CMPS 4660/6660 Reinforcement Learning – Fall 20

Lab 2 (30 points)

Due 11/24/20 before class

In this lab, you will solve LunarLander-v2 or LunarLanderContinuous-v2 (both are available in OpenAI Gym), using model-free reinforcement learning. You may pick one of them for this lab.

Here is a brief description of LunarLander-v2. The lander maneuvers by engaging thrusters (with a noisy outcome) and consuming fuel. The state has 8 components: horizontal and vertical position, horizontal and vertical velocity, angle and angular velocity, and left and right leg contact. There are four actions available: do nothing, fire main engine (push up), fire left engine (push right), and fire right engine (push left). Vehicle starts from the top of the screen (with random initial velocity) and landing pad is always at coordinates (0,0). Each simulation episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Firing side engine is -0.03 points each frame. Solved is 200 points.

The model for LunarLanderContinuous-v2 is similar, except that the action space is continuous. Action is two real values vector from -1 to +1. First controls main engine, -1..0 off, 0..+1 throttle from 50% to 100% power. Engine can't work with less than 50% power. Second value -1.0..-0.5 fire left engine, +0.5..+1.0 fire right engine, -0.5..0.5 off.

You need to solve either LunarLander-v2 or LunarLanderContinuous-v2 by implementing two algorithms: (1) Sarsa with linear function approximation, and (2) a deep reinforcement learning algorithm you choose. You should implement both algorithms using Gym and Google Colab. You may use either TensorFlow or PyTorch for training the neural network(s) in your deep RL algorithm.

Each group should turn in one zip file including the following:

1. a README file that includes a link to your Colab notebook (make sure it is runnable in Google Colab) and a detailed description of the following design choices:
    (a) features vectors for linear approximation;
    (b) the deep RL algorithm you choose and any customization you have made;
    (c) a complete list of hyperparameters for both algorithms, e.g., step size, exploration rate, number of episodes, batch size for experience replay, neural network architecture, etc.
2. at least one figure that compares the performance of Sarsa with linear function approximation and the deep RL algorithm you choose (code for generating the figure should be included in your Colab notebook).

References:

1. OpenAI Gym: `https://gym.openai.com/`
2. LunarLander-v2: `https://gym.openai.com/envs/LunarLander-v2/`

3. LunarLanderContinuous-v2: `https://gym.openai.com/envs/LunarLanderContinuous-v2/`
4. Google Colab: `colab.research.google.com`
5. Getting Started With Google Colab: `https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c`