CS234: Reinforcement Learning

Assignment 2: Playing Pong with Deep Q Learning

Due Date: 2/10 (Sat) 11:00 PM (23:00) PST.
See course webpage for the late day policy.

This assignment will provide you with practice with Q-learning with function approximators especially the deep Q-network to play games! We will use Open AI gym for the Atari environment. To be noted, it takes more than 12 hours for Pong to train. So, please start early.

- understand the basic **Q-learning** method.
- understand the use of **experience replay** , **target network** and the selection of **hyperparameter tuning**.
- implement and apply a **linear approximator** on the Pong game.
- implement and apply the **DQN** on the Pong game.
- understand the differences and tradeoffs between these two approximators.

[Optional/Ungraded]

This question is designed to give some optional practice with implementing tabular Q learning before proceeding to DQN, the starter code is available here and the question writeup here

Tasks

There are 2 parts to this assignment: written and code components.

The coding assignment, including some starter code, is available here.

The written homework questions are available here.

If you would like to typeset your solution, we have provided our assignment2 LaTex file here. **PLEASE DO NOT REDISTRIBUTE THIS FILE WITH ANYONE OUTSIDE THE CLASS!**

Setup

Working remotely on Azure

(highly recommended for training on Pong)

As part of this course, you can use Azure for your assignments. We recommend this route for anyone who is having trouble with installation set-up, or if you would like to use better CPU/GPU resources than you may have locally. Download the setup scripts here. Please see the set-up tutorial here for more details. :)

Working locally

While you are waiting for the Azure subscriptions to be set up, you can get started all problems except q4 and q5 (Pong).

Note: Please be sure you have Python 2.7.x installed on your system. The following instructions should work on Mac or Linux. If you have any trouble getting set up, please come to office hours and the TAs will be happy to help.

[Optional] virtual environment: Once you have unzipped the starter code, you might want to create a virtual environment for the project. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed on your machine. To set up a virtual environment, run the following:

```
cd assignment2
sudo pip install virtualenv      # This may already be installed
virtualenv .env                  # Create a virtual environment
source .env/bin/activate         # Activate the virtual environment
pip install -r requirements.txt
```

```
# install gym
git clone https://github.com/openai/gym
cd gym
sudo apt-get install cmake
sudo apt-get install zlib1g-dev
sudo pip install gym[atari] # install atari
sudo apt-get install ffmpeg
# Work on the assignment for a while ...
deactivate                       # Exit the virtual environment
```

Install requirements (without a virtual environment): To install the required packages locally without setting up a virtual environment, run the following:

```
cd assignment2
pip install -r requirements.txt
# install gym
git clone https://github.com/openai/gym
cd gym
sudo apt-get install cmake
sudo apt-get install zlib1g-dev
sudo pip install gym[atari] # install atari
sudo apt-get install ffmpeg
```

Notes

NOTE 1: Please install and make sure you're using the tensorflow 1.0 version and up-to-date numpy.
NOTE 2: Please use Python 2.7.X to develop your code.
NOTE 3: Please start early!

Submitting your work

Submit both your written assignment and coding assignment by following the **Submission Instructions**.

Reference papers

- Human-level control through deep reinforcement learning, V. Mnih et al., Nature, 2015.
- Playing Atari with Deep Reinforcement Learning, V. Mnih et al., NIPS Workshop, 2013.