CS 6364 Homework 7

November 18, 2021

Deadline for the first submission: **Nov-22-2019**.
All assignments **MUST** have your name, student ID, course name/number at the beginning of your documents.
Your homework **MUST** be submitted via Blackboard with file format and name convention as follows:
HW#_Name_writeup.**pdf** (for writing part)
HW#_Name_code.**zip** (for coding part)
If you have any questions, please contact me.

For the following questions, if you need a GPU to run, Google provide a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. Here is the link:
`https://colab.research.google.com/notebooks/welcome.ipynb?hl=en#scrollTo=5fCEDCU\_qrC0`

In this homework, we aim to solve the "cliff walking" problem using reinforcement learning (RL) techniques. We call "agent" who is to learn to behave intelligently, and here, agent's aim is to reach the goal. The size of the grid is 6×10 (see below). Agent starts at the leftmost cell in the bottom, that is, $(6, 1)$. The goal is the rightmost cell in the bottom (blue), that is, $(6, 10)$. All the cells between $(6, 2)$ and $(6, 9)$ is the cliff (red). If the agent enters the cliff, which means the agent falls into the cliff, then the agent will die. So the aim of the agent is to reach the goal (dark green) alive. An example of the route of an agent is shown below.

Agent can move only one cell at a time to the neighboring cell, that is, up, down, right and left, unless the agent touches the border. When the agent touches the border, the action that makes the agent cross the border is not performed but it must remain stopped at the point waiting until the next action. For example, if the agent is at (1, 3) and the action is to up, then agent remains at that point, and if the next action is to right, then it moves to (1, 4), or when the next action is down then agent moves to (2, 3).

In RL, an agent takes one state at a time chosen from predefined all possible states. In each of those states, all possible actions are given each with a probability of how likely the action will be chosen. Agent in RL behaves following its policy. Thus, RL is defined with a set of states $\mathcal{S}$ and a set of actions $\mathcal{A}$. Then, we have a table called policy in which each pair of all possible states and actions is given its probability to be occured. Starting with a random assignment of this probability at the beginning, the policy will be renewed according to the agent's experience.

We set a negative reward $r = -5$ for each transition and a discount factor $\gamma = 0.7$. Our goal is to reach the "Goal" in least number of actions. In other words, it aims to estimate the parameters θ such that the expected sum of rewards is maximized. In this work, to make it simple, the reward and discount factor are fixed and neural networks could be chosen as your favourite.

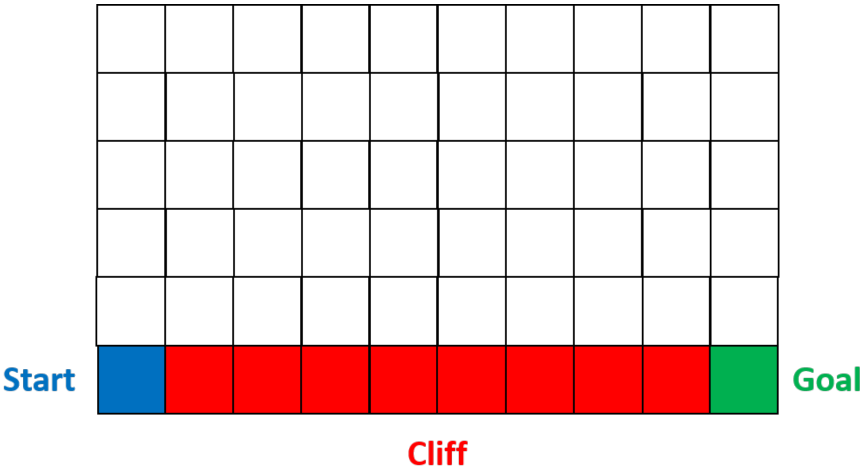
Q1 Apply REINFORCE algorithm for policy learning and implement using pytorch.

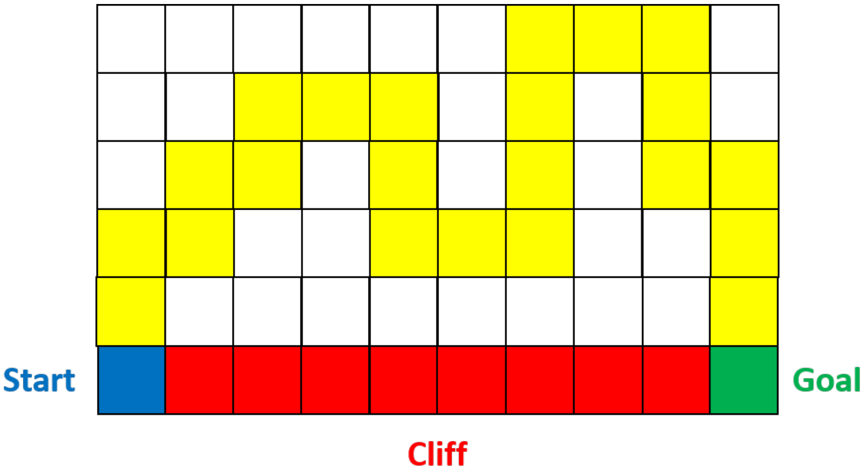
Q2 Apply Actor-Critic algorithm for policy learning and implement using pytorch.

Please note that both two questions are programming assignments. Please use **Pytorch** for the implementation. Please submit your code along with the initial and final policy tables for both questions.

If you have difficulties with this homework, the following references may help:

- Reinforcement Learning: An Introduction - Chapter 13: Policy Gradient Methods
- `https://github.com/dennybritz/reinforcement-learning/tree/master/PolicyGradient`
- My notes in class.


Start
Goal
Cliff


Start
Goal
Cliff