CSE 546: Reinforcement Learning Assignment 1

Abhilash Sampath
sampath2@buffalo.edu

Abstract

We formulate a grid based game based on Markov decision process and visualise it. The game will have deterministic and stochastic environments in which the agent will be taught the optimal policy through reinforcement learning - tabular methods to achieve the goal state.

The code can be found on Github

1 Environments

In reinforcement learning, Environment is the Agent's world in which it lives and interacts. The agent can interact with the environment by performing some action but cannot influence the rules or dynamics of the environment by those actions.

The environment of the game is defined by a 9 x 9 matrix with 81 states. The main objective of the agent is to attain the goal state.

- States : 81
- Actions: Up, Down, Left, Right, Top-Left, Top-Right, Bottom-Left, Bottom-Right
- Rewards:
  - : HP = +3
  - : Toxin = -3
  - : Demon = -100
  - : Goal = +100
  - : Move = -1

1.1 Deterministic Environment

In a deterministic environment, the next state of the environment can always be determined based on the current state and the agent's action.

The grid is deterministic by *default* where all the actions of the agent are sure events leading to a definite state.

For example, in our context of the game,

- When the agent moves left from a tile with positions [2, 0], it always reaches [1, 0].
- All the artifacts in the grid have fixed positions.

1.2 Stochastic Environment

In a stochastic reinforcement learning environment, we cannot always determine the next state of the environment from the current state by performing a certain action.

The grid can be made stochastic by randomizing the outcomes of actions performed by the agent. For example, in our context of the game,

- When the agent moves left from a tile with positions [2, 0], it moves left to [1,0] with the probability of 0.99 and stays in [2, 0] for the remaining probability.
- The artifacts like HP and Toxin is randomly distributed across the grid.

1.3 Deterministic vs Stochastic

The deterministic environment's response to agent's action is certain. However, in stochastic environments, the response to the agent's action is not set in stone. As we saw in the earlier subsections the environment could randomly respond with a different state or reward than before, in consecutive time steps.

In the game, the stochastic environment is simulated through random distribution of artifacts in the grid as opposed to deterministic environment

2 Visualization

Figure 1: Agent

Figure 2: Toxin

Figure 3: HP

Figure 4: Demon

Figure 5: Goal

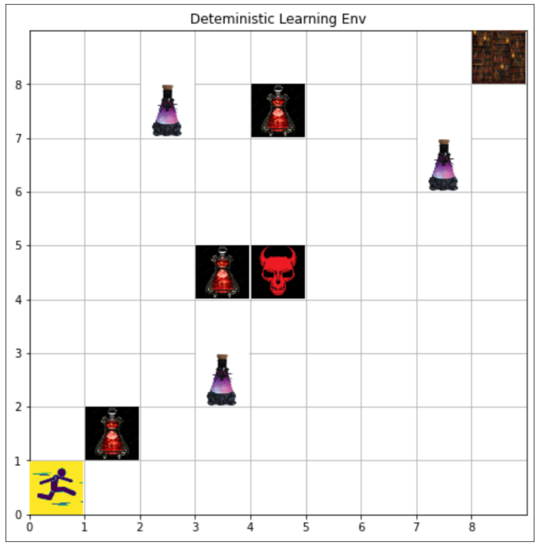

Figure 6: Initial positions in a deterministic environment
The initial positions of the grid in a deterministic environment with the Toxin and HP in fixed positions and
Agent in [0, 0]
Demon in [4, 4]
Goal in [8, 8]

Figure 7: Initial positions in a stochastic environment
The initial positions of the grid in a stochastic environment with the *Toxin and HP varying in numbers and positions*

