CITY UNIVERSITY
LONDON
EST 1894

INM707: Deep Reinforcement Learning
Coursework

V1.0

Submission deadlines:
Report and Code: Sunday 11th April 2020, 5pm
Presentation: Thursday 6th May 2021, 5pm

Introduction

This coursework builds on the material covered in the tutorials and lecture. On completing this coursework, you should be able to implement and understand classical tabular Reinforcement Learning (week 1-5) as well as Deep Reinforcement Learning (week 6-10) algorithms. This coursework builds on the material covered in the tutorial and lecture slides. You will make use of the different concepts learned in the module:

- How to define and implement a Reinforcement Learning (RL) problem
- How to implement solutions in Pythons
- How to evaluate different algorithms

Python should be used for all implementations. Deliverables are:

- A one-page report "Summary" that summarizes some key elements (described in this document).
- A written report of your work (max 10 pages). You are free to provide a single or multiple documents (one for each task).
- Your practical implementation (code)
- An individual presentation (10 minutes) recorded and submitted on moodle.

Module marking:

CW and presentations will both be 100 marks. You need 50 in each of these components to pass.
Your final mark will be 70% Coursework (Code and Report) and 30% Presentation.

Teamwork

This coursework should be completed either in **groups of two** or **individually**; We encourage you to work in pairs, and no additional marks will be granted if you do the Coursework alone. If you decide to work in pair, you should declare it on the report.

All team members are expected to contribute to all parts of the work: both the coding and the report. Teamwork does NOT mean division of labour. You can distribute the leading role for each assignment, but each of you must contribute to all the tasks. If you don't, you will not be evaluated for the tasks that you did not prepare. Distributing the assignments is considered a form of academic misconduct.

You are required to explain your personal contribution to each task in the Summary report, in the reflection part.

There is a maximum number of modules you can work within the same team. You cannot operate with the same team in more than 2 modules per term and no more than 4 in total.

The Coursework is divided into 4 different Tasks. Even if you are in a team, Task 2 should be solved individually, and no teamwork is allowed for the Task 2.

Submission

Submission is through Moodle, and no other method of submission will be accepted. You should submit the following files:

- Summary report: 1 page
- Main Report: Task 1-3, 10 pages
- All references are counted as additional pages, and you can put as many as necessary.
- Optionally: Task 4, 2 additional pages
- There will be no appendix.
- Zip file of the Code for Task 1-4

Format for reports: pdf format, single column, standard A4 margins, standard default line spacing of 1.15, font Arial 11, including all figures. If you are working alone, you can present a single report. If you are working in teams, you are free to submit a different reports or to concatenate them.

Late submissions will score 0. You can upload work to Moodle more than once, so there is no need for last minute submission. Don't leave final submission to the last minute.

Presentation

You will be recording an individual presentation of 10 minutes. You need to prepare adequate material (slides) to serve as support for your video (but you don't have to submit the slides). During this presentation, you will present the results for the tasks 1-3. Optionally, you can also present task 4 using an additional 2 minutes.

This is an individual exercise.

The Audio and Video should be of a quality sufficient to follow your presentation. When in doubt, you can record a short sample and send it to us for feedback.

We will mark the content as well as the material (quality of the slides, delivery of the message).

Feedback

In the labs and surgeries, we can check your progress and give formative feedback. Evaluative feedback and marks on your coursework will be given out after the presentations.

Code

- You should have one jupyter notebook per task
- Class and function definitions should be moved to external modules that the jupyter notebook imports.
- Make sure that the import works on windows.
- Each task should be presented as an individual Jupyter Notebook with additional modules and packages that you developed and are used by the notebooks.
- Code quality, clarity and organization will be taken into account in the marking.
- Code should be documented to allow straightforward understanding of the different components (don't document every single line, but comment enough so that we understand the role and logic of each method / function / class)
- All the code should be included in the submission as a single zip file.
- You can take inspiration from sources and borrow portion of the code. However this should be declared in the summary as percentage of code borrowed, and it will influence your mark.
- All your results should be reproducible using the jupyter notebooks.
- Don't forget about the README if you wish to give details about third-party libraries that we might need to install to run your code.

The code will be marked independently from the report. How much code was borrowed will factor in the overall grade.

The Tasks

In this coursework, you are expected to demonstrate what you have learned in the module in terms of Tabular RL and Deep RL. Additionally, you have the occasion to work on a problem of your choosing (related to RL) for additional points.

The maximum number of marks which can be scored is 100. You can gain up to 20 additional marks in Task 4, but the maximum of marks obtained for the Report and Code is capped at 100.

In all tasks, you can use the built-in libraries of python (math, random, …), numpy, and matplotlib. If you think that you might benefit from using another library, you can ask about it on Moodle. You will use PyTorch in Task 3, and you can use any library in Task 4.

