Assignment 1 - Defining & Solving RL Environments

Indra teja Pidathala

September 30, 2022

Changes

The changes from Part-1(Report) are as follows:

1. Rewards are changed and redefined in the below sections.
2. Action indices are exchanged and are as follows:
    - previously : 0 - LEFT, 1 - UP, 2 - RIGHT, 3 - DOWN
    - currently : 0 - LEFT, 1 - RIGHT , 2 - UP, 3 - DOWN

1 Introduction

Here we have our Agent , Agent-ZERO-ZERO-VISION. He is dying due to the lack of vibranium on earth and the only way he can survive is by wielding the infinity gauntlet(target). At last he is at the grid where the gauntlet is located. but here is the catch, he doesn't have much energy left(-ve -1 reward) and needs to get to the gauntlet as soon as possible as with every move he makes his skin will get peeled, so he needs to move as little as possible and on top of it, he has thanos and his sidekick ebony-maw(-ve rewards) waiting for him in the grid. But don't worry He's got his friends wanda ironman (+ve rewards) with him inside and they can share a little energy of theirs but first agent has to meet them.

So How can Vision survive this impending doom? he remembered that he has a friend in college who took RL 546 and has good knowledge about solving these kind of tricky situations(Grid Problems). he called his friend which is 'us' and explained about the Grid Environment. so yeah let's go ahead and find the optimal path for our friend vision using Q-Learning SARSA so that he can live today and fight tomorrow.

2 Definition

Our both Environments **DETERMINISTIC** and **STOCHASTIC** have the **SAME set of STATES, REWARDS and ACTIONS** which are provided below. the only **difference** is the policy of the agent which are described separately below.

$States \to S : \{S0, S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15\}.$

$Actions \to A : \{LEFT : 0, RIGHT : 1, NORTH : 2, SOUTH : 3\}.$

$Rewards \to R : \{-1, -1, 25, -1, -1, -1, -1, -1, -1, -10, -1, -20, 10, -1, 100\}.$

$Policy - Deterministic \to \pi_d(s) : a.$

$Policy - Stochastic \to \pi_s(a|s) : P[A = a|S = s].$

Actions in any Stochastic Environment are Probabilistic. Below table (Table-1) shows the transition probability of different outcomes against all actions in all states for our stochastic env.

ACTION	**OUTCOME MOVES PROBABILITY DIST**
LEFT	[LEFT:0.8, NOCHANGE:0.2]
NORTH	[(NORTH:0.7, SOUTH:0.3)]
RIGHT	[(RIGHT:0.9, NOCHANGE:0.1)]
SOUTH	[(SOUTH:0.5, NOCHANGE:0.25,NORTH:0.25)]

Table 1: Action vs Outcome Probability

All the states allow all the actions. the rewards for each state has been shown in the below section(env visualizations)

2.1 Env Visualizations

All the Figures shown has captions to describe what it represents. in case of a reward, the value of reward has been shown just beside the reward name.

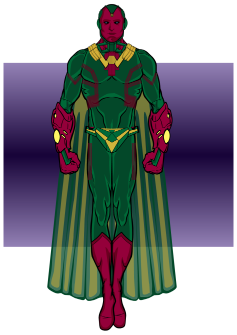


Figure 1: Agent 00Vision

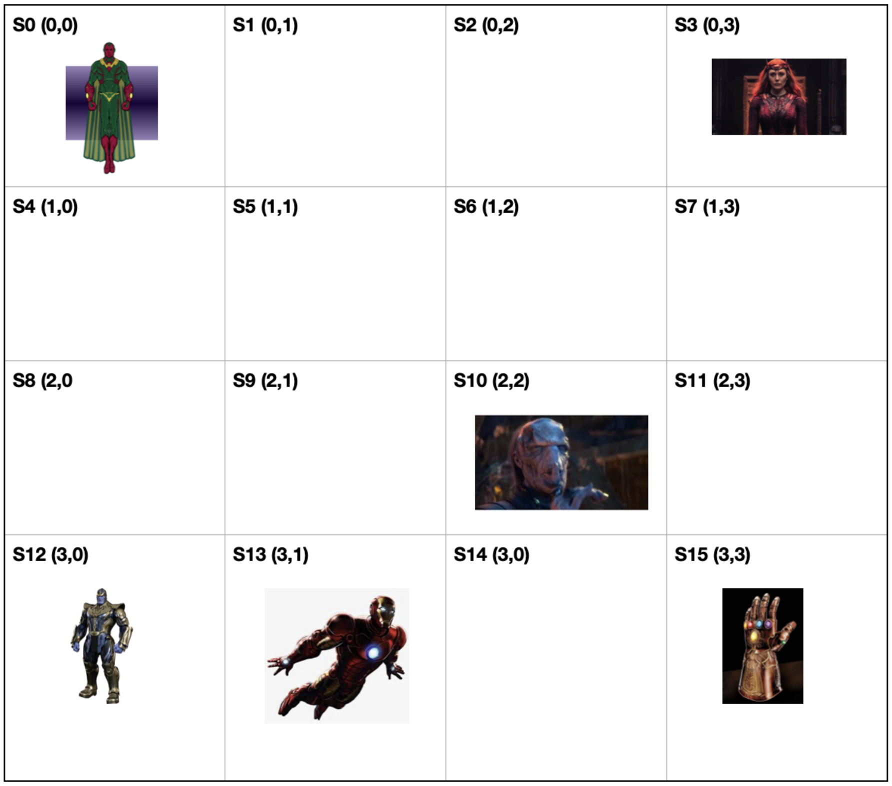
Figure 2: Reward - Wanda +25

Figure 3: Reward - Iron-man +10

Figure 4: Reward - Maw -10

Figure 5: Reward - Thanos -20

Figure 6: Reward - Gauntlet +100



S0 (0,0)	S1 (0,1)	S2 (0,2)	S3 (0,3)
S4 (1,0)	S5 (1,1)	S6 (1,2)	S7 (1,3)
S8 (2,0	S9 (2,1)	S10 (2,2)	S11 (2,3)
S12 (3,0)	S13 (3,1)	S14 (3,0)	S15 (3,3)

Figure 7: Environment - Grid - Initial - State

3 Algorithms

3.1 Q-Learning

Q-learning is a model-free, value-based, off-policy learning algorithm for RL.

- Model-free: The algorithm that estimates its optimal policy without the need for any transition or reward functions from the environment.
- Value-based: Q learning updates its value functions based on equations, (say Bellman equation) rather than estimating the value function with a greedy policy.
- Off-policy: The function learns from its own actions and doesn't depend on the current policy

Q-Learning poses an idea of assessing the quality of an action that is taken to move to a state rather than determining the possible value of the state (value footprint) being moved to.

In this, we keep Q-Table to estimate q-values for different state action pairs based on which agent learning happens

The Bellman Equation for Q-Learning is given below:

$$Q(S, A) = Q(S, A) + \alpha[(R + \gamma(Max_a(Q(S\prime, A))) - Q(S, A)]$$

.

3.2 SARSA

SARSA is an model-free, value-based,on-policy algorithm which means that while learning the optimal policy it uses the current estimate of the optimal policy to generate the behaviour The function learns with the help of the current policy.

SARSA converges to an optimal policy as long as all state-action pairs are visited an infinite number of times and the policy converges in the limit to the greedy policy ($\epsilon = \frac{1}{t}$).