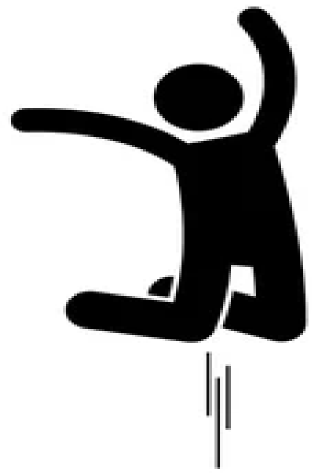

Figure 8: Agent interaction with HP

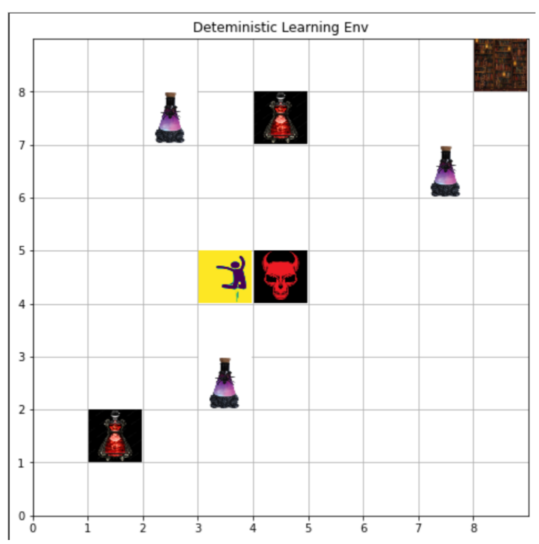

Figure 9: The agent consumes HP and gains 3 points

Figure 10: Agent interaction with Toxin

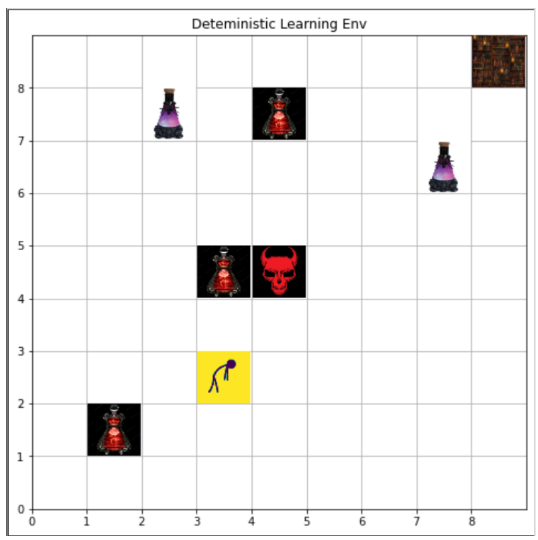

Figure 11: The agent consumes Toxin and loses 3 points

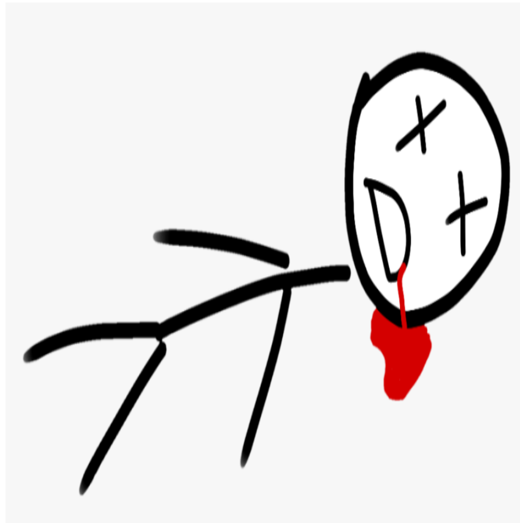

Figure 12: Agent interaction with Demon

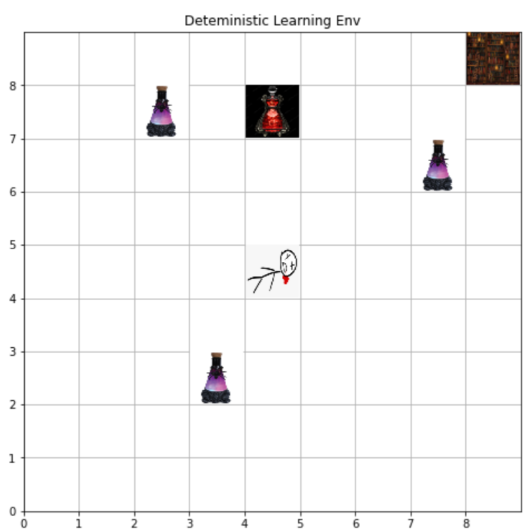

Figure 13: The agent loses 100 points when it interacts with Demon and the game stops

