Defining Solving RL Environments

Vamsi Krishna Velivela
University at Buffalo
UB No:- 50318486
vvelivel@buffalo.edu

1 Deterministic and Stochastic Environments

Set of Actions :- Left,Right,Down,UP
States :- 36
Rewards :- 3 set of Rewards
Destination state (5,5):- 100
state (2,2) :- -3
state (1,4) :- -4
Rest all the states have reward of -1
Objective :- The main Objective is to reach the end corner position (5,5)

2 Visualizations

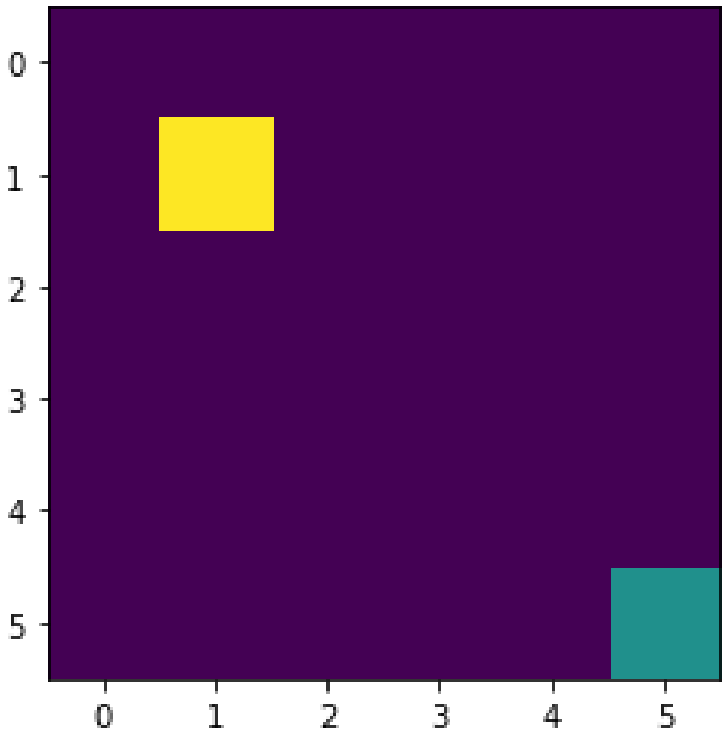


Figure 1: Deterministic Environment after moving agent from 0,0 to 1,1

3 Stochastic Environment Implementation

For all the four actions

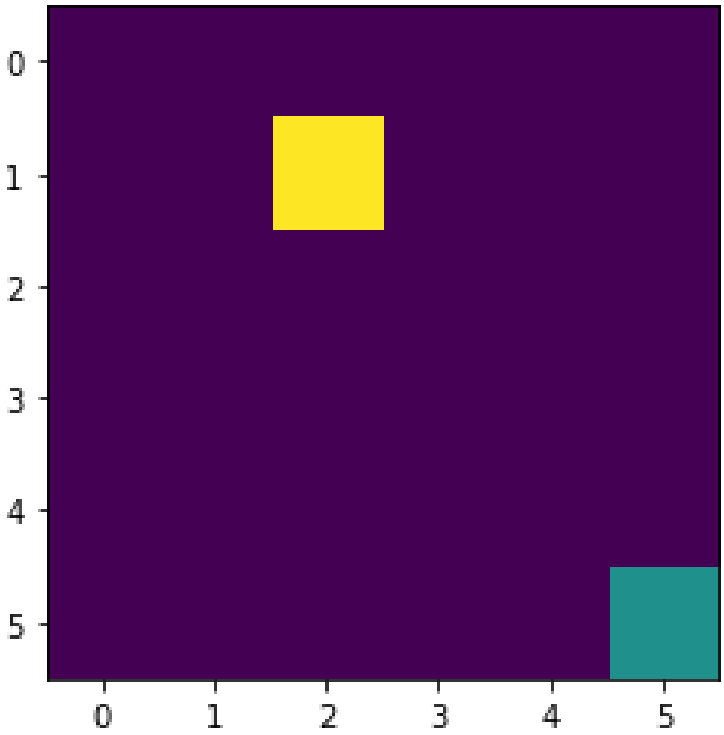


Figure 2: Stochastic Environment after moving agent from 0,0 to 1,2

- **Left** :- The agent goes to Left with a probability of 0.2 and stays back in it's own position with a probability of 0.8 with in bounds
- **Right** :- The agent goes to Right with a probability of 0.8 and stays back in it's own position with a probability of 0.2 with in bounds
- **Down** :- The agent goes to Down with a probability of 0.8 and stays back in it's own position with a probability of 0.2 with in bounds
- **Up** :- The agent goes to Up with a probability of 0.2 and stays back in it's own position with a probability of 0.8 with in bounds

4 Deterministic vs Stochastic Environments

4.1 Deterministic Policy

In deterministic policy, the action is mapped to the state. The agent follows the action and comes to a state without considering probability. In a way, the actions take deterministic outcomes without any uncertainty. π(s) = a

4.2 Stochastic Policy

In Stochastic policy, the action results to many possible states. The agent follows the action and comes to a state basing on the probability of action. π(a—s) = Pπ[A = a and S = s]