In this, we keep Q-Table to estimate q-values for different state action pairs based on which agent learning happens

The Bellman Equation for SARSA is given below:

$$Q(S, A) = Q(S, A) + \alpha[(R + \gamma(Q(S\prime, A\prime))) - Q(S, A)]$$

. the major difference between SARSA and Q-Learning is that while SARSA in on-policy(considers the agent's current action) while Q-Learning is off-policy method(considers only the greedy action)

4 Results Analyses & Visualizations

Finally, we have implemented the Q-Learning and SARSA algorithms to solve the grid and observed the following results(plots).

4.1 Q-Learning on Deterministic Env

In Q-Learning we have took 2 approaches to solve it using a random start state for all episodes and predefined start state(5 in this case) for all episodes

Following are our parameters/hyper-parameters for all of our cases

- Total No of Episodes = 1000
- Discount Factor = 0.95
- Learning Rate = 0.15
- Epsilon(inital) = 1
- Epsilon Decay Rate = 0.001
- Max Time steps = 50

Following are our Results (Descriptions of plots explain themselves)

4.1.1 Random Start State

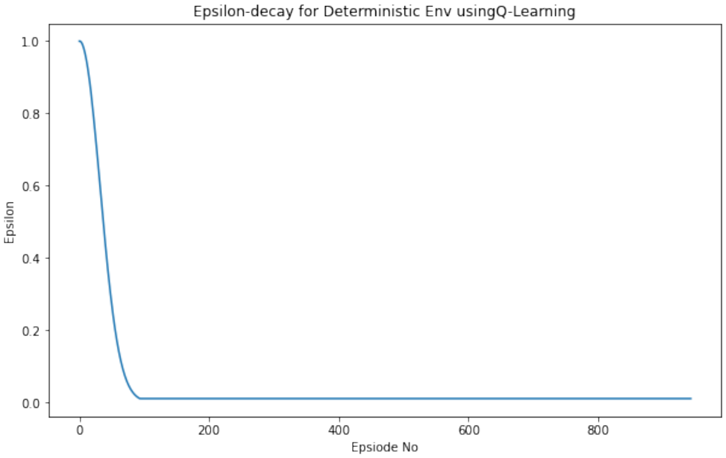


Figure 8

Analysis After training, from the plots, we observed that epsilon (Fig-8) has been decayed almost by just 100 iterations. the total reward per episode (Fig-9) has been started to converge to a range after around 50-70 iterations.

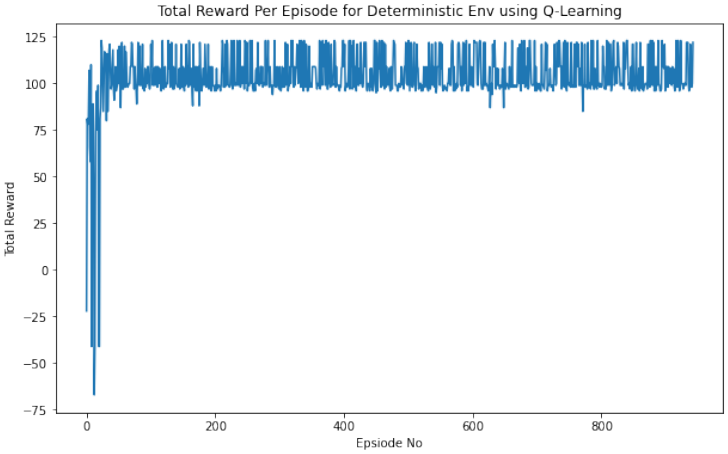


Figure 9

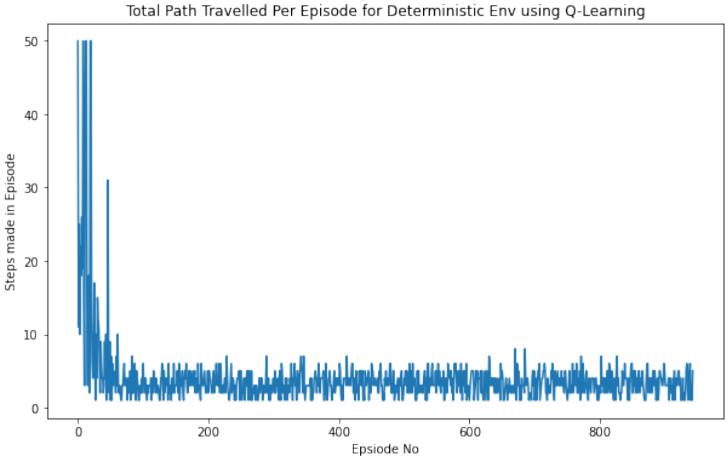


Figure 10

the fluctuations are due to the random start state we are defining before each episode. only the q-values SUM (Fig-11) has not converged which means we can still train the agent more(this can be viewed in the zeroes present in the final q-table (Fig-13)

4.1.2 Predefined Start State

Analysis : after training, from the plots, we observed that epsilon(Fig-14) has been decayed almost by just 100 iterations. the total reward per episode(Fig-15) has been been fluctuating even after all training. the fluctuations are due to the random actions being taken throughout the training(due to the stochastic

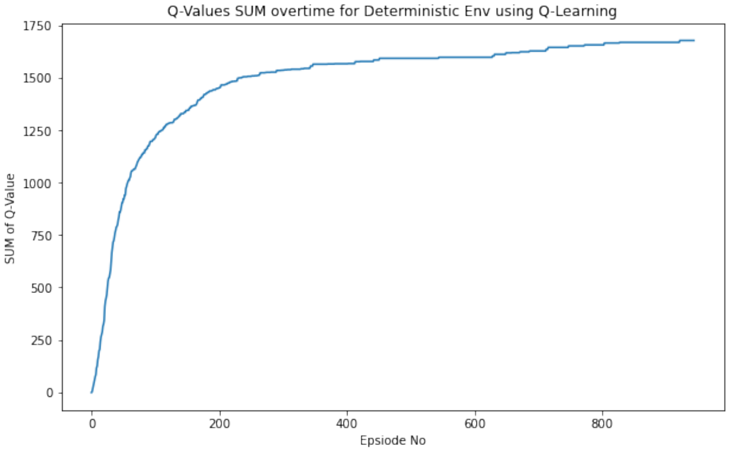


Figure 11

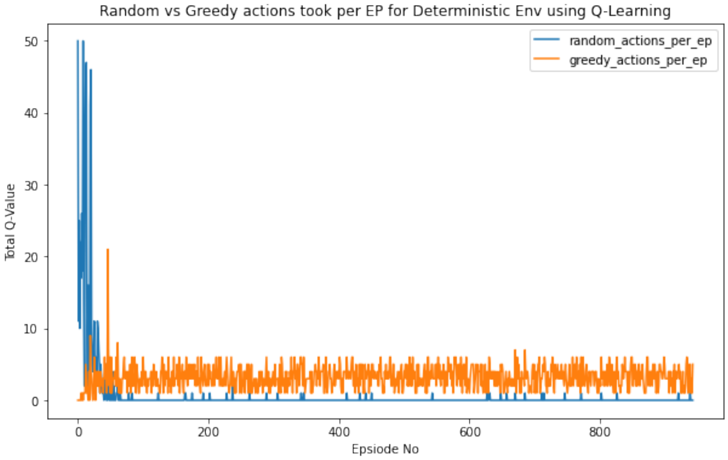
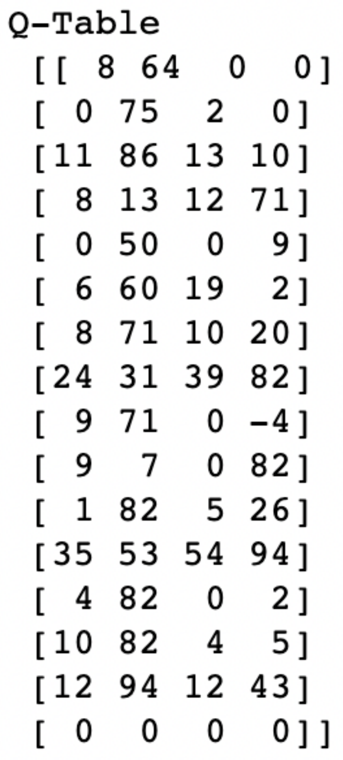