Figure 14: Agent interaction with Goal

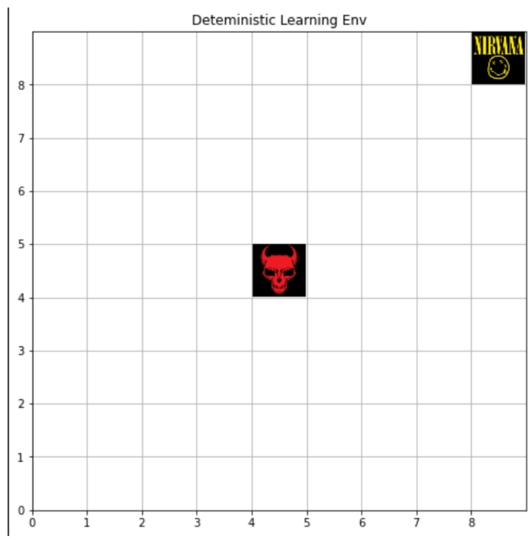

Figure 15: The agent gets 100 points when it interacts with Goal and the game stops

3 Safety in AI

AI Safety is collective termed ethics that we should follow so as to avoid problem of accidents in machine learning systems, unintended and harmful behavior that may emerge from poor design of real-world AI systems

In the context of our game, we ensure the safety of AI by enforcing boundaries for our agent. Whenever the agent tries to go past the grid, it is put back in its original position.

We also stop the agent from exploration after a defined max time step.

4 Tabular methods used in the solution

4.1 Q Learning

This algorithm is one of the tabular methods which is driven by the value function of state action pair.

Q Learning is model free, which means it does not depend on the transition probability distribution of the states.

It should also be noted that Q Learning is an off policy algorithm. This means that the target value can be calculated without any regard to how experience was generated. In simpler terms, it does not need an entire sequence of an episode to compute Q value of states.

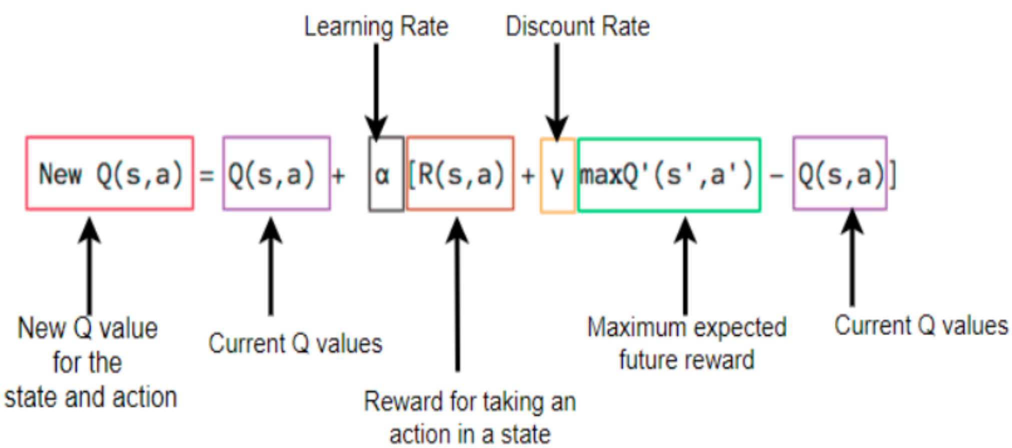

Figure 16: Q Learning Equation

4.2 Double Q Learning

Q Learning performs poorly in some stochastic environments. The poor performance is due to the large over estimations of action values. This bias is introduced due to the fashion in which Q Learning works, using maximum action value as approximation for expected maximum estimated action value.

This can be combated by using Double Q Learning where the first Q table works as the estimator and the second one is used evaluate the Q value of the first table.