5 Q-Learning Algorithm

Q-learning is a model-free reinforcement learning algorithm. It updates the values of function basing on Bellman's Equation but not through a variable policy.update function

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$$

6 Temporal Difference(TD(0)) Algorithm

TD(0) is a policy based algorithm and model free algorithm. It's a combination of Monte Carlo and Dynamic Programming algorithms. The main advantage of TD(0) is, it need not reach the terminal stage for evaluation. Update Function

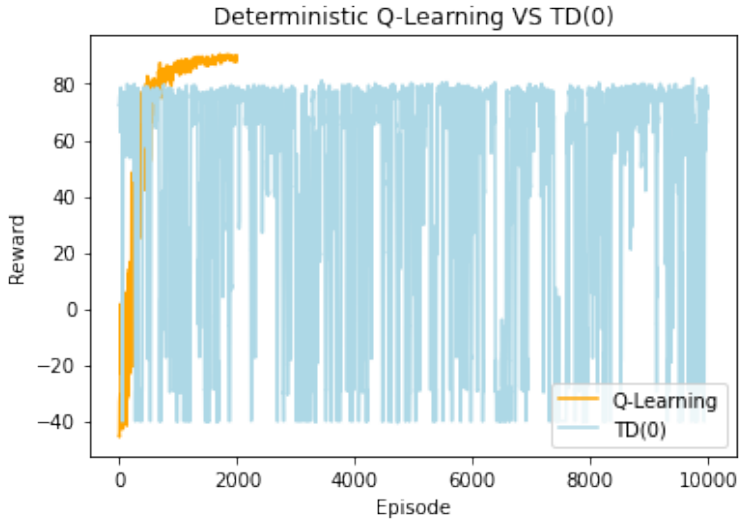
$$V(s_t) \leftarrow V(s_t) + \alpha\left[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\right]$$

7 Results

The entire environment is implemented with a learning rate of 0.6 and discount factor of 0.9

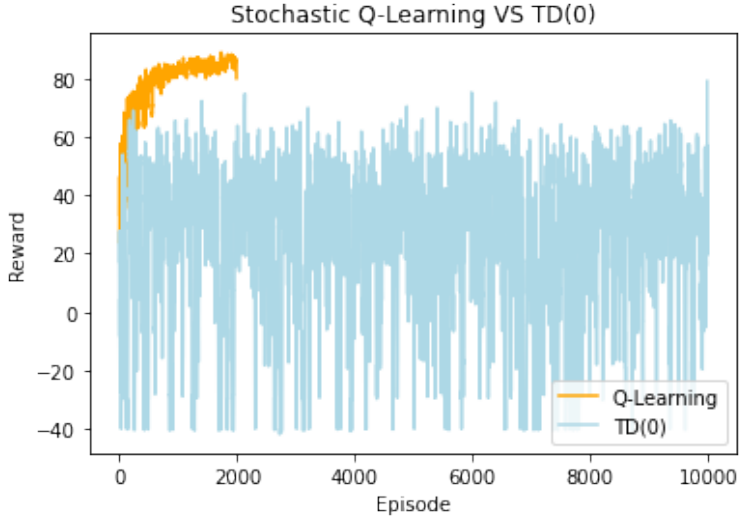
7.1 Deterministic Comparison

We ran the Q-Learning algorithm for 2000 episodes and TD(0) algorithm for 10000 episodes on same deterministic environment. We got better results for Q-Learning algorithm



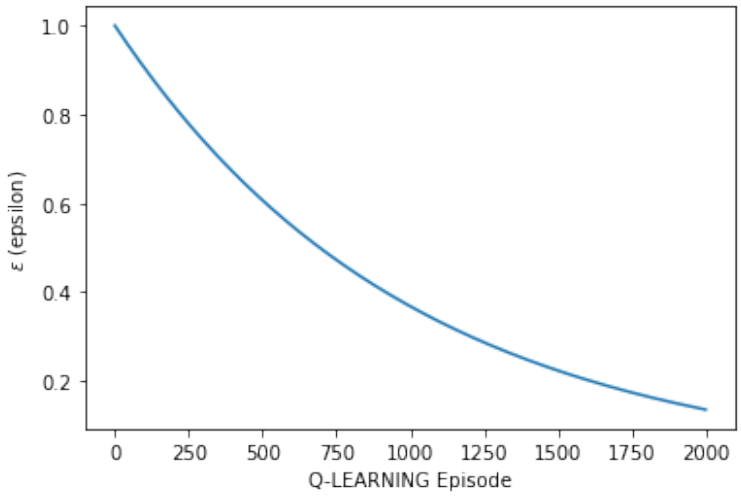
7.2 Stochastic Comparison

We ran the Q-Learning algorithm for 2000 episodes and TD(0) algorithm for 10000 episodes on same deterministic environment. We got better results for Q-Learning algorithm



7.3 Epsilon

The epsilon decay in Q-Learning algorithm in both Stochastic and Deterministic Environments. For each episode, epsilon is multiplied with a factor of 0.999. Hence, it gave the Q-Learning agent to explore in the beginning and exploit in the end.



8 Safety in AI:

The agent may get stuck in an continuous loop. To avoid that, we keep track of the number of steps taken and end the process when a max limit of number of steps is reached. In this way, we are ensuring the safety of our system.