Due Date: 24 SEP 2021 at 11:59 PM EST

PSet4 – CS 7648

CS 7648 Interactive Robot Learning
Instructor: Nakul Gopalan

Instructions:

- You may work with one or more classmates on this assignment. However, all work must be your own, original work (i.e., no copy+pasting code). You must list all people you worked with and sources you used on the document you submit for your homework
- You should work with Python >= 3.6.12 and install the dependencies by `pip install -r requirements.txt`.
- To receive full credit on the assignment, you must complete problems 1 & 2 (the coding and the short answer). Your short answers should be in .pdf form and included in your coding zip.

Problem 1:

We would like you to provide feedback and train a Pong agent using "Training an Agent Manually via Evaluative Reinforcement" (TAMER). Specifically, we are implementing Algorithm 1 of the paper. We have provided template code in the zip folder on Canvas, where you will again need to fill in code where we have marked (`#TODO: INSERT CODE HERE`). Once again, some locations that are marked will require more than one line. The Pong environment is similar to Problem Set 2, but we have made it compatible with OpenAI gym env interface. Note that this interface is the most common used environment definition for most RL environments.

In this TAMER setup, you have two main steps:

1) Provide feedback and train your agent (`tamer.py`)
2) Test the learned policy (`play_pong.py`)

The agent is initialized randomly, and the only way you could "control" it is by giving feedback. It might not be as easy as giving a demonstration, but that is also part of the reason we want you to try it! `tamer.py` will save the policy and exit the game upon pressing the "down" arrow key. If you desire, you may add logic to exit on other conditions (such as a high `r_score` value), but this is not necessary. We have already included code to provide rewards to the agent using the number keys. The reward keys are [1, 2, 3, 4, 5, 6, 7, 8, 9, 0] and correspond to values of [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4], respectively (so the "6" key is zero reward, and the numbers go up/down from there with "1" being the lowest and "0" being the highest). The game will pause every 5 steps that the ball is on your half of the screen and coming towards you, providing you a

chance to give reward. You may wish to modify the frequency or the conditions under which you provide reward (this is marked with an optional `TODO`).

The success criteria of `play_pong.py` would be successfully return the ball three times by using your saved model. We will run the code with your model to examine.

Note that you might need to tune hyperparameters and do some feature engineering in order to make this happen. Hyperparameter tuning and feature engineering is indeed cumbersome (and arguably you won't need feature engineering if you have good-enough neural network hyperparameter tuning!), but sometimes they are essential to implement RL systems. Again, the focus of the homework is to understand and implement the algorithm, and we will give partial credits for the right, and hyperparameter tuning of the final result will only be a small portion of the grade. The environment/network you are given will work with a single feature (the difference in y-position between the ball and the paddle), but you may modify the feature space as desired (this is marked with an optional `TODO`).

To receive full credit on the coding assignment, please submit all of your code and saved model file in a zip folder.

Problem 2:

Please respond to the following questions in short-answer form. **To receive full credit on the short answers, please include a pdf of your solutions in your coding zip.**

1. What is an Advantage function? Please provide the mathematical definition and explain how it is different from a Q function and a value function.

2. What are some of the techniques we could use to handle suboptimality in demonstrations? Please list two of them and briefly explain your solutions.

3. In a few sentences each, please compare TAMER with Behavior Cloning (Pset2) and Inverse Reinforcement Learning (PSet3). Please list pros and cons for each of the method.

4. In a few sentences, compare TAMER and COACH.