Figure 12

environment). same with other plots too(Fig-16,17,18). the q-table(Fig-19) has converged for some state,actions pairs but not all.still our agent has learnt some kind of greedy policy for some state action pairs

4.2 Q-Learning on Stochastic Env

Analysis : after training, from the above plots, we observed that epsilon(Fig-20) has been decayed almost by just 100 iterations. the total reward per episode(Fig-21) has been started to converge to a range after around 100 iterations. the fluctuations are due to the random actions being taken throughout the training(due to epsilon being a min of 0.01). only the q-values SUM has not



```
Q-Table
 [[ 8 64  0  0]
 [ 0 75  2  0]
 [11 86 13 10]
 [ 8 13 12 71]
 [ 0 50  0  9]
 [ 6 60 19  2]
 [ 8 71 10 20]
 [24 31 39 82]
 [ 9 71  0 -4]
 [ 9  7  0 82]
 [ 1 82  5 26]
 [35 53 54 94]
 [ 4 82  0  2]
 [10 82  4  5]
 [12 94 12 43]
 [ 0  0  0  0]]
```

Figure 13

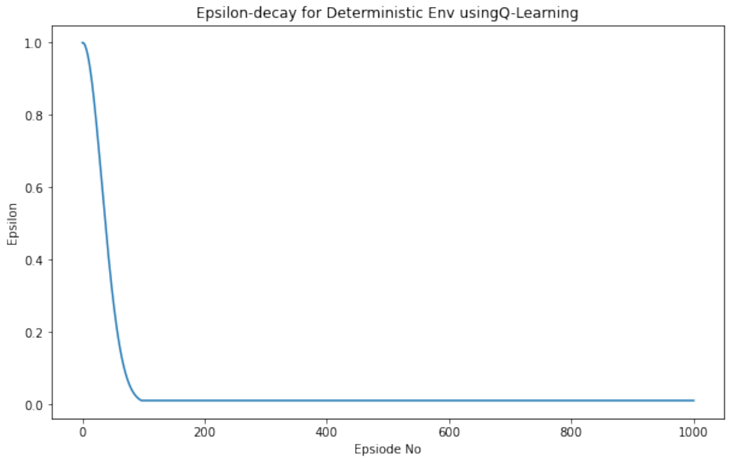


Figure 14

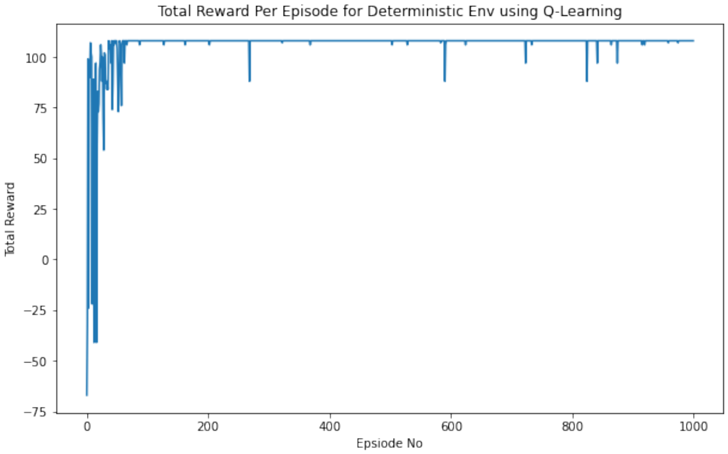


Figure 15

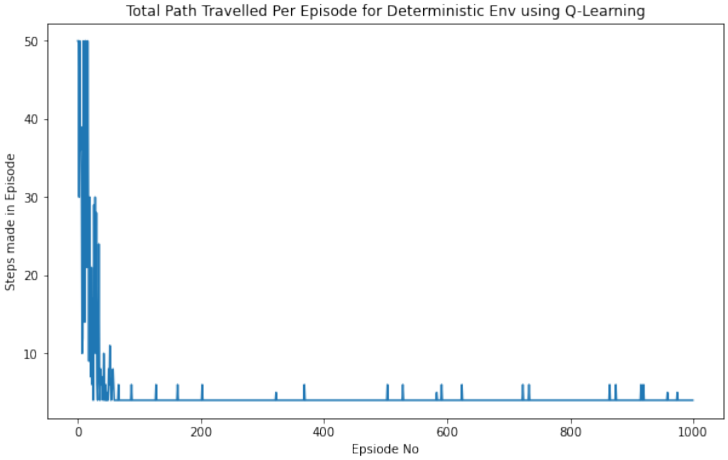


Figure 16

converged which means we can still train the agent more(this can be viewed in the zeroes present in the final q-table(Fig-25))

4.3 SARSA on Deterministic Env

Analysis : after training, from the plots, we observed that epsilon(Fig-26) has been decayed almost by just 100 iterations. the total reward per episode(Fig-27) has been started to converge to a range after around 150-170 iterations. the fluctuations are due to the random start state we are defining before each episode. only the q-values SUM((Fig-29)) has not converged which means we can still train the agent more(this can be viewed in the zeroes present in the

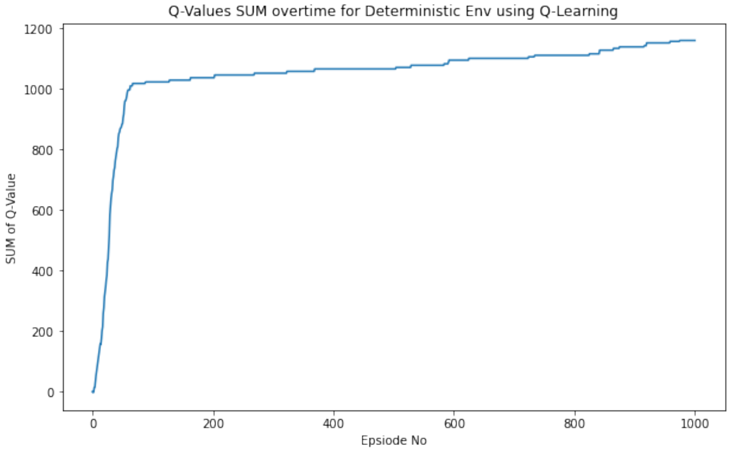


Figure 17

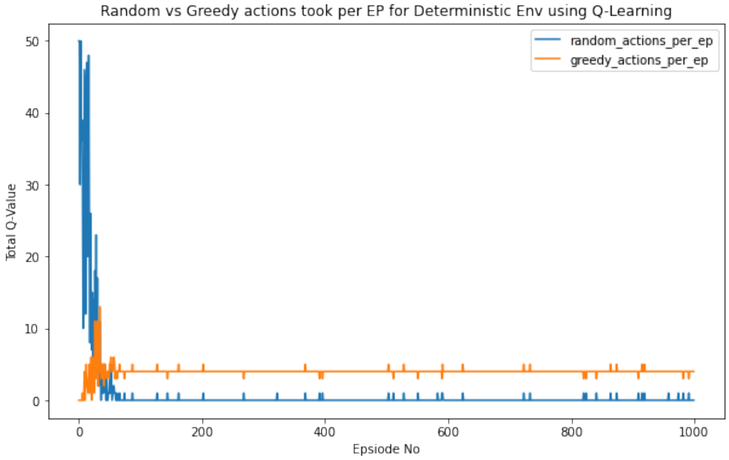
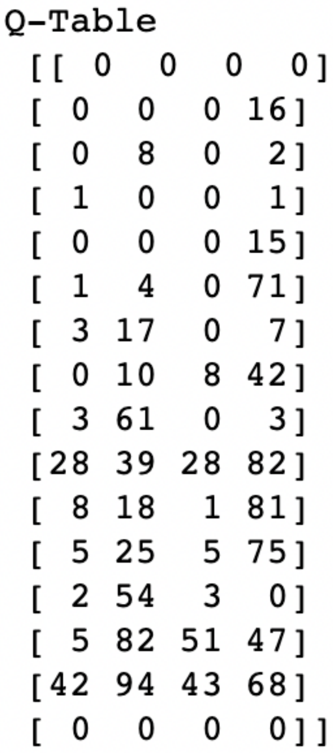


