CMPE58Y - Robot Learning
Homework 2: Q-learning with function approximation

TA: Alper Ahmetoglu
ahmetoglu.alper@gmail.com

March 26, 2022

1 Introduction

In this homework, you will implement Q-learning with function approximation for the cart pole task in OpenAI Gym environment.

2 Function Approximation

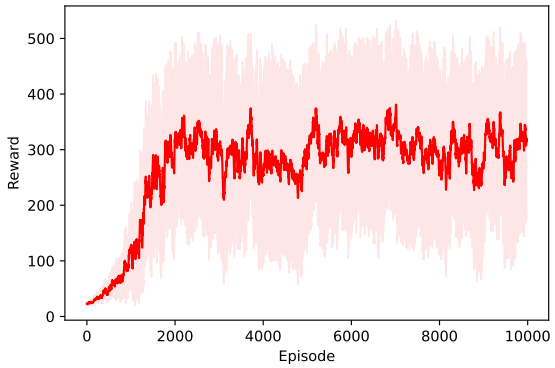
As you might have noticed in the first homework, the number of entries in the table grow very quickly when you discretize continuous-valued variables. Discretization is also a quite cumbersome work since we might not know anything about the environment beforehand (i.e., how to discretize, what is important, and so on). Alternatively, we can use a function to represent $Q(s, a)$ instead of discretizing continuous-valued variables. This allows us not to fill a possibly very large table but only learn a few parameters.

You will use a multi-layer perceptron (MLP) as the function approximator. You are given a minimal MLP implementation in python, however, its backpropagation pass is omitted. You need to implement its backward pass to calculate the gradients using backpropagation. You are free to use your own MLP implementation, instead of the given one.

One important thing about the given implementation is that the model always holds the last activations. So, if you pass the next state to estimate the target reward for the current state (i.e., $r+\gamma \max_{a'} Q(s_{next}, a')$), the model will hold the next state's activations, even though you need the current state's activations to correctly calculate the gradients. This might not be an issue depending on your implementation, just make sure you calculate the correct gradients. The following hyperparameters produced good results, though I suggest you to try different hyperparameters as well.

```
net = Network(4, 16, 2, 1)  # (MLP structure: 4 -- 16 -- 16 -- 2)
learning_rate = 0.0001
gamma = 0.99  # for the horizon
epsilon = 1.0  # decay it with 0.999 after each episode and fix it at 0.1
```

Here is the reward throughout episodes averaged over 10 runs.



Plot the reward over episodes and submit your code to ahmetoglu.alper@gmail.com. Feel free to ask any questions. Cheating will be penalized.

Deadline: 08.04.2022

Extended deadline: 15.04.2022 (-20 points)