This process ensures a better convergence.

$$Q^A(s,a) \leftarrow Q^A(s,a) + \alpha(s,a)\left(r + \gamma Q^B(s', a^*) - Q^A(s,a)\right)$$
$$Q^B(s,a) \leftarrow Q^B(s,a) + \alpha(s,a)\left(r + \gamma Q^A(s', b^*) - Q^B(s,a)\right)$$

Figure 17: Double Q Learning Equation

5 Q Learning agent in a Deterministic Environment

The agent was trained using Q Learning algorithm in the deterministic environment discussed, with the following parameters

- Max Timesteps = 300
- Discount Factor = 0.99
- Learning Rate = 0.3
- Epsilon = 1.0
- Episodes = 1000

The results of the learning are visualized in the plots

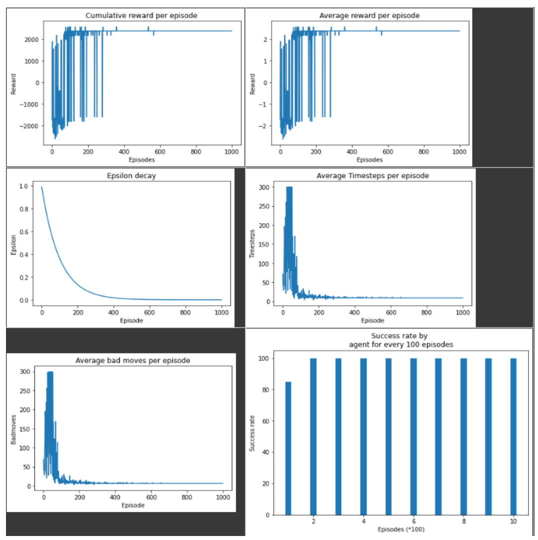

Figure 18: Results of Q Learning in a Deterministic Environment

The cumulative rewards start to stabilize around 250 episodes. This illustrates that the agent has learnt the optimal policy to maximize discounted future cumulative rewards.

The same inference can also be drawn when we observe the epsilon decay plot. The agent starts to favor exploitation over exploration as it improves the learnt policy.

It should be noted that the average number of time steps and the average number of bad moves by the agent are almost identical. The reward setting in the environment guides the agent to not waste time steps. Therefore the penalized moves will be commensurate with the number of time steps.

6 Q Learning agent in a Stochastic Environment

The agent was trained using Q Learning algorithm in the stochastic environment discussed, with the following parameters

- Max Timesteps = 300
- Discount Factor = 0.99
- Learning Rate = 0.3
- Epsilon = 1.0
- Episodes = 1000

The results of the learning are visualized in the plots

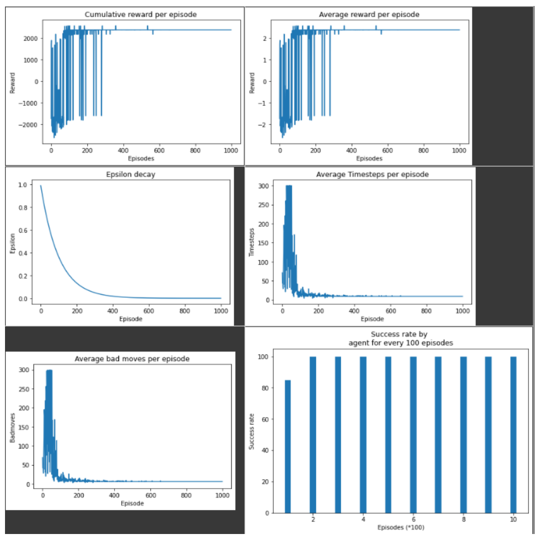

Figure 19: Results of Q Learning in a Stochastic Environment

The cumulative rewards start to stabilize around 275 episodes. This illustrates that the agent has learnt the optimal policy to maximize discounted future cumulative rewards.

The inferences drawn from the discussion under deterministic environment holds true here as well.

Despite the stochastic environment being technically challenging we notice that the convergence is comparable with the deterministic environment results.