Figure 18

final q-table(Fig-31)) still our agent has learnt some kind of greedy policy for some state action pairs and can learn more

4.4 SARSA on Stochastic Env

Analysis : after training, from the above plots, we observed that epsilon(Fig-32) has been decayed almost by just 100 iterations. the total reward per episode(Fig-33) has been started to converge to a range after around 500-530 iterations.the fluctuations are due to the random actions being taken throughout the training(due to epsilon being a min of 0.01). only the q-values SUM(Fig-35) has not converged which means we can still train the agent more(this can be



```
Q-Table
 [[ 0  0  0  0]
 [ 0  0  0 16]
 [ 0  8  0  2]
 [ 1  0  0  1]
 [ 0  0  0 15]
 [ 1  4  0 71]
 [ 3 17  0  7]
 [ 0 10  8 42]
 [ 3 61  0  3]
 [28 39 28 82]
 [ 8 18  1 81]
 [ 5 25  5 75]
 [ 2 54  3  0]
 [ 5 82 51 47]
 [42 94 43 68]
 [ 0  0  0  0]]
```

Figure 19

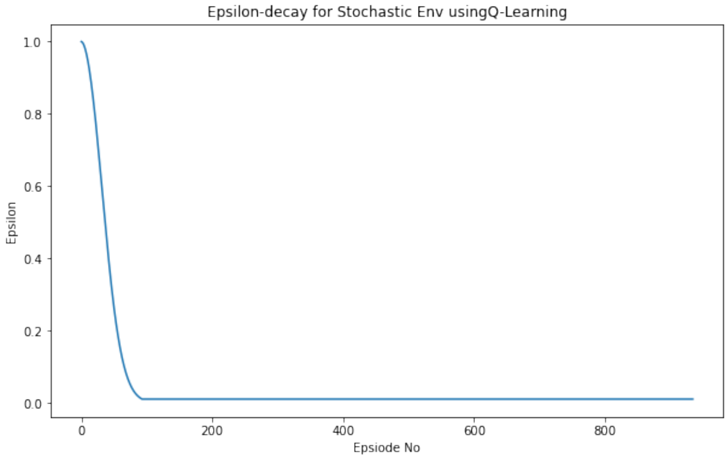


Figure 20

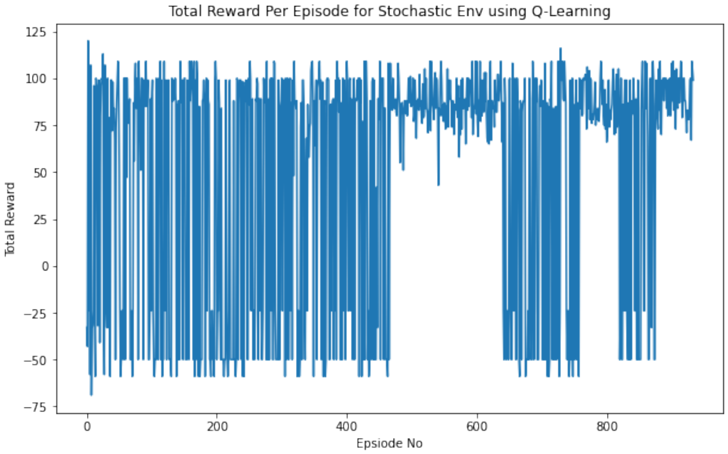


Figure 21

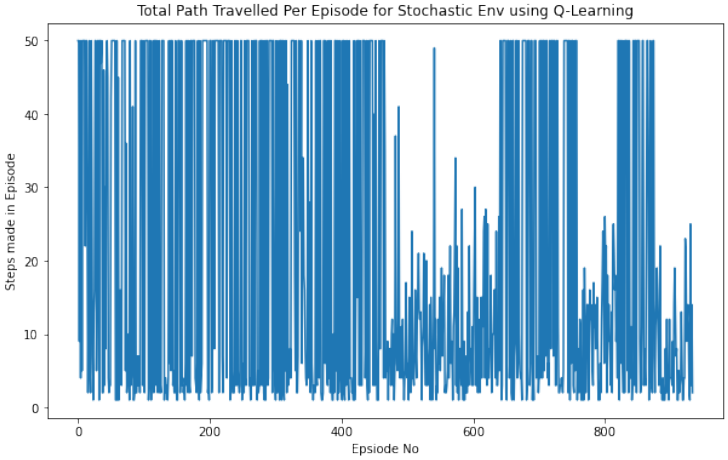


Figure 22

viewed in the zeroes present in the final q-table(Fig-37))

