Reinforcement Learning Assignment

COMS4061A/COMS7071A

William Hill (william.hill1@students.wits.ac.za)
Simon Rosen (simon.rosen1@students.wits.ac.za)

Due date: 22 October 2024, 23:59

1 Introduction

Power networks are an essential part of modern society, with nearly everything we do on a daily basis having some dependence on electricity. However, they are complex and need to be constantly managed in order to deal with any power perturbations or interruptions that might occur and ensure a reliable and stable supply of electricity to all parts of the network.

Your task will be to implement an agent that acts on the powergrid, subject to a number of constraints (i.e. operational rules), with the goal of ensuring a reliable flow of power. You will submit a report of your findings, amongst other things.

2 Environment

Grid2Op (Donnot 2020) is a library that simulates the performance of a powergrid, whilst various changes are made to it (such as changing generator setpoints, implementing load shedding, and altering the topology of the grid etc.). The original goal of Grid2Op was for the "Learning to Run Power Network" (L2RPN) competitions (Marot *et al.* 2020 2021 2022), which aimed to train RL agents to act as grid moderators and ensure a stable flow of electricity across the grid.

The GitHub repo can be found at `https://github.com/rte-france/Grid2Op` and the official documentation can be found at `https://grid2op.readthedocs.io/en/latest/`. Look at these for installation and setup instructions (Grid2Op is compatible with the OpenAI Gym interface, which you are required to use in this assignment). Specifically, the following two pages handle the observation and action spaces, respectively:

- `https://grid2op.readthedocs.io/en/latest/observation.html`
- `https://grid2op.readthedocs.io/en/latest/action.html`

3 Requirements

You are allowed to work in groups of at most 4 people. Each group will be required to do the following for this assignment:

1. Implement one learning algorithm from the course, and one algorithm that has not been covered in the course (if in doubt, ask one of the tutors). These **two** agents will serve as a base from which you will make improvements.
2. For the algorithm that has not been covered in the course, you need to explain the algorithm in your report, to show that you understand how it works.
3. The **core** part of the assignment will be to iteratively improve the base agents. Specifically, you will need to implement the base design, evaluate its performance, identify areas for improvement, and then make those improvements. This iterative process will need to be done a few times (i.e. simply making one improvement will not be sufficient).
   (a) For each iteration, you will need to present the results and analysis of your agent's performance, and use this to motivate the design changes to make for the next iteration.
   (b) The improvements cannot be trivial. They need to be some change to either the learning algorithm, the policy model, some preprocessing of the various elements of the MDP (e.g. action or observation spaces, the reward function) etc. *For example, one such improvement may be use a subset of the observation space. However, simply changing hyperparameters is insufficient.*
   (c) We require a minimum of **TWO** improvements to be made — you are allowed to make further improvements, and these will be considered for extra credit. So, at a minimum, your investigative pipeline would be:

	implement base agent	
$\rightarrow$	evaluate agent	**(Eval 1)**
$\rightarrow$	make a potential improvement	**(Improvement 1)**
$\rightarrow$	evaluate agent	**(Eval 2)**
$\rightarrow$	make a potential improvement	**(Improvement 2)**
$\rightarrow$	evaluate agent	**(Eval 3)**

   (d) **Remember that this iterative process of improvement needs to be done for both of your chosen algorithms.**
4. Once you have completed all your improvements to each agent, compare their final performance and their strengths and weaknesses. This could include points like how well each agent achieves the goal, and their rates of learning.
5. If you use any existing research or code in any of your solutions, you are required to cite it, AND make some kind of modification to it (e.g. shaping the reward, augmenting input, using sub-task curricula etc.) and describe it in the report.

We are mainly interested in your approach to the solution, not necessarily in how well you can solve the problem. So make sure your agent evaluations are complete and provide evidence for your design decisions, which will then be reasonable.

When you are presenting results, please choose a suitable format to convey them. For example, don't present the return over episodes in a table; rather use a line graph.

