CSE 4/546: Reinforcement Learning (Spring 2022)

Assignment 1: Defining and Solving RL Environment

By: Akhil Goyal - 50415687

1.1 Defining Environment:

Environment defined for this assignment have 121 states defined in 11x11 grid (shown in Figure 1). The agent is indicated with yellow pixel and goal position with the dark green pixel. The objective for the agent is to maximize rewards and reach goal with minimum steps while avoiding traps indicated by black pixels. The agent can take 4 possible actions {up, down, left, right} and reward as {-10, 0 ,7, 10 ,20}, -10 for traps, {7, 10} for first checkpoints (light green pixels) and 20 for reaching the goal. The agent will also lose 1 point for each step taken to ensure the bot reaches the target with minimum steps.

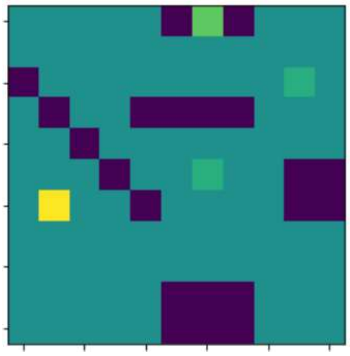
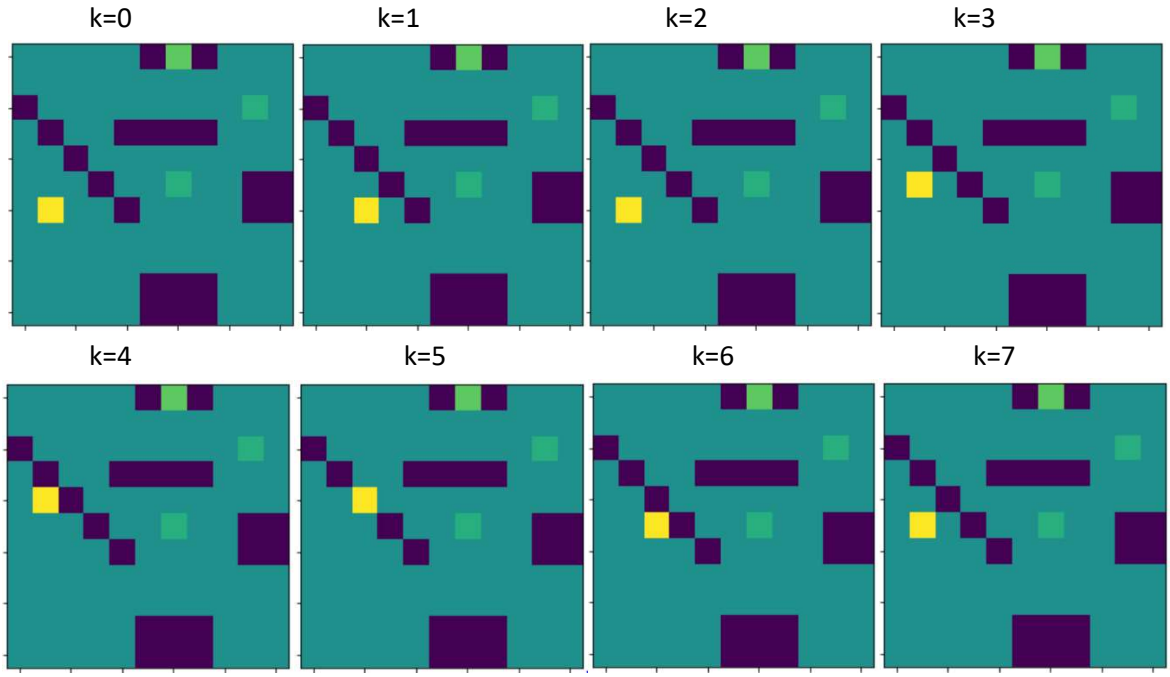


Figure 1: Map

1.2 Visualizations for Environment:

Steps taken by the agent before learning in the designed environment is shown in figure 2 for k=10 interations. The agent takes random action and rewarded or penalized.