Task 1: 20 marks

You need to design and develop a tabular RL environment that follows an Markov Decision Process. In this task, the environment will provide a state that is used to train your RL algorithm. Your report should answer the following points:

- Description of the environment
- Description of the agent and its actions

- Description of the different dynamics of the environment (the rules of the game), as well as rewards

The environment should be non trivial and:

- Include some stochasticity (for example, obstacles that moves randomly)
- Terminate (define the termination conditions)
- Be different from the environments proposed in the labs.

In this first task, the number of different states that the agent can reach should be finite. The total number of states can be very large, but bear in mind that classical RL algorithms might not converge. So a good idea is to parameterize the number of states (e.g. in the labs, the number of different states where dependent on the size of the environment), to make sure that your problem is solvable.

An alternative is to use an external environment that you modify. However the modifications should be substantial and demonstrate an effort on par with the development of a new environment.

Task 2: 30 Marks

The second task is about implementing classical RL algorithms. You have the choice between implementing Q-learning or Dyna-Q.

For this task, you should:

- Describe and explain the algorithm you chose.
- Conduct a case study of how this algorithm performs on the environment you implemented in Task 1.
- Evaluate the performance of your algorithm.
- You should evaluate the effect of the different hyper-parameters.
- If your environment is parameterizable, a good addition would be to evaluate how your RL algorithms performs when the environment becomes more and more complex.

Task 3: 50 Marks

The third task requires to implement a Deep Reinforcement Learning algorithm. You cannot present DQN and its different improvements for this Task, as we will cover them in the labs.

You can select any of the following algorithms to implement and evaluate:

- Policy Optimization (e.g. Proximal Policy Optimization, Advantage Asynchronous Actor Critic, …)
- Q-learning: Hindsight Experience Replay
- World Models
- Soft Actor Critic

For reference, the algorithms are presented here:

https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#citations-below

If you want to implement a different algorithm, please ask on Moodle whether it is suitable.

You can use any Deep RL environment available online, or use a scaled up version of your environment developed in Task 1. For example, if the complexity of the environment in Task 1 can be scaled up to the point where tabular approaches (e.g. Q-learning and Dyna-Q) can't perform well, then it is appropriate to use them. You might need to adapt your environment slightly to return observations instead of states.

For this task, you should:

- Describe and explain the algorithm you chose.
- Describe the environment, if it is different from Task 1.
- Conduct a case study of how this algorithm performs on the environment you choose.
- Evaluate the performance of your algorithm
- You should evaluate the effect of the different hyper-parameters
- If your environment is parameterizable, a good addition would be to evaluate how your RL algorithms performs when the environment becomes more and more complex.

Task 4: 20 Marks – optional

For this last optional task, you can work on a topic of your choosing. It must be related to the Module, and different from Tasks 1 to 3. Note that it can constitute a continuation of Task 1, 2 or 3.

Reports

You must submit a Summary page that includes:

- Your name, student id, and team member
- For each task, the percentage of borrowed code and reference to the sources. You need to present a fair estimate, and you should declare if you wrote everything yourself.
- Your personal reflection

Your final report should cover each of the aspects described in this document (and any other element of your work that you believe should be reported). Graphically illustration of your results is expected as well as numerical results.

You should present the results clearly and concisely and provide a discussion of the results, with conclusions related to problem being addressed. The conclusion might propose some further work based on the results of the coursework.

Personal reflection

You should reflect briefly on the outcome of this module and coursework. This part does not influence the marking, but is mandatory.

In the case of teamwork, the reflection part should briefly address who did what. You might also report further conclusions and discussion of your work from your individual perspective in this section.

Note

You are not necessarily being marked on how good the results are. What matters is that you try something sensible and clearly describe the problem, method, what you did, and your interpretation of the results.

Coding & Referencing

This is, in large part, a coding assignment. If you use code (or other materials) written by someone else, you must **cite** that code (or other material). If you do not cite work appropriately you will have committed academic misconduct. Making superficial changes to the code does not make it yours. You are also expected to make a coding contribution, so if you use a large amount of code written by someone else, and cite it appropriately, your contribution will be low, and your work marked accordingly.

Extenuating Circumstances

If you are not able to submit your coursework on time for unforeseen medical reasons or personal reasons beyond your control you should contact the Programmes Office as soon as possible and fill an Extenuating Circumstances form. Strong evidence in the form of, for instance, medical certificates or legal statements will have to be produced.

Plagiarism

If you copy the work of others (either that of another team or of a third party), with or without their permission, you will score no marks and further disciplinary action will be taken against you. The same applies if you allow others to copy your work.

Note that borrowing code from external sources is not considered plagiarism if it is properly declared.

Checklist

You will lose points if you don't provide the following:

- Summary page with all the required information
- Code as jupyter notebook + modules
- Reports with citations
- Presentation of good Audio/Video quality