Project 5: AI Gym

Submit Assignment

Due May 7 by 11:59pm **Points** 20 **Submitting** a text entry box or a file upload
File Types py

This homework will give you hands-on experience applying reinforcement learning to important real world tasks: racing a car in a 2D physical environment, **or** crushing the classical Atari Space Invaders game. We will be working in the OpenAI Gym environment: **https://gym.openai.com/** **(https://gym.openai.com/)** This project has multiple steps to set up, implement, and train, so start early.

This project can be done in pairs (2 people) or individually. If working in pairs, very clearly list team members in the "comment" field when submitting, and also at the top of DeepRLAgent.py.

Note to seniors planning to graduate*:* you **must** submit your project by Wednesday May 9th, even if you have more than 2 grace days remaining (i have to compute and submit your grades).

Overview:
You have an option to either train your system to play the Atari game of Space Invaders OR to train an agent to drive a 2-D race car. Both problems are challenging in slightly different ways, but if you solve one, you can re-use the solution to solve the other with little or no change to your code.
The input to your program will be frames of a video game (Atari Space Invaders or simplified Race Car). Your program will have to pre-process this input to make it amenable for learning, and create and train a neural network model to successfully predict the best action to take at each time step to play reasonably well [defined specifically to each game]. The most challenging part of this homework will be to synthesize a large number of conceptual and programming tools required, so start early and ask questions often!

Installation and setup:

- **Optional (but strongly recommended): Anaconda**
  Anaconda distribution of python makes life much easier: **https://www.anaconda.com/download/** **(https://www.anaconda.com/download/)** *Note:* download the Python 3.6 version.

- **AI Gym Installation and Setup:**
  MacOS and Linux: Follow the very detailed instructions for installing AI Gym here:
  https://github.com/openai/gym **(https://github.com/openai/gym)**

  Windows: Follow the detailed instructions here: **https://medium.com/@SeoJaeDuk/archive-post-how-to-install-open-ai-gym-on-windows-1f5208c16179** **(https://medium.com/@SeoJaeDuk/archive-post-how-to-install-open-ai-gym-on-windows-1f5208c16179)**
  Skip the Mujoco step (not needed for your project)

PyCharm configuration: After installation, in PyCharm, set the project interpreter to the (new?) 3.6 Python binary, if not the default.

Note 1: Install at least one of the Box2D and Atari environments or both.
Note 2: Run the provided starting files to ensure correct installation (python SpaceInvadersDriver.py)

- **TensorFlow Installation and Setup:**
  Follow the detailed instructions here for all operating systems: **https://www.tensorflow.org/install/ (https://www.tensorflow.org/install/)** Note: installing Anaconda makes tensorflow much easier to install, as it requires a bunch of dependencies pre-installed with Anaconda.

What to submit:

- SpaceInvadersDriver.py (and/or RaceCarDriver.py) where you implement the "driver" code to read observations, pre-process the data, train and test your agent, and to read and write your trained model files. Skeleton driver files to be provided, these are ***not*** complete.
- DeepRLAgent.py where all your neural network definition, training and prediction code will reside.
- Model file(s) needed by your RL Agent code to test a fully trained model.

What to do:

Starting files: obtain and unzip starting examples provided **here: p5.zip** which contain the starting skeleton files for this assignment, and a fully solved CartPole balancing example and a discrete QLearning agent, to demonstrate training and testing "classical" Q-learning for AI Gym.

Problem 1 (7 pts): implement the driver file to do necessary pre-processing of the frame data [hints/suggestions can be shared in Discussion]
Here you chose to do one of two options:

- Option 1: RaceCar: **https://gym.openai.com/envs/CarRacing-v0/ (https://gym.openai.com/envs/CarRacing-v0/)**
- Option 2: Atari Space Invaders: **https://gym.openai.com/envs/SpaceInvaders-v0/ (https://gym.openai.com/envs/SpaceInvaders-v0/)**

Your driver must implement the following:
- Process command line arguments "-train" and "-test" to either train the network and save trained network to file, or test the network by loading pre-saved model from file, play the game 10 times, and record/print the score.
- Pre-process screen (pixel) input to make it amenable for learning (lots of hints available in class and on the interwebs)
- Save trained model to file after training, and load from file for testing (filename of model files up to you, just keep in same directory as your code)

Problem 2 (8 pts): Design and implement the Policy Gradient version of deep Q-learning as described in Chapter 13 of Reinforcement Learning book: **http://incompleteideas.net/book/bookdraft2018jan1.pdf** **(http://incompleteideas.net/book/bookdraft2018jan1.pdf)** and to be introduced in class. Do not be scared: the main idea is to estimate action probabilities directly instead of indirectly through Q-values. As a head start, there is a very nice and clear post by Andrej Karpathy (one of the creators of AI Gym) here: **http://karpathy.github.io/2016/05/31/rl/** **(http://karpathy.github.io/2016/05/31/rl/)**
Note: To implement the network, you must use the low-level TensorFlow API. (do *not* use higher-level libraries like Keras or Cafe).

Problem 3 (5 pts): *Train* your network on either your own computer (SLOW) or on Google Cloud Platform instance with 1 or more GPUs.
Success criteria is different for each option, averaged over 10 test trials:
SpaceInvaders: Score >= X1 --> 3pts; Score >= X2: 6pts; Score >= X3: 8pts (point values TBD)
RaceCar: Score >= Y1: 3pts; Score >= Y2: 6pts; Score >=Y3: 8pts (point values TBD)

Extra Credit (up to +5pts): if your main task was SpaceInvaders, also train your network to play Breakout: **https://gym.openai.com/envs/Breakout-ram-v0/** **(https://gym.openai.com/envs/Breakout-ram-v0/)**
If your main task was RaceCar, also train your network to walk a 2-D robot: **https://gym.openai.com/envs/BipedalWalker-v2/** **(https://gym.openai.com/envs/BipedalWalker-v2/)**

Instructions on how to create a GCP instance: here: **GCP resources**

Good luck, have fun, and post lots of questions on Discussions board!