7 Double Q Learning agent in a Deterministic Environment

The agent was trained using Double Q Learning algorithm in the deterministic environment discussed, with the following parameters

- Max Timesteps = 300
- Discount Factor = 0.99
- Learning Rate = 0.3
- Epsilon = 1.0
- Episodes = 1000

The results of the learning are visualized in the plots

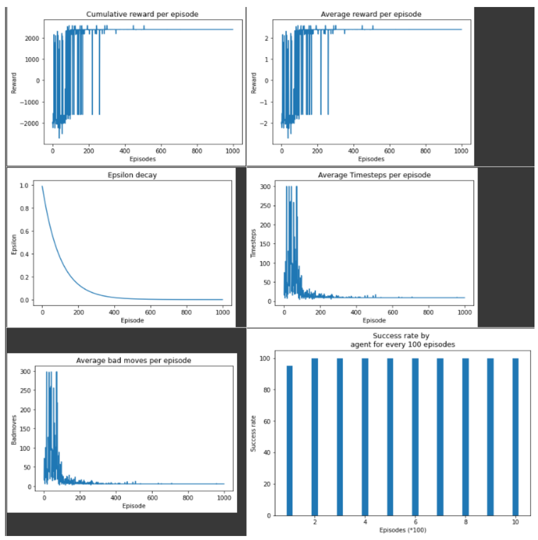

Figure 20: Results of Double Q Learning in a Deterministic Environment

The cumulative rewards start to stabilize around 200 episodes. This illustrates that the agent has learnt the optimal policy to maximize discounted future cumulative rewards faster than the Q Learning algorithm, as expected.

The same inference can also be drawn when we observe the epsilon decay plot. The agent starts to favor exploitation over exploration as it improves the learnt policy. Again, proving this to be better than Q Learning.

8 Double Q Learning agent in a Stochastic Environment

The agent was trained using Double Q Learning algorithm in the stochastic environment discussed, with the following parameters

- Max Timesteps = 300
- Discount Factor = 0.99
- Learning Rate = 0.3
- Epsilon = 1.0
- Episodes = 1000

The results of the learning are visualized in the plots

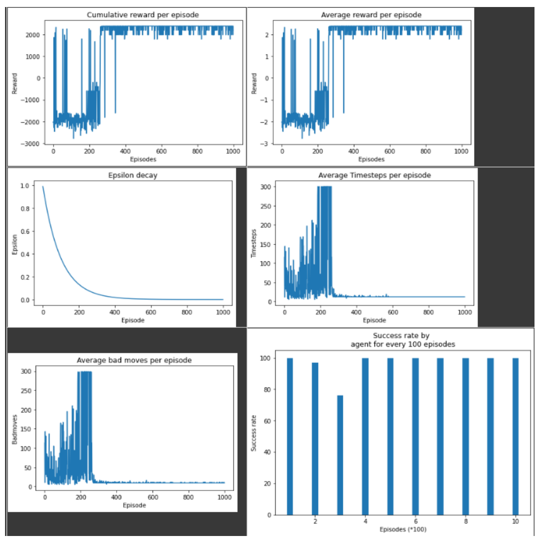

Figure 21: Results of Q Learning in a Stochastic Environment

The cumulative rewards start to stabilize around 400 episodes. This illustrates that the agent has learnt the optimal policy to maximize discounted future cumulative rewards.

The inferences drawn from the discussion under deterministic environment holds true here as well.

In this environment, the convergence must be delayed in comparison to deterministic and we see that. However, there is an interesting observation here - This algorithm is slower than Q Learning for Stochastic environment. This instance of results is possibly an outlier.

9 Evaluation Results

The Q Learning Agent and the Double Q Learning Agent were run for 30 episodes each using the learnt optimal policy.

The policies are optimal which can be observed in the linear plots of cumulative rewards for each kind

9.1 Q Learning Agent in the Deterministic Environment

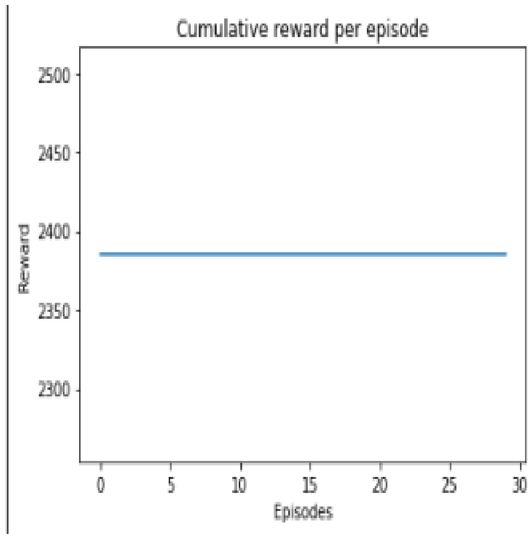

Figure 22: Cumulative rewards for the Q Learning Agent following optimal policy in the deterministic environment