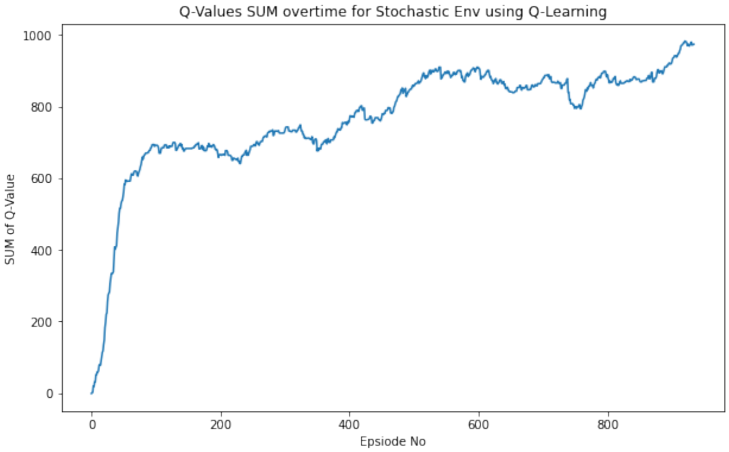


Figure 23

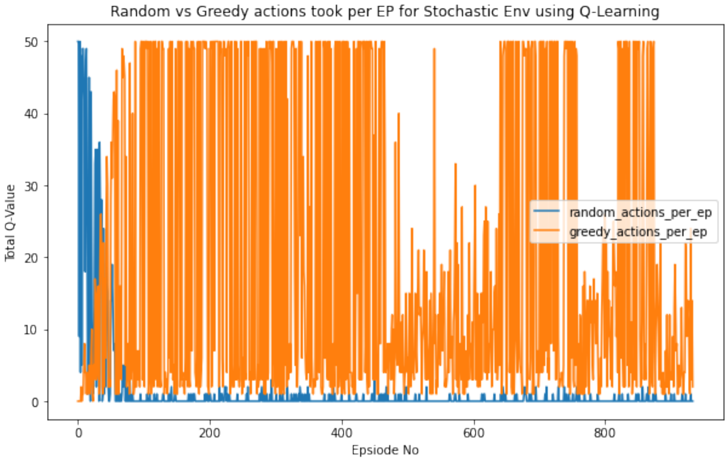
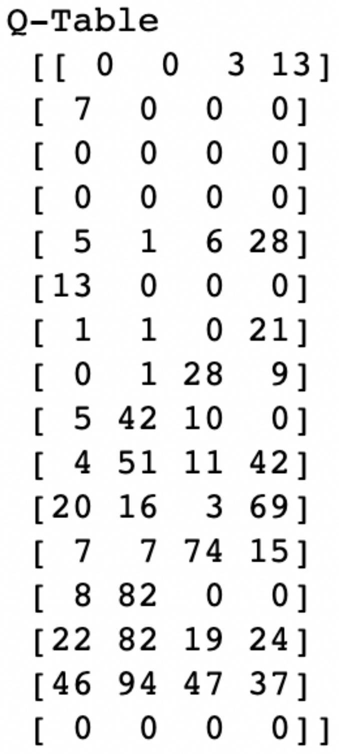


Figure 24



```
Q-Table
 [[ 0  0  3 13]
 [ 7  0  0  0]
 [ 0  0  0  0]
 [ 0  0  0  0]
 [ 5  1  6 28]
 [13  0  0  0]
 [ 1  1  0 21]
 [ 0  1 28  9]
 [ 5 42 10  0]
 [ 4 51 11 42]
 [20 16  3 69]
 [ 7  7 74 15]
 [ 8 82  0  0]
 [22 82 19 24]
 [46 94 47 37]
 [ 0  0  0  0]]
```

Figure 25

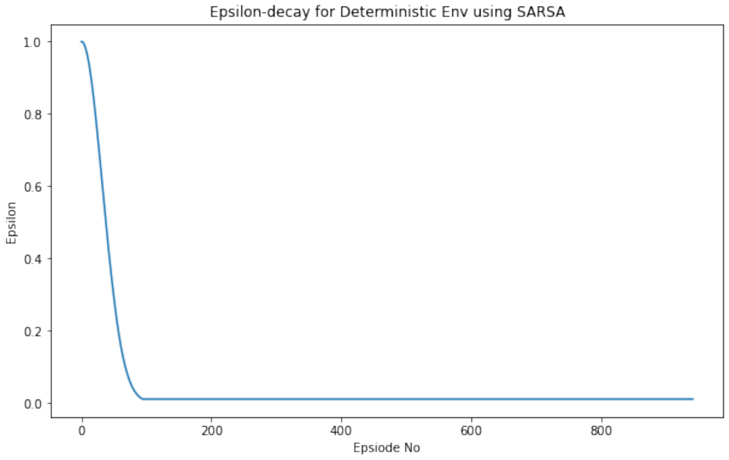


Figure 26

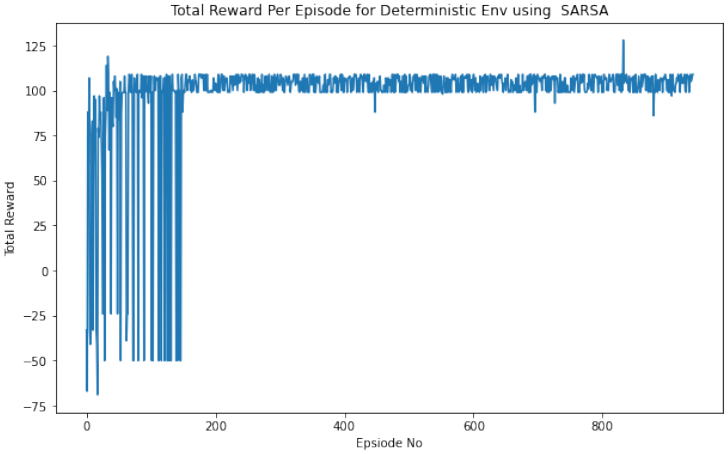


Figure 27

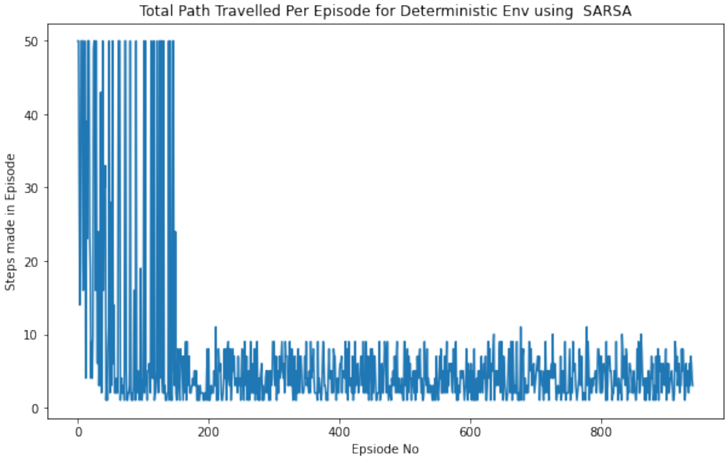


Figure 28

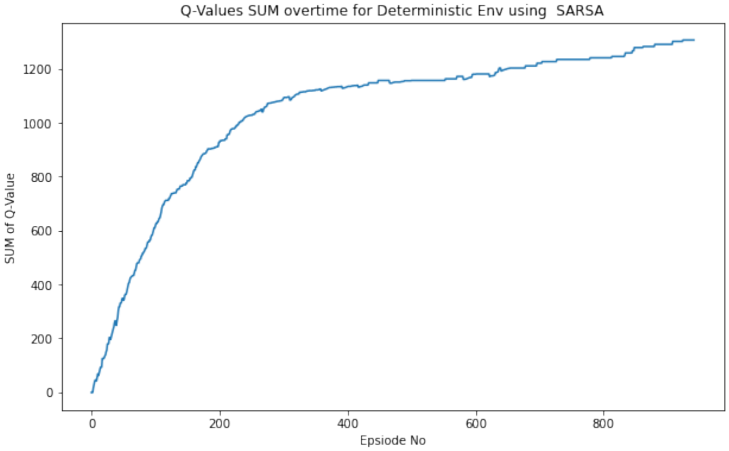


Figure 29

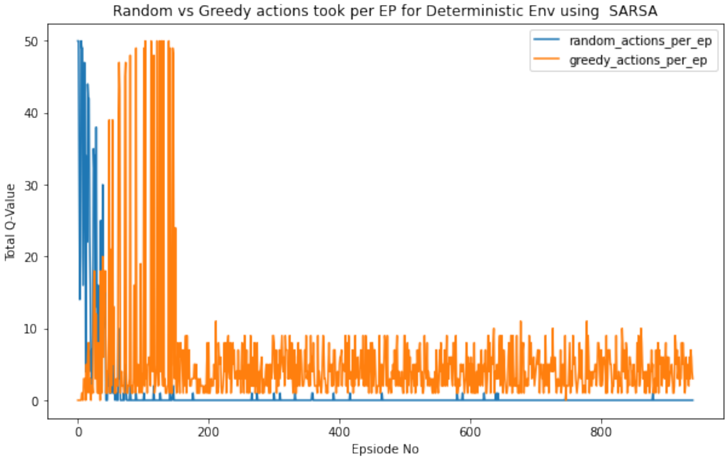
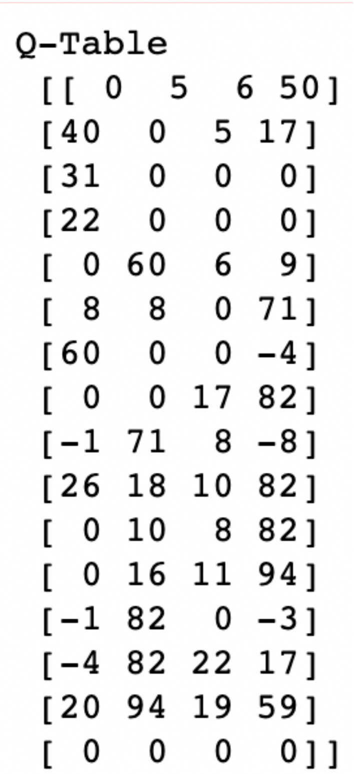