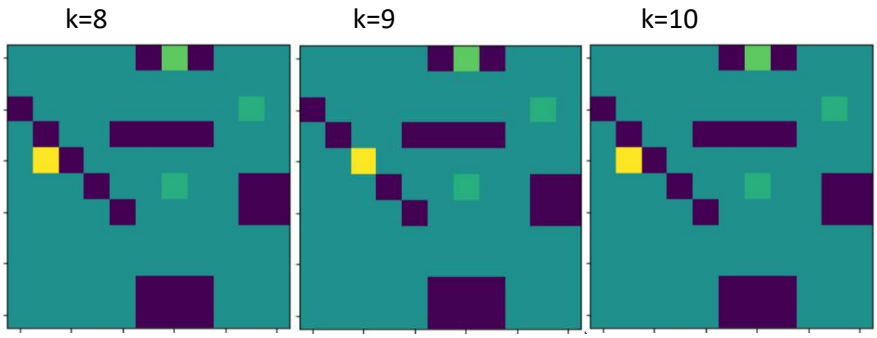


Figure 2: Visualization of Environment

1.3 Defining stochastic Environment:

The stochastic environment defined for the assignment have probabilistic action model, i.e the probability of reaching the desired state after taking the desired action is p(S=s'|A=a)=0.9.

1.4 Difference in Deterministic and Stochastic Environment:

The major difference between the deterministic and stochastic environment is that the agent will certainly reach the desired state

For deterministic environment: $p(s',r|s,a) = \{0,1\}$

As for stochastic environment reaching the desired state given the action taken is probabilistic i.e there is uncertainty that it might not reach the desired state for 9 out of 10 times and will end up at some other state.

For stochastic environment: $\sum_s \sum_r p(s', r \mid s, a) = 1$

1.5 Safety in AI:

As agent will lose points for each step taken, thus, this ensures the agent can't go out of the play area. As for instance, if the agent is at the boundary say at top left corner of the map it can't move left so if left action is taken it will remain at same place and will lose points for remaining at same state.

2 Learning for the agent:

Learning models used for the assignment are Q-Learning and Double Q-Learning for the both the environments (Deterministic and Stochastic). Update function used for both the models is given below

For Q-Learning:

```
Q_table[prev_pos,action]=Q_table[prev_pos,action] + alpha*(reward + gamma*np.max(Q_table[pos,:])-Q_table[prev_pos,action])
```

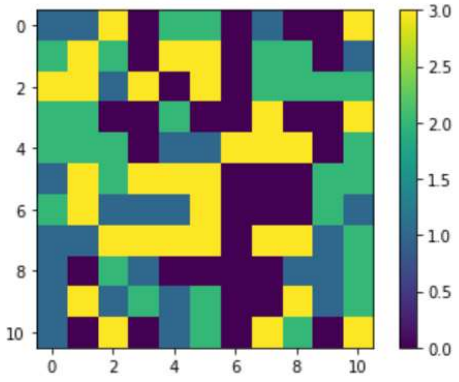
For Double Q-Learning:

```
if np.random.choice([0,1], p = [0.5,0.5]):
      QA_table[prev_pos,action]=QA_table[prev_pos,action] + alpha*(reward + gamma*QB_table[pos,np.argmax(QA_table[pos,:])]-QA_table[prev_pos,action])
else:
      QB_table[prev_pos,action]=QB_table[prev_pos,action] + alpha*(reward + gamma*QA_table[pos,np.argmax(QB_table[pos,:])]-QB_table[prev_pos,action])
```

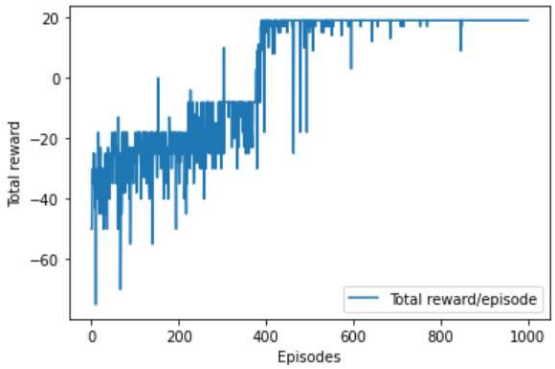
As the name suggests the double Q-learning model uses two tables which it randomly updates as shown above whereas normal Q-learning just uses one.

