Assignment 1 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. Which among the following are features of a reinforcement learning solution to a learning problem?

   (a) trial and error approach to learning

   (b) exploration versus exploitation dilemma

   (c) learning based on rewards

   (d) absence of any feedback or supervision

   Sol. (a), (b) & (c)
   The last option is not correct, since rewards act as feedback/supervision for RL agents.

2. Modelling a living organism as an RL agent, the environment encompasses

   (a) everything external to the organism

   (b) some portions internal to the organism as well

   Sol. (b)
   In living organisms, reward signals are generated within the brain. In the idealised RL framework this is part of the environment.

3. It was mentioned in the lecture that reinforcement learning solutions have to be able to handle delayed rewards. What exactly is/are the problems that arises due to delayed rewards?

   (a) problem in storing rewards

   (b) problem in assigning rewards to the actions that caused them

   (c) problem in mapping states to the rewards that can be achieved from those states

   Sol. (b) & (c)
   Different techniques can be used to maintain reward information, so the first option is not really an issue. The next two options do pose challenges to an RL agent. Remember that rewards, in general, depend both on the chosen action and the state in which that action was taken.

4. In the tic-tac-toe problem, it can be observed that many board positions (states) are symmetric. Assuming an imperfect opponent with a stationary policy/strategy who is also taking advantage of symmetries in the board positions, which of the following statements are true, if we take advantage of symmetries by maintaining a single (probability) value for all symmetric states?

(a) time taken to learn the values reduces

(b) space required to store the values reduces

(c) the final policy learned in this approach is better than the policy learned when not taking advantage of symmetries

Sol. (a) & (b)
Option (b) is obviously true. Option (a) is true because for each value, the experience generated from all corresponding symmetric states are being used to update that value, and hence we can expect faster convergence (when compared to the case where symmetries are being ignored). Option (c) is false because in both cases, given an opponent with a stationary policy, we expect to arrive at the same policy, the difference is only in the time it takes.

5. Considering again the question of taking advantage of symmetric board positions, is it true, that symmetrically equivalent positions should necessarily have the same value?

   (a) no

   (b) yes

   Sol. (a)
   Consider the case where the opponent does not make use of symmetries. Assuming an imperfect opponent, we can consider two states a and b which are symmetrically equivalent. Now, assume that the actions taken by the opponent when in these two states is not symmetrically equivalent. For example, in state a, the opponent may play a sound move, but in state b, the opponent may make a mistake that can be exploited by us. Thus, to be able to best respond to the opponents imperfection as in the above case, it would be beneficial for us to maintain separate values for the states a and b even though they are symmetric.

6. This question is related to the one discussed in class. Recall the temporal difference learning approach to the tic-tac-toe problem. Suppose that the probability of winning at a particular state is 0.6, the max probability value in the next set of states is 0.8, and based on our exploration policy, we choose a next state which has probability value 0.4. Should you backup the current state's probability value based on this choice of next state (i.e., move probability value 0.6 closer to 0.4) or not, given that the agent never stops exploring (i.e., the agent always makes an exploratory move some fraction of the time)?

   (a) backup the value

   (b) do not backup the value

   Sol. (a)
   There are valid arguments for both backing up, as well as not backing up values on exploration steps. The argument in favour of backing up is that by backing up values, even on exploration steps, the value function (i.e., the function specifying the value at each state - here, the probability of winning) actually represents the behaviour you are following (because according to your behaviour, you do explore some fraction of the times). The argument against backing up is that exploration is performed only for learning purposes, and that if we were to actually go by our learned policy, say when the agent is deployed in a competition, we will perform only exploitative steps to maximise reward. In this sense, we should only backup values on greedy moves.

Now, in the question, we are given that the agent never stops exploring. So, if we choose not to backup the values during exploration steps, we are actually learning a value function (and hence, policy), which is different from the actual policy the agent will follow. Thus, it makes sense to backup the values during all steps.

7. Suppose we want an RL agent to learn to play the game of golf. For training purposes, we make use of a golf simulator program. Assume that the original reward distribution gives a reward of +10 when the golf ball is hit into the hole and -1 for all other transitions. To aide the agents learning process, we propose to give an additional reward of +3 whenever the ball is within a 1 metre radius of the hole. Is this additional reward a good idea or not? Why?

   (a) yes, the additional reward will help speed-up learning
   (b) yes, getting the ball to within a metre of the hole is like a sub-goal and hence, should be rewarded
   (c) no, the additional reward may actually hinder learning
   (d) no, it violates the idea that a goal must be outside the agents direct control.

   Sol. (c)
   In this specific case, the additional reward will be detrimental to the learning process, as the agent will learn to accumulate rewards by keeping the ball within the 1 metre radius circle and not actually hitting the ball in the hole. This example highlights the importance of being cautious in setting up the reward function in order to prevent unintended consequences.

8. Suppose that we are given a 10 armed bandit problem and are also told that the rewards for each arm are deterministic. How many times should each arm be tried before we can identify the best arm with 95% probability?

   (a) 1
   (b) 2
   (c) 95
   (d) none of the above

   Sol. (a)
   Since we are told that the rewards are deterministic, it is sufficient to pull each arm once and then subsequently pick the arm for which we observed the maximum reward. Note that this is only possible because the rewards were deterministic (i.e., non-stochastic) and also because we knew before hand that the rewards were deterministic.

9. Suppose that you have been given a number of different drug formulations to treat a particular disease and you job is to identify one among them that best meets certain criteria with regards to its efficacy in treating the disease. Before you run the experiments, you need to provision for the samples that would be required. Treating this as a multi-armed bandit problem, which kind of solution method would you prefer for identifying the best option?

   (a) asymptotic correctness
   (b) regret optimality
   (c) PAC optimality

Sol. (c)
A PAC optimal solution allows us estimate the number of samples that are required to achieve the required level of performance for the chosen arm.

10. Suppose we have a 10-armed bandit problem where the rewards for each of the 10 arms is deterministic and in the range (0, 10). Which among the following methods will allow us to accumulate maximum reward in the long term?

    (a) ϵ-greedy with $\epsilon = 0.1$

    (b) ϵ-greedy with $\epsilon = 0.01$

    (c) greedy with initial reward estimates set to 0

    (d) greedy with initial reward estimates set to 10

    Sol. (d)
    Since the rewards are deterministic, we only need to select each arm once to identify an optimal arm. The greedy method with initial reward higher than all possible rewards ensures that each arm is selected at least once, since on selecting any arm, the resultant reward estimate will necessarily be lower than the initial estimates of the other arms. Once each arm has been selected, the greedy method will settle on the arm with the maximum reward.