graphs, Source code links (to google collabs) , relevant images 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/sztrbdgcbyty`

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

Upload a zipped version to the moodle

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.