Figure 30



```
Q-Table
 [[ 0  5  6 50]
 [40  0  5 17]
 [31  0  0  0]
 [22  0  0  0]
 [ 0 60  6  9]
 [ 8  8  0 71]
 [60  0  0 -4]
 [ 0  0 17 82]
 [-1 71  8 -8]
 [26 18 10 82]
 [ 0 10  8 82]
 [ 0 16 11 94]
 [-1 82  0 -3]
 [-4 82 22 17]
 [20 94 19 59]
 [ 0  0  0  0]]
```

Figure 31

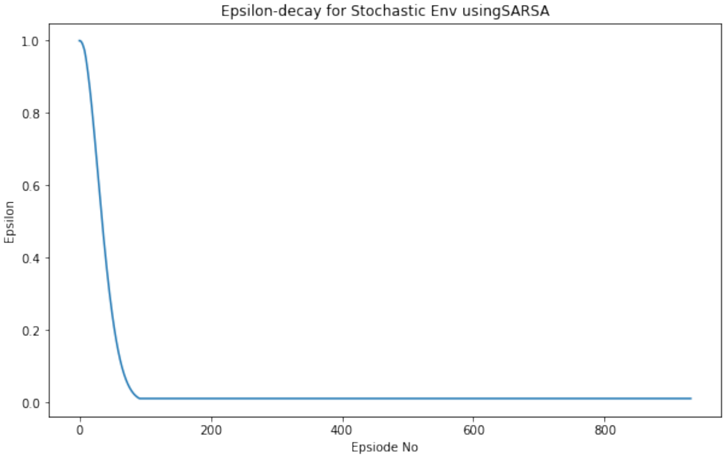


Figure 32

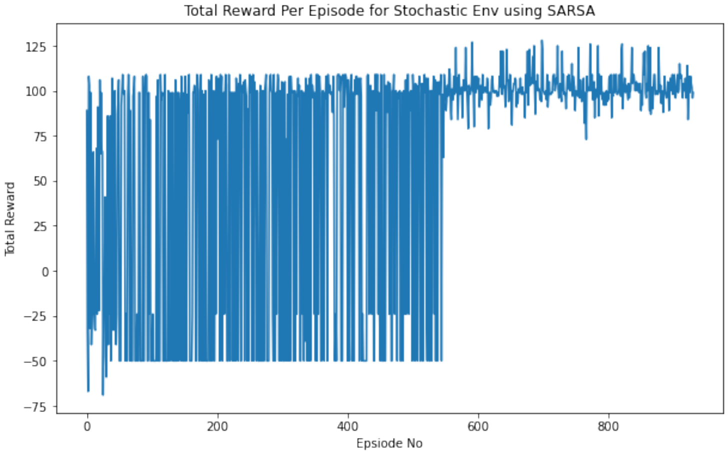


Figure 33

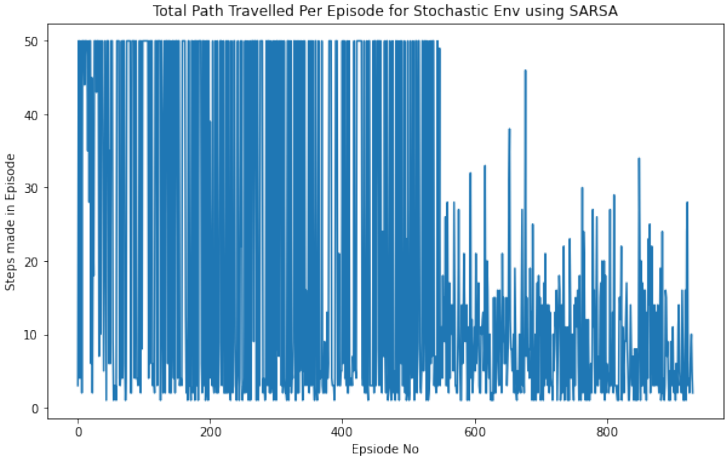


Figure 34

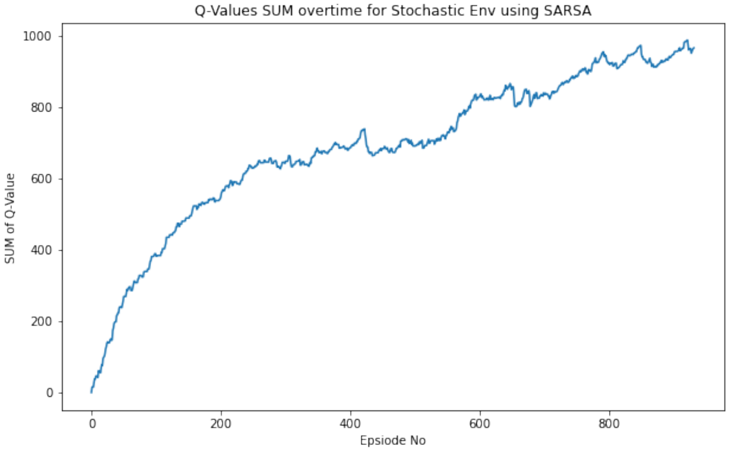


Figure 35

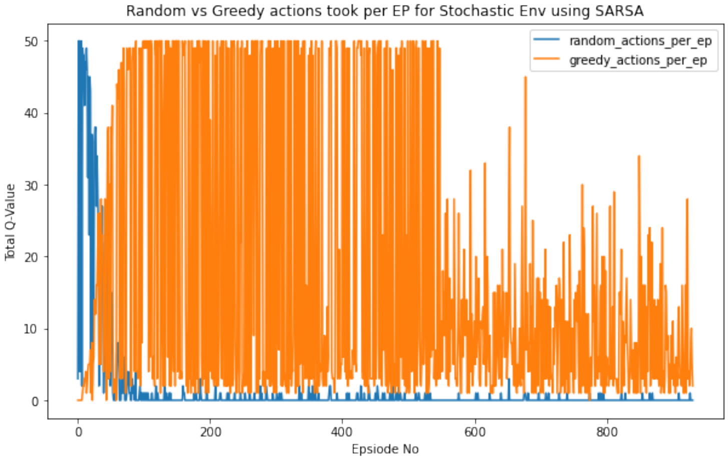
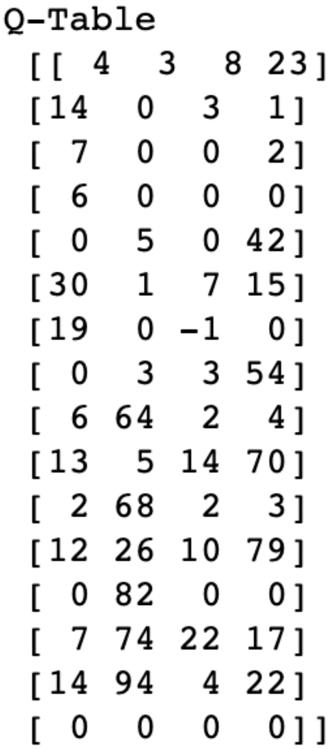


Figure 36



```
Q-Table
 [[ 4  3  8 23]
 [14  0  3  1]
 [ 7  0  0  2]
 [ 6  0  0  0]
 [ 0  5  0 42]
 [30  1  7 15]
 [19  0 -1  0]
 [ 0  3  3 54]
 [ 6 64  2  4]
 [13  5 14 70]
 [ 2 68  2  3]
 [12 26 10 79]
 [ 0 82  0  0]
 [ 7 74 22 17]
 [14 94  4 22]
 [ 0  0  0  0]]
```

Figure 37

5 Q-Learning VS SARSA

let's compare the two algorithms on both of the environments we defined.