2.1.1 Deterministic Environment with Q- Learning:

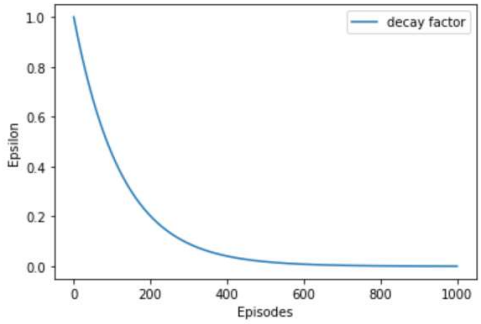
Parameters: *Alpha = 0.25*
Gamma = 0.95
Epsilon = $e^{(-8*episode/Total\_episodes)}$
Total Training episodes = 1000



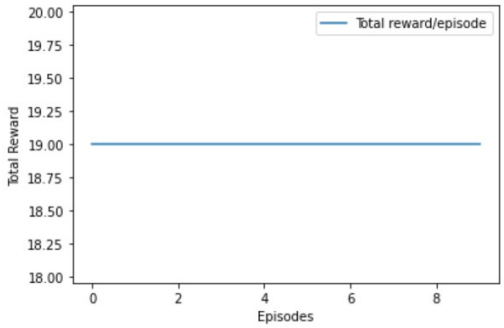
(a)



(b)



(c)



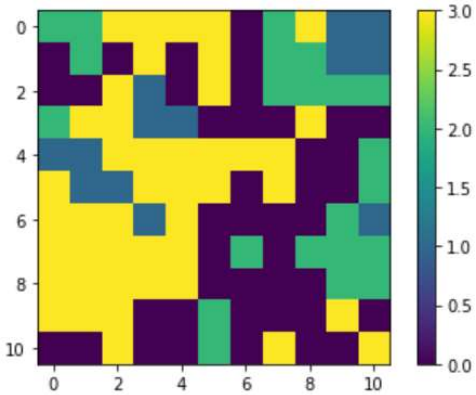
(d)

Figure 3

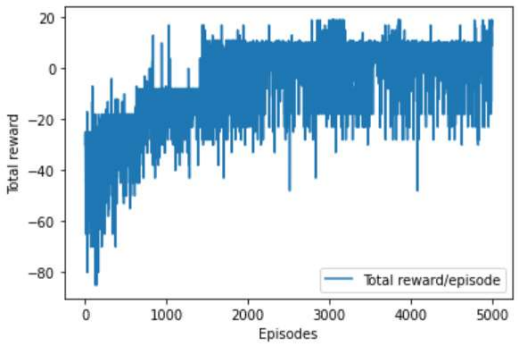
fig: 3a represents optimal policy for each state (black: UP, blue: DOWN, green: LEFT, yellow: RIGHT), fig 3.b represents reward accumulated in every episode during learning phase. The fig 3.c represents epsilon decay based on greed algorithm, fig 3.d represents reward accumulated in every episode during action (post learning) phase.

2.1.2 Stochastic Environment with Q-Learning:

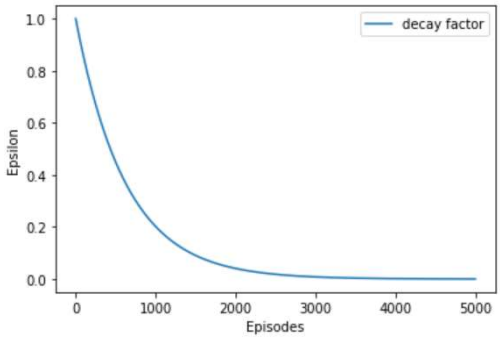
Parameters: *Alpha = 0.25*
Gamma = 0.95
Epsilon = $e^{(-8*episode/\ Total\ \_episodes)}$
Total Training episodes = 5000



