HW 5 CS391L Reinforcement Learning

Avani Agarwal, EID: aa88539
Department of Computer Science
The University of Texas at Austin
avaniagarwal@utexas.edu

1 OVERVIEW

Reinforcement algorithms are used to train an agent to learn an optimal policy by making it directly interact with the environment and giving a reward or penalty for actions performed by the agent while iterating with the environment. For a model free RL like Q-learning the agents learn about the environment by improving their policies when interacting with the environment. Reinforcement Learning (RL) algorithms can be used to train agents to behave in a particular fashion to maximise the rewards for a given environment.

In this assignment the aim was to design an agent that collects litter along a sidewalk while avoiding obstacles. I used the Q-learning algorithm for to achieve this goal.

In general a Q-learning algorithm can be described as below

Q-learning: Learn function $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$

Require:
States $\mathcal{X} = \{1, \ldots, n_x\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}$, $A : \mathcal{X} \Rightarrow \mathcal{A}$
Reward function $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
procedure QLearning($\mathcal{X}$, A, R, T, α, γ)
  Initialize $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ arbitrarily
  while Q is not converged **do**
    Start in state $s \in \mathcal{X}$
    while s is not terminal **do**
      Calculate π according to Q and exploration strategy (e.g. $\pi(x) \leftarrow \arg\max_a Q(x, a)$)
      $a \leftarrow \pi(s)$
      $r \leftarrow R(s, a)$ ▷ Receive the reward
      $s' \leftarrow T(s, a)$ ▷ Receive the new state
      $Q(s', a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$
      $s \leftarrow s'$
  return Q

2 METHODS

In this assignment we take a sidewalk of size (W x L) where the width and length of the sidewalk is 6 and 25. We initialize 8 to have two rows representing paths that are not sidewalks. The position of the agent can be described by the x and y co-ordinates of the sidewalk. The agent starts from the top left side of the sidewalk and must move towards the right hand side.

The agent is not allowed to go backwards. The agent may move up, down, left , right or sideways and by doing so it may encounter litter for which it is rewarded or an obstacle for which it is penalized. The objective is to reach the end of the side walk. All the cells in the last column that is $(i, 24)$ where i is in range $(1, 6)$ means agent has reached the end of the sidewalk. There are 4 modules that are implemented in this assignment. Later all four modules will be linearly combined. The four modules are as following:

(1) Sidewalk Module: Staying on the sidewalk
(2) Obstacle Module:Avoiding obstacles (negative reward)
(3) Litter Module:Collecting litter along the way (positive reward)
(4) Forth Module: Reaching the end of the sidewalk

To achieve the above objectives we use Q-learning algorithm for the agent to learn the optimal policy. All four modules are trained individually first. Each module is concerned with achieving its objective. They are later cooperatively combined together to achieve all the four objectives in the most optimal way.

The Q -learning policy at each step for a module i can be described as below for a Q_i matrix of shape width x length x number of possible actions which are equal to five for this agent.

$$a(s) = max(Q_i(s, a))$$

The above equation basically maps a state-action pair given by (s, a) to a number that is the reward for performing an action at a given state. The Q-matrix is then updated according to the Bellman temporal difference learning equation that is given by:

$$\underbrace{\text{New}Q(s,a)}_{\text{New Q-Value}} = Q(s,a) + \alpha \underbrace{[R(s,a)}_{\text{Reward}} + \gamma \overbrace{\max Q'(s',a')}^{\text{Maximum predicted reward, given new state and all possible actions}} - Q(s,a)]$$

New Q-Value

Discount rate

$\alpha and \mu$ represent the learning rate and the discount factor which is responsible for determining the usefulness of future rewards for module i. The learning rate is used to determine the information change rate of the policy and as it approaches 1 the new information replaces the old Q values for a given (s, a). If the learning rate is close to zero new information has negligible effect on old $Qvalues$.

Learning rate of 0.9 and discount factor of 0.5 was used. The probability of obstacles was taken to be 20 percent. The first 20 episodes had actions that were performed by choosing from a scaled distribution so that the agent can explore the space and not miss the optimal paths. The modules were then cooperatively combined by using the best average policy that is given as:

$$a(s) = max(mean_i(Q_i(s, a)))$$

2.1 Sidewalk Module

To ensure that the agent remains within the sidewalk it's rewarded +1 to stay on the sidewalk and -1 if it is not on the sidewalk that is walking in the first and last row.

2.2 Obstacle Module

The agent is penalised with an euclidean distance based reward function between 1 and 0 if there is an obstacle in its surroundings or the window size of three into three. If the agent lands on the

obstacle it is penalised with -10. I experimented with -1,-10 and -100.

