Specification Details

Hari Vidharth
s4031180

Ashwin Vaidya
s3911888

Krishnakumar Santhakumar
s4035992

Dhawal Salvi
s4107624

1 Report

Ashwin and Dhawal were responsible for writing and editing the report. Everyone contributed in writing the discussion and conclusion.

2 Code

Hari - Responsible for developing and training the code for DDQN algorithm .
Krishna - Responsible for developing and training the code for PPO algorithm.
Ashwin - Responsible for creating the Unity Environments (for Linux and Windows) and interfacing the unity and python using openAI gym wrapper.

3 Experimentation and Figures

Hari - Was responsible for running experiments and generating results and figures for DDQN algorithm.
Krishna - Was responsible for running experiments and generating results and figures for PPO algorithm.

4 Code References

We used openAI gym wrapper to integrate the unity environments to the python gym library. The working of PPO algorithm was studied using the paper published by Open AI team (John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov - OpenAI, source : https://arxiv.org/pdf/1707.06347.pdf). In implementing the algorithms, we used tensorflow keras python library. To test the performance of the algorithm in different environments we used gym library developed by openAI and unity environments.

4.1 DDQN Implementation

The github repository (source: https://github.com/philtabor/Deep-Q-Learning-Paper-To-Code/tree/master/DDQN.) along with its related YouTube tutorials was used for understanding and implementing the DDQN algorithm, the original implementation was in pytorch for Atari games, the current implementation was implemented in Keras for unity and gym environment.

4.2 PPO Implementation

The Idea behind the Advantage Actor Critic, was studied with the help of course (Deep Reinforcement Learning) by Thomas Simonini in freecode camp (source : https://www.freecodecamp.org/news/an-intro-to-advantage-actor-critic-methods-lets-play-sonic-the-hedgehog-86d6240171d/).

To get an overview of implantation of PPO in keras, I followed the tutorial on Medium by Chintan Trivedi (Proximal Policy Optimization Tutorial) (source: https://towardsdatascience.com/proximal-policy-optimization-tutorial-part-1-actor-critic-method-d53f9 afffbf6).

4.3 Unity ML-Agents

Unity's ml-agents API was used to create the environments in unity and interface it with the python code. Additionally, the initial multi-agent soccer environment was programmed in Unity (source: https://github.com/Unity-Technologies/ml-agents/).

5 Hardware details

Below are the approximate training run times for the environments, it was observed that both the GPU gave similar run times.

Unity Basic 1000 episodes run time 1 hr.
Gym Lunar Lander 1000 episodes run time 4 hrs.
Unity Push Block 1000 episodes run time 25 hrs.
Unity Hallway 1000 episodes run time 17 hrs.

5.1 DDQN

The hardware details used to train the DQQN algorithm with different environments as follows,
System used : Alienware 15 R3
GPU used : Nvidia - GTX 1060 OC 6GB.
Operating system : Windows 10.

5.2 PPO

The hardware details used to train the PPO algorithm with different environments as follows,
System used : Msi GS75.
GPU used : Nvidia - RTX 2070 MaxQ.
Operating system : Linux.