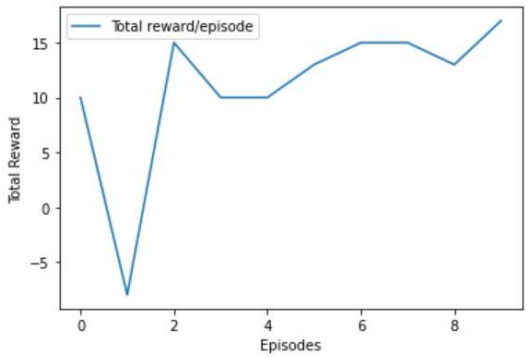
(a)



(b)



(c)



(d)

Figure 4

fig: 4a represents optimal policy for each state (black: UP, blue: DOWN, green: LEFT, yellow: RIGHT), fig 4.b represents reward accumulated in every episode during learning phase. The fig 4.c represents epsilon decay based on greed algorithm, fig 4.d represents reward accumulated in every episode during action (post learning) phase

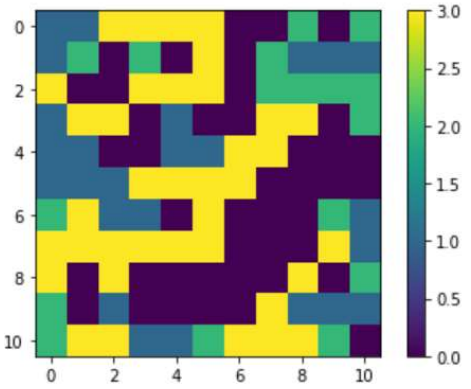
2.1.3 Deterministic Environment with Double Q- Learning:

Parameters: *Alpha = 0.25*

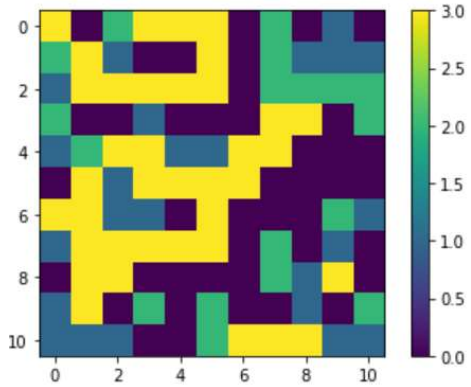
Gamma = 0.95

Epsilon = $e^{(-8*episode/Total\_episodes)}$

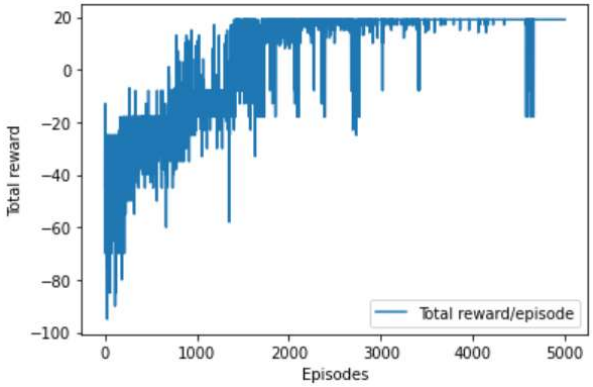
Total Training episodes = 5000



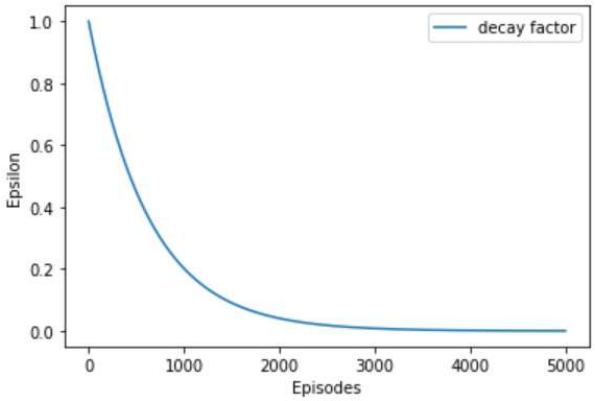
(a)



