CM30359: Reinforcement Learning Group Project

What you need to do

Apply reinforcement learning methods to solve a problem of your choice. You may choose any problem that engages your interest, but **its state/action space must be large enough so that tabular reinforcement learning methods cannot be used to solve it effectively**.

Some problems that have been tackled in past projects include:

- teaching a simulated bipedal robot how to navigate an obstacle course,
- teaching a four-legged creature how to walk,
- driving a simulated car around a racetrack,
- playing Atari arcade games,
- playing Super Mario Bros.,
- landing a simulated spacecraft on the moon,
- playing tile-matching games like Candy Crush and Bejeweled, and
- swinging-up a simulated double pendulum.

Your objective is to solve your chosen problem to the best of your ability. You are allowed to use any **function-approximated** reinforcement learning method you believe to be appropriate, and you are not limited to methods we have explicitly covered in the unit.

You should work in groups of four to six students. You will have the option of forming your groups on your own; otherwise, you will be assigned to a group randomly. Multiple groups are allowed to work on the same problem.

What you need to submit

For this project, you will need to submit five deliverables:

- A project report,
- a video presentation,
- a video of your agent(s) before and after learning,
- your source code, and
- a group contribution form.

Project report

Your project report *must* be divided into the following sections:

1. **Problem Definition:** A clear, precise and concise description of your chosen problem, including the states, actions, transition dynamics, and the reward function. You will lose marks for an unclear, incorrect, or incomplete problem definition. You should also discuss

the difficulty of your chosen problem and justify why it cannot be solved effectively using tabular reinforcement learning methods.

2. **Background** A discussion of reinforcement learning methods that may be effective at solving your chosen problem, their strengths and weaknesses for your chosen problem, and any existing results in the scientific literature (or publicly available online) on your chosen problem or similar problems.

3. **Method:** A description of the method(s) used to solve your chosen problem, an explanation of how these methods work (in your own words), and an explanation of why you chose these specific methods.

4. **Results**: A presentation of your results, showing how quickly and how well your agent(s) learn (i.e. improve their policies). Include informative baselines for comparison (e.g. the best possible performance, the performance of an average human, or the performance of an agent that selects actions randomly).

5. **Discussion:** An evaluation of how well you solved your chosen problem.

6. **Future Work:** A discussion of potential future work you would complete if you had more time.

7. **Personal Experience:** A discussion of your personal experience with the project, such as difficulties or pleasant surprises you encountered while completing it.

8. **References**

9. **Appendices:** Appendices may include (1) a detailed description of the problem domain, including the states, actions, reward function, and transition dynamics; (2) all experimental details so that the reader can fully replicate your experiments; (3) how you selected your hyperparameters (if applicable); and (4) any additional supporting results that you could not include in the main body of your report. Note that your appendices should be clearly referenced in the main body of your report.

Your report should be written using the provided LaTeX template, and should be no longer than seven pages including figures but excluding references and appendices. The content of all figures, including any embedded text, should be clearly legible; you will lose marks for figures and text that are too small to view comfortably. You should not modify the provided LaTeX template. Any Appendices should be clearly referenced in the main body of your report. All sources should be referenced appropriately. Each group should submit a single report. This coursework is not marked anonymously.

Video presentation

Your video presentation should summarise the most important pieces of information present in your report. Somebody should be able to watch your video and leave having a good understanding of your chosen problem, solution method(s), and results. Aside from that, you are free to organise your video presentation as you wish. Your video presentation should be no longer than four minutes, recorded at your natural speaking tempo. All group members must take part in the video presentation.

Video of agent performance

You should submit a video showing your agent's performance before and after training, clearly demonstrating that some learning has occurred. This video should be at most two minutes long (but can be shorter as long as it demonstrates learning), but may be sped up as long as the content remains clear. If you cannot upload your video directly to Moodle, you may include a link to it in your report's appendices; in such cases, the link must remain publicly accessible for at least three months after the submission deadline.

Source code

You should submit all source code used to complete this project. We should be able to run your code and reproduce your results ourselves if needed, so you should include a short README file explaining how to do so.

Group contribution form

You will be working as part of a group to complete this project. As such, you should ensure that you organise your time and resources effectively: finding time to plan, set expectations, discuss your work, and agree on each group member's responsibilities is essential.

We have provided you with a group contribution form to fill out and submit. To fill out this form, you must agree on the relative contributions of each group member to your final project. We will use these relative contributions to scale marks where appropriate. For more information, please see the provided group contribution form document.

Marking criteria

Taking into account your report, presentation, agent demonstration video, and code, we will be marking you against the following four criteria:

- **Quality:** How high was the performance of your trained agent relative to the difficulty of the problem that you chose to solve?
- **Rigour:** How thoroughly have you evaluated the performance of your agent?
- **Understanding:** How deep is your understanding of reinforcement learning in general, and of the methods you have implemented in particular? Have you made principled decisions when selecting your chosen solution method over potential alternatives?
- **Presentation:** How clear, concise, and well-organised are your report and video presentation?