5.1 Deterministic Env

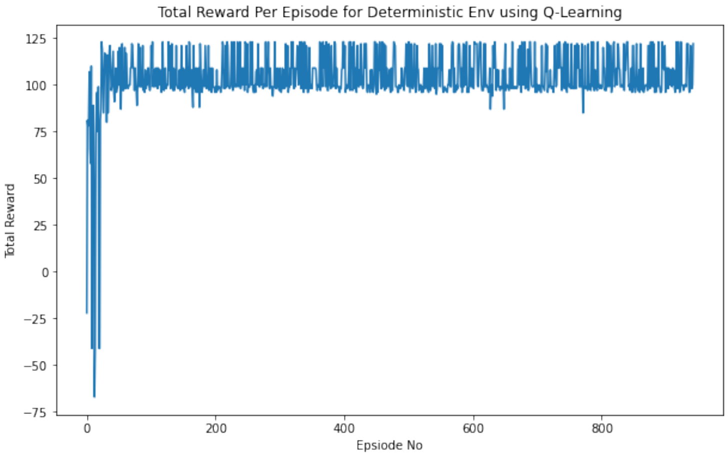


Figure 38

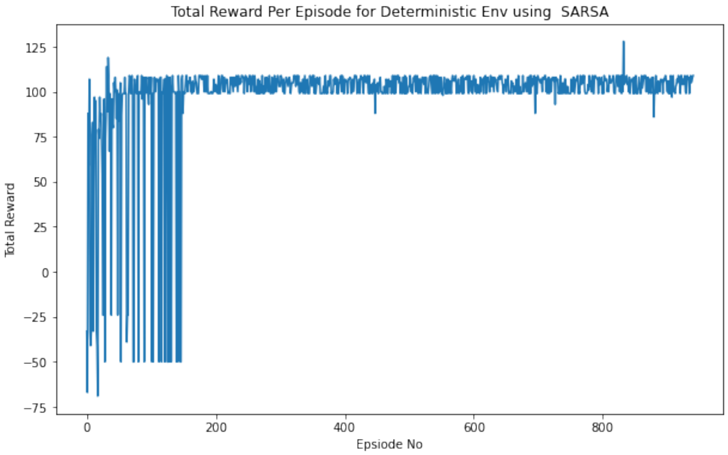


Figure 39

Analysis : For a given Deterministic Environment, we observe from both of the plots that (Fig-38,39)

- Q-Learning has converged earlier than SARSA
- Reward fluctuations(max,min among states) are lesser for SARSA and higher for q-learning
- Q-Learning has higher Rewards than for SARSA for almost all of it's states. it has achieved more reward for the same set of states Q-Learning has performed better Overall

5.2 Stochastic Env

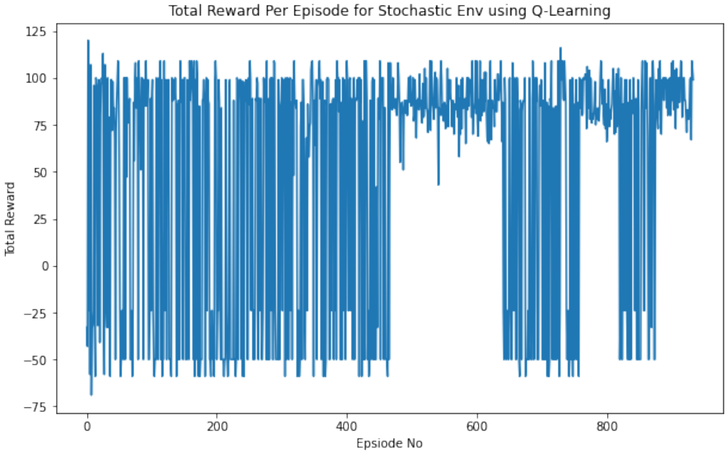


Figure 40

Analysis : For a given Stochastic Environment, we observe from both of the plots that (Fig-40,41)

- Q-Learning has started converging earlier than SARSA
- Reward fluctuations(max,min among states) are lesser for SARSA after convergence and higher for q-learning even after achieving some kind of convergence SARSA has performed better overall

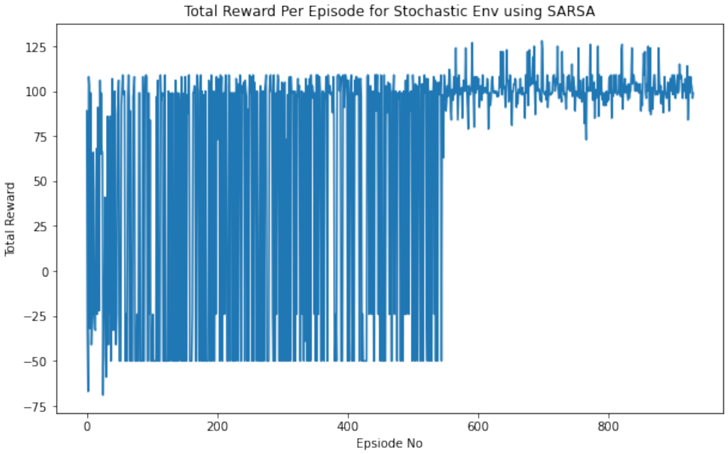


Figure 41

6 Conclusion(Ending-notes)

so yeah after solving the grid, we called our friend and informed him about our results & analyses which are that,

if he is in a non-magical grid(Deterministic Env) he should use Q-Learning to reach to the gauntlet as soon as possible(with little energy consumption). also the episodes are not enough under current parameters/hyperparameters and needs to be increased. he will meet wanda(+25) who will assist him in his quest and should not meet iron man(+10) as thanos(-20) is watching him(nearby rewards, and he can take them both out) and should look out only for maw and avoid him(-10) even if he takes a longer route and he can be there within 8 moves.

but if he is in a magical grid(Stochastic Env)(probable being set up by Doctor Strange) he should use SARSA to reach to the gauntlet as soon as possible. also he may have to deal with both thanos and maw(stochastic actions) but if luck favors both ironman and wanda(+ve rewards) will assist him. All the best to vision in this doom-day.

As for our Assignment,for both the algorithms, the episodes are not enough under current parameters/hyperparameters and needs to be increased or parameters need to be modified.