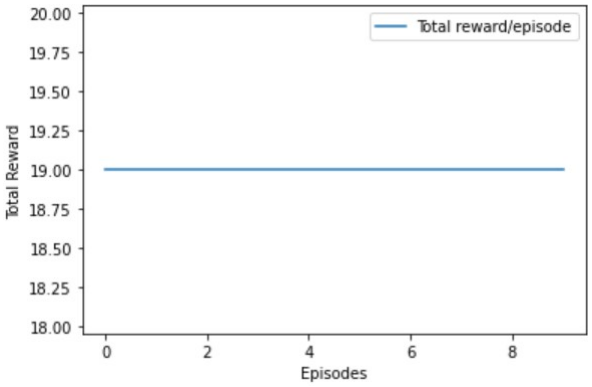
(b)



(c)



(d)



(e)

Figure 5

fig: 5a and 5b represents optimal policy from two Q tables for each state (black: UP, blue: DOWN, green: LEFT, yellow: RIGHT), fig 5.c represents reward accumulated in every episode during learning phase. The fig 5.d represents epsilon decay based on greed algorithm, fig 5.e represents reward accumulated in every episode during action (post learning) phase.

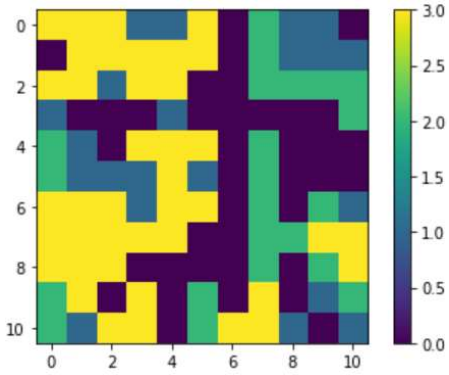
2.1.4 Stochastic environment with Double Q-Learning:

Parameters: *Alpha = 0.25*

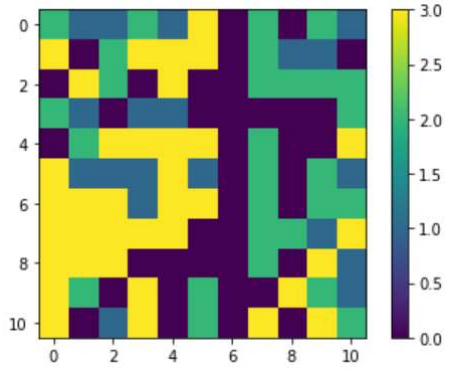
Gamma = 0.95

Epsilon = $e^{(-8*episode/Total\_episodes)}$

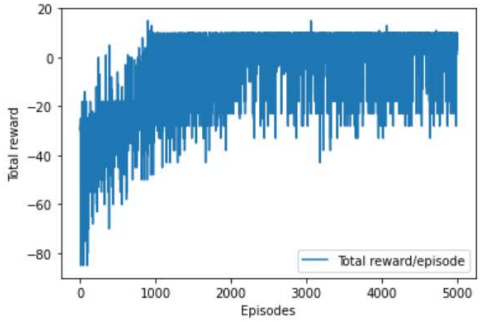
Total Training episodes = 5000



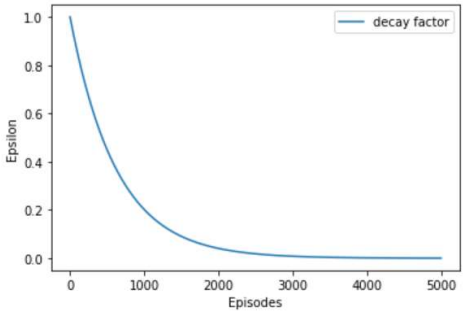
(a)



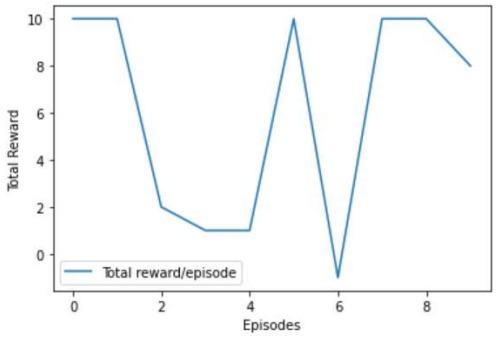
(b)