2.3 Litter Module

Litter found within the window size was rewarded with an euclidean distance based reward function between 0 and 1.If the agent landed on a cell with litter on it, the agent would be rewarded +3.

2.4 Forth Module

The agent is not allowed to go backwards. The agent may move up, down, left , right or sideways and by doing so it may encounter litter for which it is rewarded or an obstacle for which it is penalized. The objective is to reach the end of the sidewalk. All the cells in the last column that is $(i, 24)$ where i is in range $(1, 6)$ means the agent has reached the end of the sidewalk. The agent is given a reward of +10 for reaching the end of the sidewalk and the penalty of -100 for moving backwards. I also experimented with $x^+ - x$ as a penalty for all other cases that do not terminate with the sidewalk to ensure that moving forward diagonally is rewarded in the same way as moving forward in a straight line.

3 RESULTS

While individual training both the litter and obstacle module perform better after 30 episodes. By 100 episode they are close to achieving their objectives. The forth module does not discriminates between moving forward or sideways and ensures the path is completed.The sidewalk module ensures that agent does not step outside the sidewalk.The litter is in mustard yellow, obstacles in light blue, termination in yellow, path in red and not sidewalk in black.

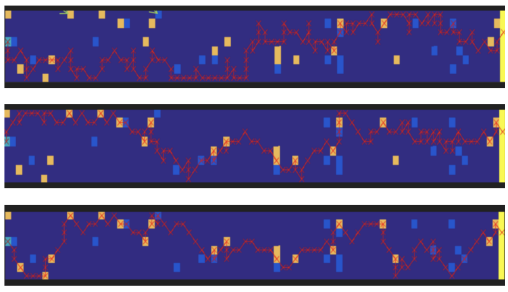

Figure 1: Litter module for first, tenth and hundredth episode.

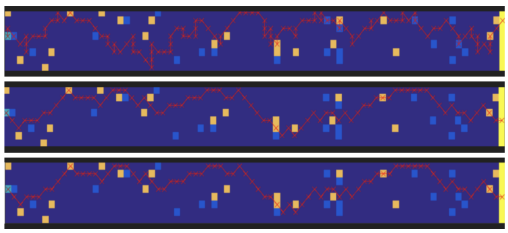

Figure 2: Forth module for first, tenth and hundredth episode.

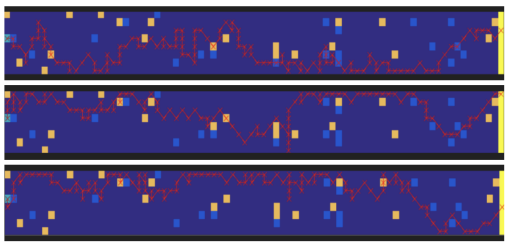

Figure 3: Obstacle for first, tenth and hundredth episode.

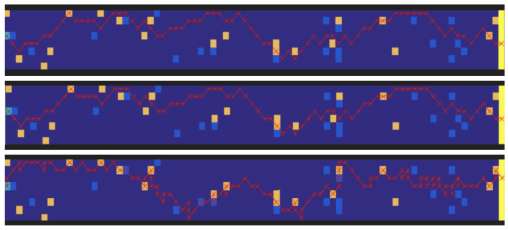

Figure 4: Sidewalk for first, tenth and hundredth episode.

In the end the integration of modules is performed. Within 10 episodes the reward per step is optimised and a more direct path is taken by the agent.

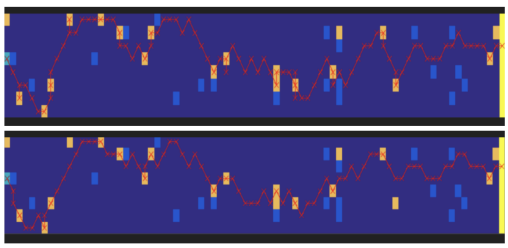

Figure 5: Integrated for first, tenth and hundredth episode.

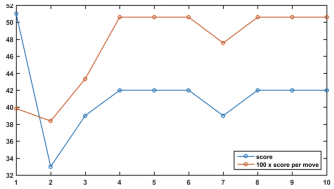

Figure 6: The score and score per step for integrated sidewalk modules.For 10 cooperative episodes the performance levels out.

4 CONCLUSIONS

Q-learning works for training an agent to walk on a sidewalk to pick litter while avoiding obstacles with setting proper learning parameters and approaching the problem as a model free RL problem but with finite number of possible actions that an agent can take in an environment. The different modules help to divide the problem into sub parts which can later be combined together to form an effective solution.

5 ACKNOWLEDGEMENT

Thanks to Prof. Dana Ballard and Jirau Song (TA) for their ideas. I would also like to add that section 5 has been taken from lecture 21 notes for Machine learning taught at CMU.

REFERENCES