7 BONUS-HyperParameter Tuning

7.1 Total No of Episodes

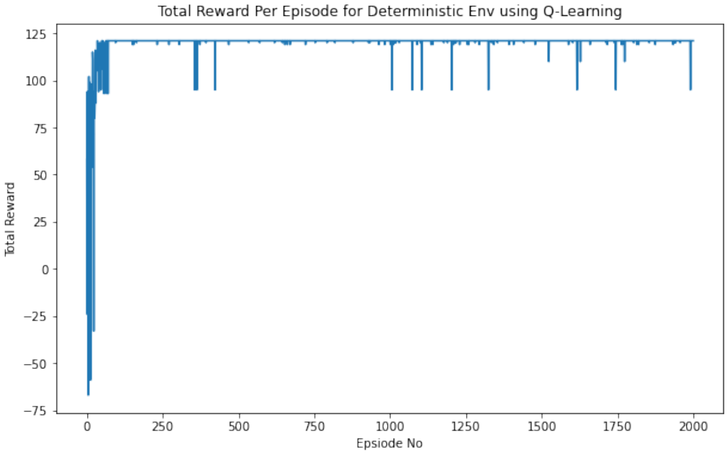


Figure 42: Episodes-2000

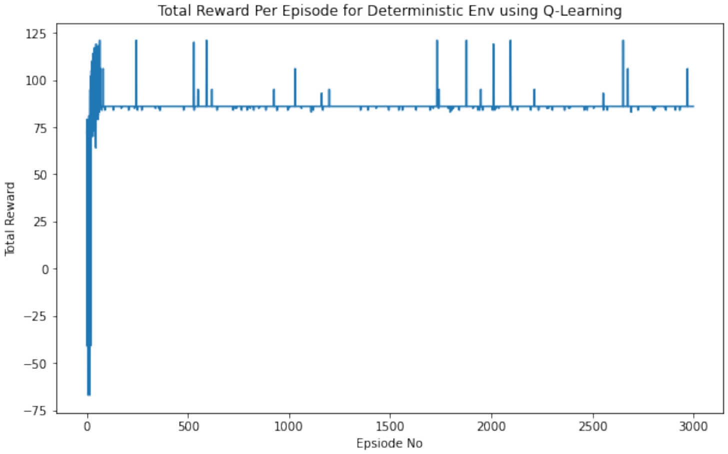


Figure 43: Episodes-3000

Here we have tuned for 3 values of total no of episodes,as mentioned in the captions. the observation is that the avg reward value for 2000 and 5000 are high but not 3000. this could be due to over learning.

the most efficient parameters for my our set up can be 2000 as the reward is not increasing much later and in some cases decreasing and we can rather reduce computational power.

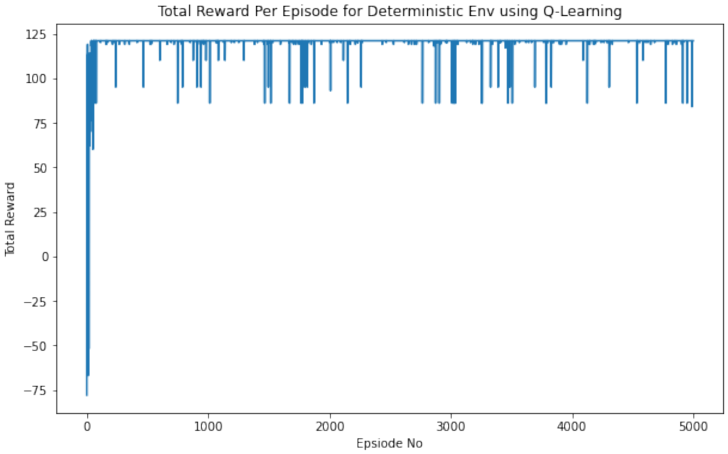


Figure 44: Episodes-5000

Figure 45

7.2 Discount Factor(gamma)

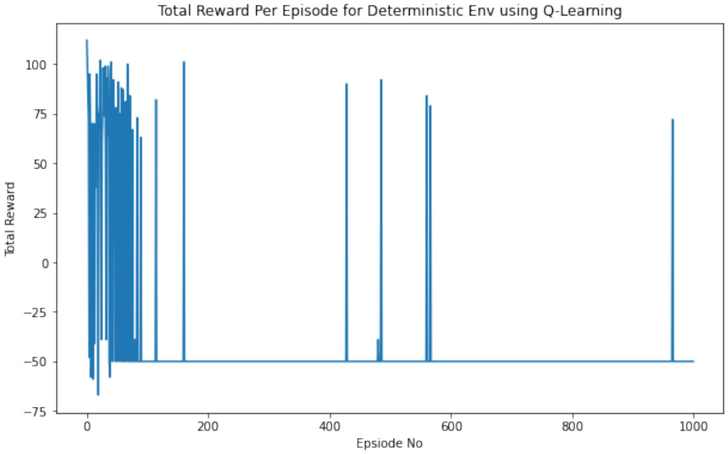


Figure 46: gamma-0.5

Here we have tuned for 3 values of gamma,as mentioned in the captions. the observation is that the for gamma =0.5, the rewards are really low indicating agent is not trying to learn the optimal policy as future rewards are getting cut off. after increasing it to 0.75 and then to 0.9, rewards are high, agent is learning and after that converging

but for

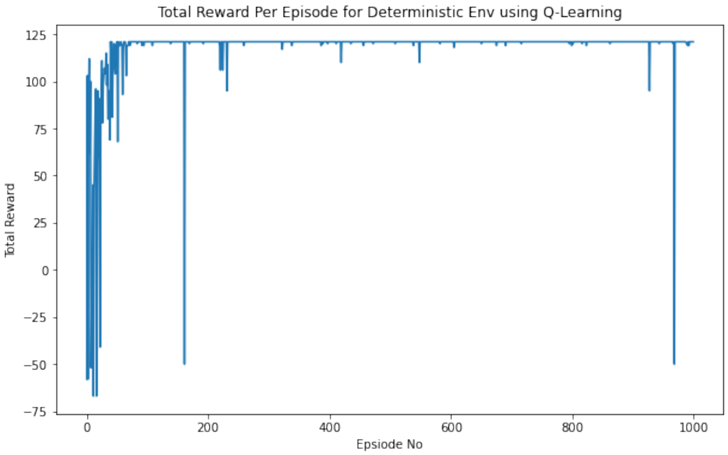


Figure 47: gamma-0.75

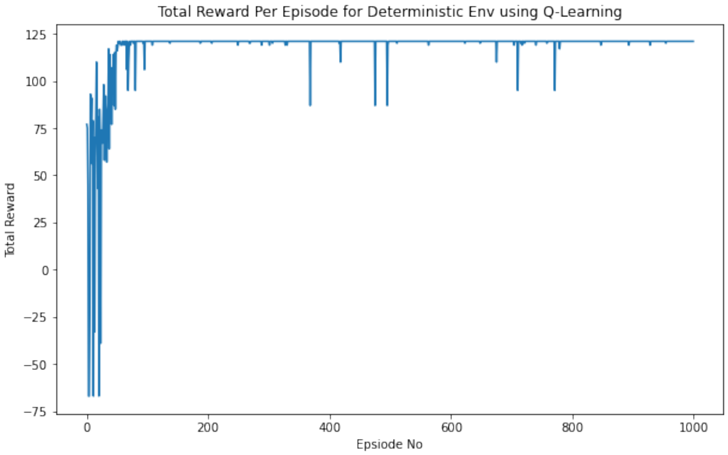


Figure 48: gamma-0.9

Figure 49

the most efficient parameters for my our set up can be gamma= 0.9 in this case, it can go even higher but not more than 1 as it would not discount future rewards against the formula and assumptions

8 References

- NIPS Styles (docx, tex)
- Overleaf (LaTex based online document generator) - a free tool for creating professional reports GYM environments
- GYM environments
- Lecture slides
- Wikipedia
- Richard S. Sutton and Andrew G. Barto, ”Reinforcement learning: An introduction” (pdf)
- google.com and internet in general(only for referencing not for copying)
- marvel comics

9 Github Link

This is my link: RL-Assg-1.