Fall 2024 CS 747 Deep Learning
Assignment 5

Due date: November 24, 2024, 11:59 pm

Announcement

- Honor Code: Everyone needs to submit their own solution individually for this assignment. When you submit your solution, it must be your own work. Any sources must be cited if it is not from you. Otherwise you may violate the GMU honor code and GMU CS honor code.

- Please note that this assignment requires using Pytorch, and it needs to run on GPUs. You are highly recommended to use Google Colab to finish this assignment, unless you have a local machine with powerful GPUs. Since the process of running one model is time-consuming, **please start the assignment early**.

1 Deep Reinforcement Learning

In this assignment, you will implement the famous Deep Q-Network (DQN) [1] and (if you would like to) its successor Double DQN [2] on the game of Breakout using the OpenAI Gym [3]. The goal of this assignment to understand how Reinforcement Learning works using deep neural networks when interacting with the pixel-level information of an environment.

The top-level notebook (`MP5.ipynb`) will guide you through all the steps of the DQN. You will mainly implement the training of the Agent in the `agent.py` file for DQN, and, if you would like to, `agent_double.py` for double DQN. We provide you with the neural network. Do NOT change the architecture of the neural network (for consistency of grading). The file `config.py` contains most of the hyper-parameters. You might want to play around with these parameters to make your training easier.

To receive full credit on this assignment, we expect you to reach a mean score of 8 using DQN or (optionally) DDQN (as a reference, it might take more than 3000 episodes depending on the parameters).

---

[1] https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf
[2] https://arxiv.org/pdf/1509.06461.pdf
[3] https://github.com/openai/gym

Note, as you look in the ipython notebook, in our terminology, a single episode is a game played by the agent till it loses all its lives (in this case, your agent has 5 lives). In the paper, however, an episode refers to almost 30 minutes of training on the GPU and such training is not feasible for us.

We recommend that you look at the following links provided.

- The official DQN PyTorch tutorial [4]

- Official DQN paper [5]

- DQN tutorial on Medium [6]

- Double DQN tutorial on Medium [7]

We highly recommend that you understand the Official DQN Pytorch tutorial before starting this assignment. This will give you a great starting point to implement DQN and Double DQN as the tutorial implements a version of double DQN for cartpole! However, we expect you to follow our code instructions and implement code in our format. Uploading code that does not follow our format will be receive a zero.

This is a computationally expensive assignment. It is expected that your code should run for at least 4 hours to complete 2000 episodes. You can stop training early if you reach a mean score of 8 in the game.

This assignment requires a GPU, so use your Google Cloud credits (colab could work for this assignment as well).

Extra Credits

- Reach a mean score of 11 using either DQN or DDQN, and elaborate on how you achieve the score.

- Train a DQN agent for one or more additional Atari games from OpenAI gym and report on any implementation/hyperparameter changes you had to make, and your agent's performance.

- Implement policy gradient training or another advanced RL method for Breakout or another Atari game and compare performance (including convergence speed) to your DQN method. You need to write your own code from scratch, not train an off-the-shelf method.

---

[4] https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html
[5] https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf
[6] https://jonathan-hui.medium.com/rl-dqn-deep-q-network-e207751f7ae4
[7] https://medium.com/@qempsil0914/deep-q-learning-part2-double-deep-q-network-double-dqn-b8fc9212bbb2

Environment Setup

The assignment is given to you in the `MP5.ipynb` file. If you are using a local machine, ensure that ipython is installed (`https://ipython.org/install.html`). You may then navigate the assignment directory in terminal and start a local ipython server using the jupyter notebook command. Instructions to install dependencies are provided at the top of the notebook. These instructions should work for local machines, google cloud, and google colab. We have test this assignment on pytorch version 1.7, so please install this version if there are other dependency issues.

If you will be working on the assignment on a local machine then you will need a python environment set up with the appropriate packages. We suggest that you use Conda to manage python package dependencies (`https://conda.io/docs/user-guide/getting-started.html`).

Unless you have a machine with a GPU, running this assignment on your local machine will be very slow and is not recommended.

Submission Instructions

Please submit all your files on GMU Blackboard Portal before the due date.

- All of your code (python files and ipynb file) in a single ZIP file. The filename should be `netid_mp5_code.zip`.

- Your ipython notebooks with output cells converted to PDF format. The filenames should be `netid_mp5_output.pdf`.

- A brief report in PDF format using this template[8], with name `netid_assignment5_report.pdf`.

---

[8]`https://docs.google.com/document/d/1coi7TJVOHxbbD5XtXtnKMRGvYEgWDJMKY-jgKAAoDdE/edit?usp=sharing`