(c)



(d)

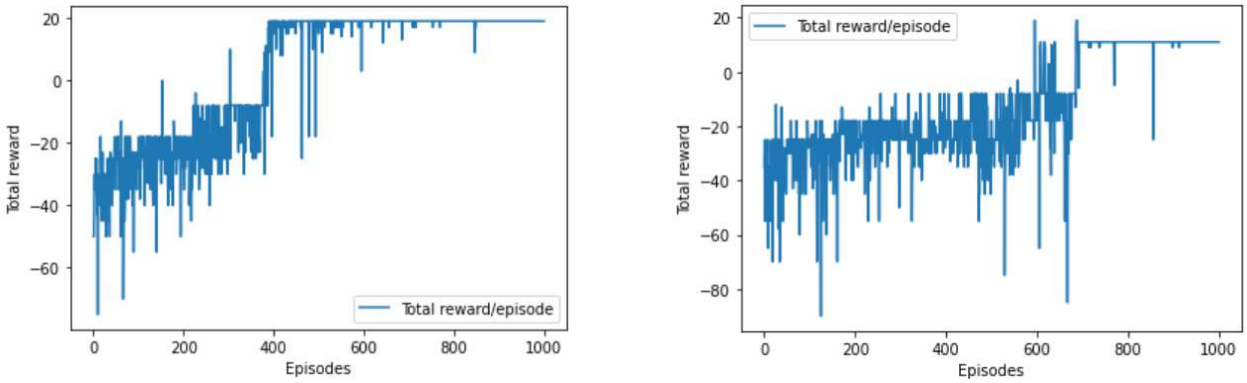


(e)

Figure 6

fig: 6a and 6b represents optimal policy from two Q tables for each state (black: UP, blue: DOWN, green: LEFT, yellow: RIGHT), fig 6.c represents reward accumulated in every episode during learning phase. The fig 6.d represents epsilon decay based on greed algorithm, fig 6.e represents reward accumulated in every episode during action (post learning) phase.

2.2 Comparison between Q-Learning and Double Q-Learning for deterministic environment:

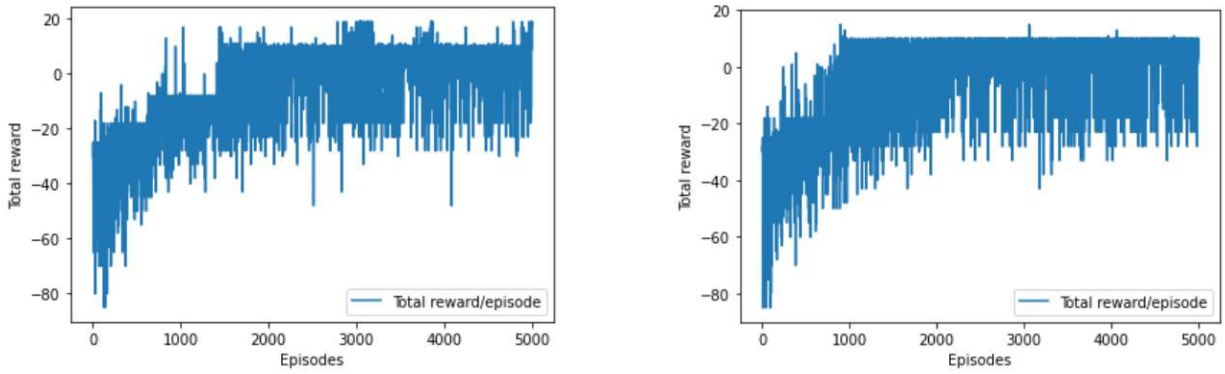


(a) (b)

Figure 7

Q-learning converges faster than Double Q-Learning as shown if figure 7.a(QL) and 7.b(DQL). In Q-learning model learning converges around 400-500 episodes where as in Double Q-learning model it converges around 700 episodes. But, after convergence there are more outliers in Q-Learning model as compared to the other model.