To achieve a grade above 40%, students must implement an agent that demonstrates a noticeable amount of learning in their chosen domain. At this level, the students' chosen problem may be relatively simple, but its state/action space must be large enough so that function-approximated reinforcement learning methods are necessary to solve it effectively. Students should perform some basic analysis of their agent's performance, but may not provide comparisons to useful baselines and alternative methods. Students should demonstrate a basic understanding of the method(s) they chose to implement, but may demonstrate only a limited appreciation of possible alternatives. The students' writing style may be weak, with little effort made to ensure correct spelling and grammar. Students may position figures haphazardly and may not discuss them appropriately. The ideas and arguments presented in the students' report and video presentation may be unorganised and unclear.

To achieve a grade above 60%, students must implement an agent that demonstrates a substantial level of learning in their chosen domain, relative to the difficulty of their chosen problem. Students should thoroughly evaluate the performance of their agent(s), and include comparisons with key baselines and some alternative approaches. Students should demonstrate a good understanding of their chosen method(s) and give reasonable justifications for their algorithmic choices. This should include demonstrating a reasonable appreciation of the strengths and weaknesses of alternative methods. Throughout the report, the students' writing style should be clear, consistent, and correct. Students should use figures effectively to present key information and results, discuss their content appropriately, and put thought into their placement. Students should present key points clearly in both their report and video presentation, and their arguments should lead to natural and well-justified conclusions.

To achieve a grade above 70%, students should implement an agent that demonstrates a high level of performance in their chosen domain, relative to the difficulty of their chosen problem and the amount of time and compute resources available. In simpler domains, a near-optimal policy may well be learned. The solution methods developed should go beyond the content covered explicitly in the unit. Students should perform an in-depth evaluation of their chosen method(s) and include comparisons to and evaluations of additional baselines and alternative methods. Students may show evidence of creativity in their algorithmic approach and analysis. Students should demonstrate a deep understanding of their chosen method(s), as well as possible alternative approaches. All algorithmic choices should be well-justified. The students' writing style should be professional and fluent throughout. Students should use figures effectively and selectively to present key information and results, as well as to complement text where appropriate. The students' report and video presentation should be well-organised, and ideas and arguments should be conveyed clearly and concisely at an appropriate level of detail.

Grades above 70% will normally be rare.

Penalties

All five deliverables are mandatory. A penalty of −10 marks will be applied for each deliverable that is missing from your submission.

A penalty of −5 marks will also be applied for each of the following issues: exceeding the specified page or time limits, not using the provided LaTeXtemplate to write the report, not following the prescribed report structure, not recording the video presentation at your normal speaking tempo, or not having all group members participate in the video presentation.

We will not read beyond the first seven pages of the main body of your report, and no marks will be given for content beyond this point. The same applies to your videos: we will not watch or award marks for content beyond the specified time limits.

Choosing a problem that can be solved effectively using tabular reinforcement learning methods will result in a failing grade. You are encouraged to get feedback on your chosen problem from tutors during lab sessions to ensure that its state/action space is large enough to be suitable for this project.

Using Third-Party Code

You may use third-party libraries to provide generic functionality or to implement your chosen environment. For example, you may use libraries like Tensorflow or PyTorch to implement neural networks, or use an existing environment implementation from a library like OpenAI Gym.

You *must* implement your chosen reinforcement learning solution method(s) from scratch. You must design your own implementation and write your own code. You must not use code you find online or in libraries like OpenAI Baselines. You will only be given marks for reinforcement learning methods you have implemented from scratch. A submission that does not implement any reinforcement learning solution methods from scratch will receive a failing grade.

The *only* instance in which it is permissible to use an existing implementation of a reinforcement learning algorithm is if you include it solely as a baseline to compare the performance of your own implementation(s) to. If you choose to do this, you must make it clear which algorithms you have implemented from scratch and which you have used existing implementations of.

If you use third-party code sources as *any* part of your implementation, you should acknowledge them in your README file. You should provide a reference for each source, and clearly explain (1) how you used it, (2) what you used it for, and (3) any modifications or extensions you implemented on top of it.

Plagiarism

Please ensure that you develop your solution and write your report independently of other groups. Cite literature sources appropriately and give clear credit to the authors of any third-party code you use as part of your solution. Please do not use generative artificial intelligence ("GenAI") tools to create any part of you submission.

Do not plagiarise. Plagiarism is a serious academic offence. Both your source code and report will be checked for evidence of plagiarism. For details on what plagiarism is and how to avoid it, please visit http://www.bath.ac.uk/library/help/infoguides/plagiarism.html.