INTELLIGENT SYSTEMS & ROBOTICS
ASSIGNMENT 2

shreenidhi.bharadwaj@northwestern.edu | ChristopherFiore2015@u.northwestern.edu

Objective

- Implement an agent that interacts with the environment via states, actions and rewards
- Implement Model-Based RL: Policy and Value Iteration using Dynamic Programming
- Implement Deep-Q Learning & Double-Q Learning Algorithms

Submissions (1 Submission/Team)

- Presentation Deck
- Python Notebook or Collab Notebook with answers to the questions below including the programming assignment.

Question 1: (Chapter 1: Exercise 1.1: Self-Play) {5 Points}

Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself, with both sides learning. What do you think would happen in this case? Would it learn a different policy for selecting moves?

Question 2: (Chapter 1: Exercise 1.2: Symmetries) {5 Points}

Many tic-tac-toe positions appear different but are really the same because of symmetries. How might we amend the learning process described above to take advantage of this? In what ways would this change improve the learning process? Now think again. Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?

Question 3: (Chapter 3: Exercise 3.1) {5 Points}

Devise three example tasks of your own that fit into the MDP framework, identifying for each its states, actions, and rewards. Make the three examples as different from each other as possible. The framework is abstract and flexible and can be applied in many different ways. Stretch its limits in some way in at least one of your examples.

Question 4: (Chapter 3: Exercise 3.3) {5 Points}

Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out - say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in - say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of where to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

Question 5: (Chapter 7: Exercise 7.3) {5 Points}

Why do you think a larger random walk task (19 states instead of 5) was used in the examples of this chapter? Would a smaller walk have shifted the advantage to a different value of n? How about the change in left-side outcome from 0 to -1 made in the larger walk? Do you think that made any difference in the best value of n? Why do you think a larger random walk task (19 states instead of 5) was used in the examples of this chapter? Would a smaller walk have shifted the advantage to a different value of n? How about the change in left-side outcome from 0 to -1 made in the larger walk? Do you think that made any difference in the best value of n?

Question 6 (Gambler's Problem - Programming) {15 Points}

{Reference: Example 4.3: Gambler's Problem, Sutton Page 84}

Implement value iteration for the gambler's problem and solve it for $P_h = 0.25$ and $P_h = 0.55$. In programming, you may find it convenient to introduce two dummy states corresponding to termination with capital of 0 and 100, giving them values of 0 and 1 respectively. Show your results graphically, as in Figure 4.3.

Are your results stable as $\theta \rightarrow 0$?

Question 7 (Deep-Q & Double-Q Learning for Atari Games) {30 Points}

Sample code attached as part of this assignment within Exercises.zip.

Question 8 {30 Points}

Identify a commercial use case where a Reinforcement Learning can be used. Programmatically, implement the solution and provide recommendations for management.