2.3 Comparison between Q-Learning and Double Q-Learning for stochastic environment:



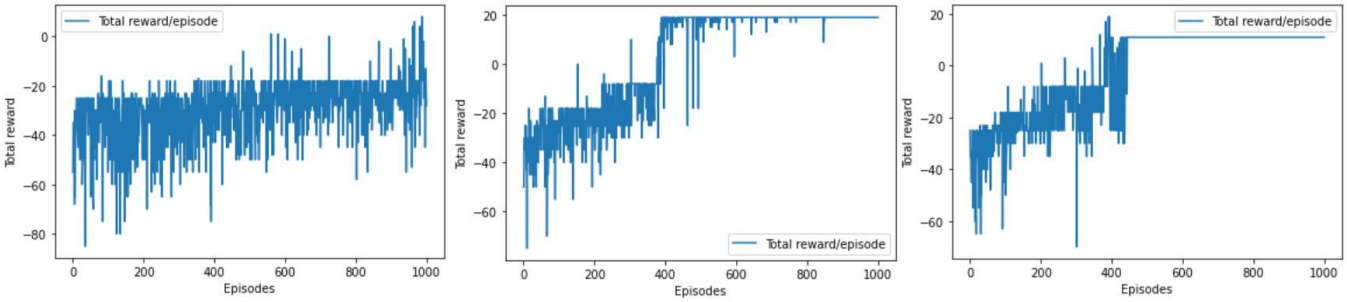
(a) (b)

Figure 8

In contrast to deterministic environment, in stochastic environment Double Q-Learning model performs better than Q-Learning model. In Double Q-Learning it converges at 1000 episodes whereas the other one converges at around 1500 episodes (as shown in figure 8.a(QL) and 8.b(DQL)). This indicates Double Q-Learning is robust, thus, perform well even for stochastic environment.

3 Tuning hyperparameter

3.1 Decay Factor:

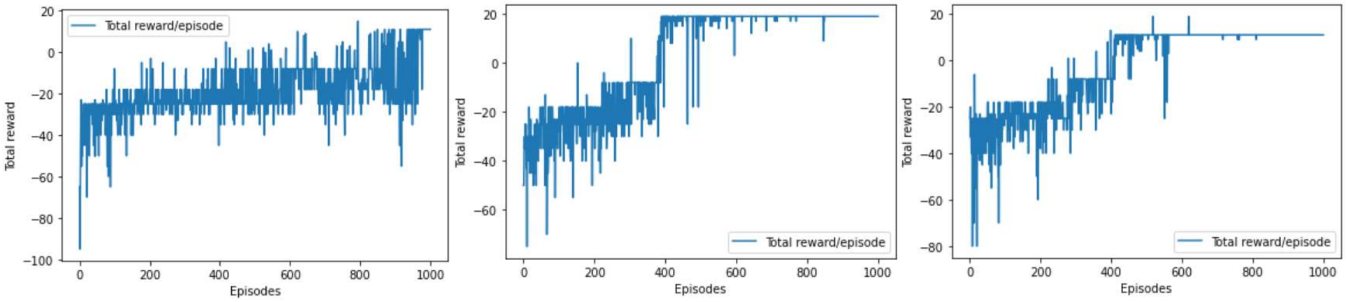


(a): e^{-1r} (b): e^{-8r} (c): e^{-15r}

Figure 9

The comparison is done for hyper parameter decay factor as shown in figure 9 for 3 different decay functions as indicated. Decay rate for a<b<c and thus convergence rate a<b<c, but when decay factor is e^{-15} agent does not accumulate all the rewards thus e^{-8r} is better where there is steady decay and agent can get accumulate all the rewards.

3.2 Learning Rate:



(a): α=0.1 (b): α=0.25 (c): α=0.5

Figure 10

The comparison is done for hyper parameter learning rate as shown in figure 10 for 3 learning rates. As learning rate is a<b<c and thus convergence rate a<b<c.