4 Particulars and Restrictions

Please take note of the following:

1. You are allowed to use libraries (like Stable Baselines) to implement and train your agents.
2. Grid2Op was designed for multiple different research communities. Please ensure that your solution is RL-based and is not something like a planning solution.
3. When installing and setting up Grid2Op, please ensure you use a gym environment interface, and not a grid2op interface. Using a gym environment will allow you to interface with libraries more easily.
4. Use the "l2rpn_case14_sandbox" Grid2Op environment (e.g. when calling `gym.make()`). The other environments available all include extra features (such as adversarial opponents) that are not required for this assignment. The specified environment simply concerns the flow of power in the grid.
5. The main reward you will need to optimise is made up of two parts:
   (a) L2RPN Reward: This aims to maximise the flow of power through the grid.
   (b) N1 Reward: This aims to satisfy the "N-1" conditions imposed on Powergrids, which just means that there cannot be a single point of failure in the network — if a component fails, power should still be able to flow to all parts of the network.
   (c) We have a `CombinedScaledReward`, which weights these two rewards into a single scalar, that the agent can use for training.

   Please note that you are allowed to alter this reward (e.g. through reward shaping), but for all evaluation you do, please include performance on this reward.
6. We have provided a wrapper to create the Gym environment and to use the required environment and rewards. It has been well-commented with what you can and can't change, and what you should be completing yourself. We have also provided links to the relevant tutorials and documentation to help you understand different areas of the problem (e.g. the observation and action spaces).
   (a) The wrapper includes a Random agent, to demonstrate the correct way of interacting with the environment.
   (b) If you want to restrict the size of observation or action spaces, then you will need to do so yourself, after having received the observation back from the environment. This will need to be done in the functions `setup_observations` and `setup_actions`.

(c) Please note that the default observation space is much larger than we require to solve the problem. So it would be beneficial to investigate the different observation elements and filter out unnecessary ones.

(d) Depending on the 3rd party library you are using, you will also need to use these functions to restructure your observations and actions to make them compatible.

7. **If you provide an invalid action to the environment to execute, it will automatically replace the action with a `no_op` action (i.e. nothing will happen).** Further information can be found here.

5 Evaluation

Marking will be done on the process of making your design decisions based on prior results, and not necessarily on how well your agent is able to perform.

However, you also need to submit your source code, so your results need to be reproducible if we test your agents ourselves.

We will be using the Combined Reward, detailed in Particulars and Restrictions (Point 5), to evaluate your agents' performances. Please make sure to include results using only this reward throughout your investigation.

6 Submission

Each group will be required to submit the following:

1. A report, detailing the content specified in Requirements. Only one group member should submit, but all group members' names and student numbers must be specified at the top of the report.
2. The source code. Marks will be awarded for the quality of your codebase — things like comments, Read Me files (with an explanation of your directory structure), run scripts, and code versioning. You should include a link to the GitHub repository in your report.
3. Make sure to include the hyperparameters used for your training in your report.

References

[Donnot 2020] B. Donnot. *Grid2op- A testbed platform to model sequential decision making in power systems.* . `https://GitHub.com/rte-france/grid2op`, 2020.

[Marot *et al.* 2020] Antoine Marot, Benjamin Donnot, Camilo Romero, Balthazar Donon, Marvin Lerousseau, Luca Veyrin-Forrer, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. *Electric Power Systems Research*, 189:106635, 2020.

[Marot *et al.* 2021] Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. Learning to run a power network challenge: a retrospective analysis. In *NeurIPS 2020 Competition and Demonstration Track*, pages 112–132. PMLR, 2021.

[Marot *et al.* 2022] Antoine Marot, Benjamin Donnot, Karim Chaouache, Adrian Kelly, Qiuhua Huang, Ramij-Raja Hossain, and Jochen L Cremer. Learning to run a power network with trust. *Electric Power Systems Research*, 212:108487, 2022.