Advanced Machine Learning Proseminar
Programming Project

Sayantan Auddy

15.12.2020

1 Environment

The acrobot environment (`https://gym.openai.com/envs/Acrobot-v1/`) consists of a 2-link pendulum in which only the joint between the two links is actuated (force can be applied). The objective is to swing the end-effector to a height of one link above the base, as shown in Fig. 1.

State: The state space is continuous. The state consists of the the sin() and cos() of the two rotational joint angles and the joint angular velocities: $[\cos(\theta_1)\ \sin(\theta_1)\ \cos(\theta_2)\ \sin(\theta_2)\ \dot{\theta_1}\ \dot{\theta_2}]$. For the first link, an angle of 0 corresponds to the link pointing downwards. The angle of the second link is relative to the angle of the first link.

Action: The action space is discrete and consists of 3 possible actions: applying +1, 0, or -1 torque on the joint between the two links.

Reward: The reward is 0 if the goal is reached, or -1 for every timestep otherwise.

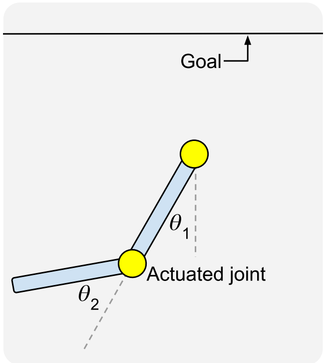

Figure 1: Acrobot

View further descriptions in the source code for this environment at `https://github.com/openai/gym/blob/master/gym/envs/classic_control/acrobot.py`.

After installing *gym*, the acrobot environment can be set up using the following snippet:

```
import gym
# Create the gym environment
env = gym.make("Acrobot-v1")
```

Refer to the code from the Proseminar for the details of how to use an environment.

2 Tasks

For this project, you will be working in **teams of 2 people** (any deviation from this rule must be discussed with the instructor).

The aim of this project is to train agents using Reinforcement Learning algorithms for solving the acrobot task. For this, you will need to implement at least 2 different RL algorithms that are capable of solving environments with continuous states and discrete actions:

(i) Semi-gradient SARSA **OR** Q-Learning with a neural network as the function approximator.

(ii) A modern deep RL algorithm of your choice which makes use of value functions (e.g. DQN, any actor-critic RL algorithm, etc.).

You have to describe your experiments and findings in a written PDF report and you also need to submit the code that you used for your experiments. The details about what you need to submit are provided below:

2.1 Report

- Your project report should consist of 4 to 5 pages (excluding references). **Use the report template provided to you**.
- The names of both team members **must** be mentioned in the designated place in the report.
- The report should have the following sections. Given in the brackets is the suggested length of each section:
  - **Introduction** ($\frac{1}{2}$ page): Describe the problem you are solving (very briefly), and also include a short overview of your approach and how the rest of the report is organized. Point the reader to your main findings.
  - **Methods** ($\frac{1}{2}$ page): Mention the algorithms you have chosen, and describe the reasons for your choice. You **don't** have to describe the algorithms in detail. Remember to include relevant references.
  - **Setup** (1 page): Describe your experimental setup. Clearly mention what hyperparameters you use for training your chosen algorithms. You can also describe here the structure of the neural networks that you use. Feel free to include any other setup-related information.
  - **Experiments and Results** (2 pages): Describe in detail how you trained your agents, and what the result of the training was. Compare the performance of your chosen algorithms against one another. As a lower baseline, also compare the performance of these algorithms against an agent that always takes random actions. Comment on the hyperparameters that you feel are important for performing well. You can also remark on specific features of your implemented algorithms that you feel contributed to their good or bad performance. All plots and tables should be included in this section (remember to use proper axis labels, legends, and titles for your plots).
  - **Conclusion** ($\frac{1}{2}$ page): Analyze your results and summarize your main findings. Also, share your thoughts on how the performance of your algorithms can be improved further.
  - **References**: Bibliography for the papers, or book chapters that you reference in the report.

2.2 Code

In addition to the report, you also need to submit the code that you used for the results in the report. For this, refer to the following:

- Create a zipped folder with all your code files and documentation.
- The zipped archive **must** include a requirements file listing the dependencies. If any special steps are to be followed for some libraries, please mention them.
- The archive **must** contain a README file describing the contents. This README should also state the steps that are necessary for reproducing the results in the report (which scripts to run, the sequence of scripts, etc.).
- You can organize your code in Python scripts or in Jupyter notebooks, but mention how your code should be run in the README.
- Please include adequate comments in your code and use proper docstrings for your functions so that your code is nice and readable.
- You are free to use any libraries you need, but you need to implement the RL algorithms yourselves.
- You are also free to reuse your code from the Proseminar exercises and assignments.
- If you utilize a lot of code from some online source, make sure that you acknowledge this source.

3 Divison of Points

The total number of points for this project is 40. Your final project report and code will be evaluated on several factors over which the total points for the project will be distributed. These are:

1. Report **[25 points]**
   - Clarity of writing [5 points]
   - Description of experimental setup and hyperparameters [5 points]
   - Presentation of experiments (including properly labelled plots, images) [10 points]
   - Analysis of results [5 points]
2. Code **[15 points]**
   - Organization (proper README, requirements file, installation instructions) [2 points]
   - Readability (proper comments and docstrings) [3 points]
   - Reproducability (how easily can your results be reproduced) [10 points]

4 Submission Instructions

- **Due date** for submission is Tuesday 02-February-2021 at 23:59.
- You need to have 2 files for submission:
  1. Project report titled **user1_user2_report.pdf** (replace user1 and user2 with your c-numbers).
  2. Code archive titled **user1_user2_code.zip** (replace user1 and user2 with your c-numbers).
- Only one member of the team should submit the report in PDF format and the complete source code archive.
- You will receive separate instructions regarding how to submit your report and code.