

[CSCI-GA 3033-090] Special Topics: Deep Reinforcement Learning

Homework - 2, Deep Q Learning++

Deadline: Oct 8, 2021

Introduction

This homework is designed to follow up on the lecture about Deep Q-Learning. For this assignment, you will need to know about the basics of the deep Q learning algorithm we talked about in class. If you have not already, we propose you brush up on the lecture notes. Furthermore, you will have to learn about a couple of algorithms that we have not discussed in detail in the class: either from the original papers or from blog posts around the internet.

You are allowed to discuss this homework with your classmates. However, any work that you submit must be your own -- which means you cannot share your code, and any work that you submit with your write-up must be written by only you.

Code folder

Find the folder with the provided code in the following google drive folder:

<https://drive.google.com/drive/folders/14VehoGYvliKFJBbGKZdRquTCnIS9J4kv?usp=sharing>.

Download all the files in the same directory, and then follow the instructions in the `env_installation.md` file, and then complete the `drql.py/utils.py/config.yaml` files.

Thanks to Denis Yarats for the template for this code.

Submission

Please submit your homework using this google form link:

<https://forms.gle/H1BzdhNKT4eJWuPZ9>

Deadline for submission: **October 8th, 2021, 11:59 PM Eastern Time.**

Points

- Questions 1-4 are 5 points each.
- Bonus question: 5 points.
- Total: 20 points (max 25 with bonus).

Deep Q-learning

In the class, we learned about the Deep Q-Network (DQN) Learning algorithm, which is considered the first large scale success for any deep reinforcement algorithm. This method is quite dated now, but for a lot of algorithms used today, the roots can be traced back to DQN.

One of the algorithms improved on DQN is Rainbow (<https://arxiv.org/abs/1710.02298v1>), which combined some of its contemporary improvements over DQN into one algorithm, like dueling DQN, double DQN, and Prioritized Experience Replay. Finally, more recently, Data-regularized Q-learning (DrQ) has improved on this baseline by adding image augmentations to DQN.

In this homework, we provide you an example implementation of DQN. We ask you to add some of those improvements made in Rainbow and DrQ to get to an almost state-of-the-art deep RL algorithm.

Environment

The environment we will use in this homework is built upon the Pong, Space Invaders, and Breakout environment from OpenAI gym Atari environments (<https://gym.openai.com/envs/#atari>). In this homework, we will attempt to learn these agents from image observations. We already have a working implementation of DQN on the code folder, which you can run as `python train.py env=Breakout` and so on. Your job is to complete all the TODOs, and turn on the completed features one by one. You can download the code folder, with every file associated, from here https://drive.google.com/drive/folders/1T8B3gSNWjQU-JpifHkEDm9FfoxJA6wB_?usp=sharing

Question 1

Download the code folder, and run the code for the three given environments. Make a plot of their performance over time. This is your baseline, and you will compare your future improvements to the code with this baseline to test their validity.

Question 2

First, add [Double Q-learning](#) onto the model (find the place by searching “TODO: double Q learning” in the code files.) Make another plot by running the three environments on your code that use double Q learning.

Question 3

Next, implement Prioritized Experience Replay. In the `replay_buffer.py` file, we already have an implementation of a prioritized replay buffer. Use that in your code. Find “TODO prioritized replay buffer”, and fix the priority update for the prioritized replay buffer. Make another set of plots, and compare them to the plots from Q2. Is your performance better or worse now? Try explaining your observations.

Question 4

Finally, implement [Dueling DQN](#). Find the places where you have to put your code by searching for “TODO dueling DQN”. As before, plot your performance in the three environments. Now that you have an almost complete implementation of Rainbow, combine all of your plots together and show the improvement over vanilla DQN.

Question 5

Bonus points: We still haven’t used DrQ anywhere. Read the code to try and figure out how to use DrQ on top of DQN. You can read more in the blog post by the authors of the original DrQ paper here: <https://sites.google.com/view/data-regularized-q>. You will have to:

- Figure out what the best data augmentations are for the environment you have,
- Add those augmentations into the training process, and
- Report (improved) results from using those augmentations.

Python environment installation instructions

1. Make sure you have conda installed in your system. [Instructions link here.](#)
2. Then, get the `conda_env.yml` file, and from the same directory, run `conda env create -f conda_env.yml`.
3. Finally, run the code with `python train.py` once you have completed some of the to-do steps in the code itself.