

CSE4/510: Introduction to Reinforcement Learning

Spring 2020

Instructor: Alina Vereshchaka

Assignment 2 - Value Function Approximation

Due Date: Sunday, April 5, 11:59pm

1 Assignment Overview

The goal of the assignment is to explore OpenAI Gym environments and implement value function approximation algorithms. In the first part of the project we will implement deep Q learning (DQN), following DeepMind's paper that explains how reinforcement learning algorithms can learn to play Atari from raw pixels. The purpose of this project is to understand the effectiveness of deep neural networks as well as some of the techniques used in practice to stabilize training and achieve better performance. We will train our networks on two OpenAI gym or other complex environments. In the second part of the project we will implement an improvement to the DQN algorithm, focusing on Double Deep Q-learning (DDQN) or Prioritized Experience Replay (PER).

Part 1 [80 points] - Implementing and applying DQN

1.1 Implement DQN [40 points]

Implement DQN from scratch following DeepMind's paper ([mnih2015human] and [mnih-atari-2013]). You may use Keras/Tensorflow/Pytorch.

1.2 Apply DQN to Complex Environments [20 + 20 points]

Test your DQN algorithm on any TWO OpenAI Gym environments or Google Research Football. You may also use your custom made multi-agent environment or any other complex environment, that you will use for your Final Project (this has to be confirmed with the course staff). Compare the results. Describe the environments that you used (e.g. possible actions, states, agent, goal, rewards, etc). Provide reward dynamics (average reward in t -steps).

Important notes:

- One of the environments has to be either Google Research Football or has to use CCN (Convolution Neural Network) for the state preprocessing (e.g. OpenAI Atari).
- The environment with multiple versions considers as one environment.

Suggested environments:

- OpenAI CartPole
- OpenAI LunarLander

- OpenAI Atari Breakout
- OpenAI MountainCar
- OpenAI Space Invaders
- Google Research Football

Part 2 [20 points] - Improving DQN

DQN had a lot of success in applications in various domains. Some improvements have also been done to the vanilla (DQN) algorithm. In this part we will implement one of improved algorithms that is based on DQN. Modify your DQN code from Part 1.1 to one of the improved version and apply it to two environments, that were used in Part 1.2.

Algorithm, that built on top of vanilla DQN:

- Double DQN
- Dueling DQN
- Prioritized Experience Replay (PER)

2 Deliverables

There are two parts in your submission:

2.1 Report

Report should be delivered as a pdf file, NIPS template is a suggested report structure to follow.

In your report discuss:

- Part 1
 - What is the benefit of using experience replay in DQN?
 - What is the benefit of the target network?
 - What is the benefit of representing the Q function as $\hat{q}(s, \mathbf{w})$?
 - Describe the environments that you used (e.g. possible actions, states, agent, goal, rewards, etc).
 - Show and discuss your results after applying your DQN implementation on the environments (plots may include epsilon decay, reward dynamics, etc)
- Part 2
 - What algorithm you implemented and what is the main improvement over the vanilla DQN?
 - Show and discuss your results comparing to vanilla DQN (including learning speed, reward dynamics, etc)

2.2 Code

The code of your implementations should be written in Python. You can submit multiple files, but they all need to have a clear name. All project files should be packed in a ZIP file named *YOUR_UBID_assignment2.zip* (e.g. *avereshc_assignment2.zip*). Your Jupyter notebook should be saved with the results. If you are submitting python scripts, after extracting the ZIP file and executing command `python main.py` in the first level directory, all the generated results and plots you used in your report should appear printed out in a clear manner.

3 References

- [NIPS Styles \(docx, tex\)](#)
- [GYM environments](#)
- [Google Research Football](#)
- [Human-level control through deep reinforcement learning](#)
- [Prioritized Experience Replay](#)
- [Deep Reinforcement Learning with Double Q-learning](#)
- [Lecture slides](#)

4 Submission

To submit your work, add your pdf, ipynb/python script to the zip file *YOUR_UBID_assignment2.zip* and upload it to UBlearns (Assignments section). After finishing the project, you may be asked to demonstrate it to the instructor if your results and reasoning in the report are not clear enough.

5 Important Information

This assignment is done individually. The standing policy of the Department is that all students involved in an academic integrity violation (e.g. plagiarism in any way, shape, or form) will receive an F grade for the course. Please refer to the [UB Academic Integrity Policy](#).

6 Important Dates

April 5, Sunday, 11:59pm - Assignment 2 is Due