COMP47590

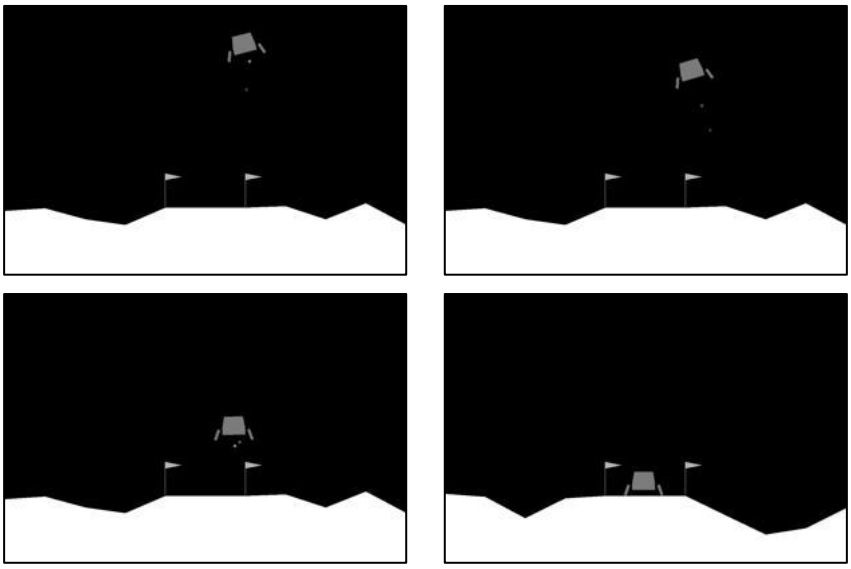
Advanced Machine Learning

Assignment 2: LunarEirLander

Introduction

Nasa Perseverance rover landed on Mars in early 2021. One phenomenal part the landing system is the *sky-crane* which uses thrusters to briefly hover over the Martian surface and lower the rover into place. This video of the landing is simply phenomenal: https://www.youtube.com/watch?v=GUqsH5y1j1M

In this assignment we will train a simplified sky-crane type craft using the OpenAI Gym Lunar Lander environment. In the OpenAI Gym Lunar Lander environment (https://gym.openai.com/envs/LunarLander-v2/) the player's job is to control a small spaceship to land if safely on a landing pad. There are three *thrusters* which can be used for control. These work in three directions: *up, left,* and *right.* The player can also choose to do nothing. A dataset has been collected from an expert player of Lunar Lander that contains screenshots of the state of the game and the player's associated action (*none*, *left, up,* and *right).*



Tasks

Perform the following tasks:

1. Train a **supervised machine learning** model to control the Lunar Lander craft based on the image dataset and perform a suitable evaluation experiment (based on the dataset) to determine how effective the model trained is.

   Note the following:

   - A dataset is provided containing multiple examples of an expert plying the game along with their move for each frame. The dataset can be downloaded from here:

     https://www.dropbox.com/sh/8svd09y5ppxb333/AAAT7Vp4zdgqvvIOVzadWLn-a?dl=0

- Each image shows the state of the world and the last digit in the image filename indicates the action that the expert player took in that scenario (0: *none*, 1: *left*, 2: *up*, 3: *right*).
- A simple LeNet-5 convolutional neural network model architecture is a good place to start for this problem.
- **LunarEirLander.py** includes a modified version of the LunarLander game that should be used for all tasks.

You should include at least two of the following variations (or others of your own):

- You should consider resizing the images.
- You should consider using a pre-trained model (e.g. VGG-16 trained on ImageNet)
- You should consider using multiple sequential frames as input to the model rather than a single frame.
- You should consider handling the class imbalance in the dataset.

2. Use the **DeepQLearning** reinforcement learning algorithm to train an agent to play the Lunar Lander game and perform a suitable evaluation to determine how effective the model trained is.

   Note the following:

   - Use either a pixel-based representation or feature-vector-based representation for state in this model.
   - If using pixel-based representation, consider using multiple sequential frames as input to the model rather than a single frame.

3. Deploy each of the two models trained to the Lunar Lander game to play 200 episodes and analyse the reward achieved by the models trained using each approach.

   - The **LunarLanderImageClassifierAgent.ipynb** contains an example of loading a saved model and running an iteration of LunarEirLander using that model.

4. Write a short document of **no more that 500 words** (either in PDF or Jupyter notebook format) describing your work

   - Explain any decisions you made in designing your approach.
   - Present the results of your experiments
   - Reflect on the performance of each model.
   - Reflect on the amount of computation required to train each model.

Notes

The following notes may be useful:

- **Can I Use Scikit-Learn, Keras And Other Python Packages?** Yes, and you absolutely should!
- **It's Taking Forever!** The datasets are big, and reinforcement learning takes a long time. If you find things are taking too long feel free to down-sample the dataset or not let reinforcement learning run too long. Submissions will not be penalised for this. Google Colaboratory (https://colab.research.google.com/) might be a useful resource for accessing computation.
- **Can I Work In A Team?** Teams of up to two people are allowed. All team members will receive the same mark. There is no penalty for submitting as a team, and no reward for submitting as an individual.
- **My Computer Controlled Players Are Not Very Good!** It is tricky for the supervised machine learning models to learn a complete model of the game so performance is unlikely to be amazing. Also the transition from desktop based evaluation looking at accuracy scores to actual deployment can shine a light on hidden limitations of a model. Lastly, if you do a lot of sampling of the data you might struggle to build something really good. Don't worry though, there will be no penalties for lack of performance as long as sensible modelling decisions are being made.
- **What No Templates?** No there are no templates, but there are plenty of template notebooks from the tasks performed during the course that you can use as a starting point.

Submission

The key submission details for the assignment are as follows:

- **Submission date:** Sunday 18th April, 2021 before 23:30.
- **Submission method:** Submissions should be made through the module Brightspace site.
- **Submission format:** Submissions should compose a zip file containing the following:
    - Completed jupyter notebooks or python source files for each of the main tasks (image based learning, reinforcement learning)
    - .html exports of each Jupyter notebook after execution that contains all output (for Jupyter notebooks used)
    - any other files required to execute your code (e.g. saved model files) – but do not include the datasets
    - a short notebook describing your conclusions from the experiments performed.
- **Late submissions:** Late submissions will be penalised at 5% penalty per day.

Marking

Marking of tasks will be based on the following weighting.

- Task 1 45% Image based supervised learner
- Task 2 35% RL based learner
- Task 3 5% Deployed evaluation experiment
- Task 4 15% Experiment reflections