9.2 Q Learning Agent in the Stochastic Environment

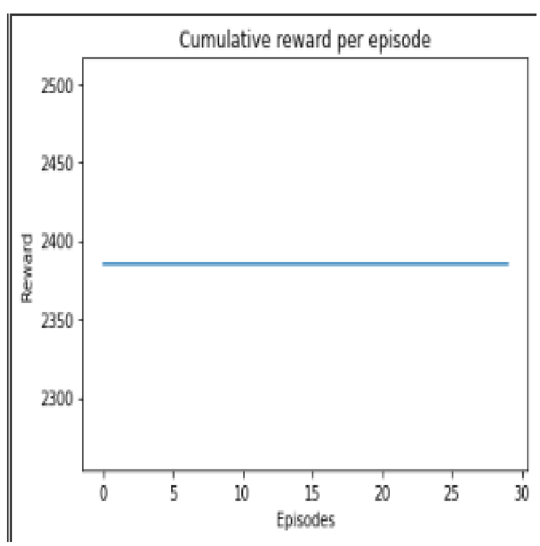

Figure 23: Cumulative rewards for the Q Learning Agent following optimal policy in the stochastic environment

133 ### 9.3 Double Q Learning Agent in the Deterministic Environment

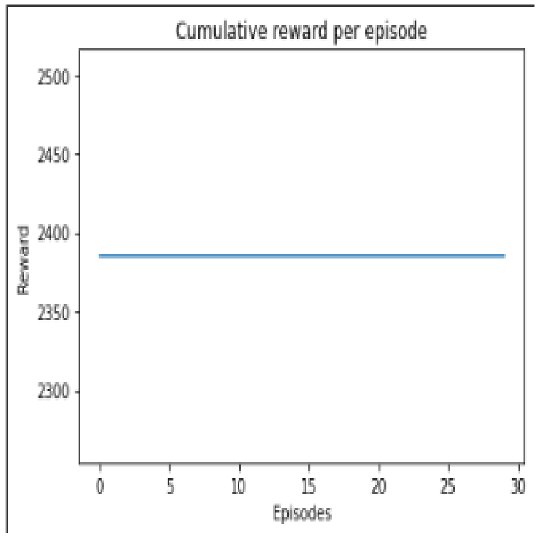

Figure 24: Cumulative rewards for the Double Q Learning Agent following optimal policy in the deterministic environment

134 ### 9.4 Double Q Learning Agent in the Stochastic Environment

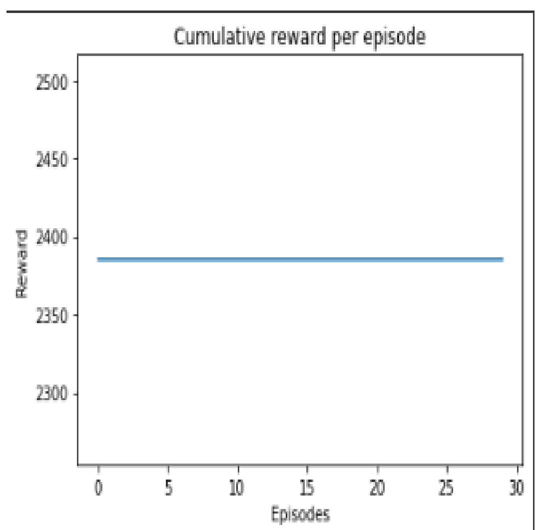

Figure 25: Cumulative rewards for the Q Learning Agent following optimal policy in the stochastic environment

10 Q Learning vs Double Q Learning in the Deterministic Environment

Double Q Learning performs better than Q Learning and it can be visualized in the plots

10.1 Q Learning performance metrics

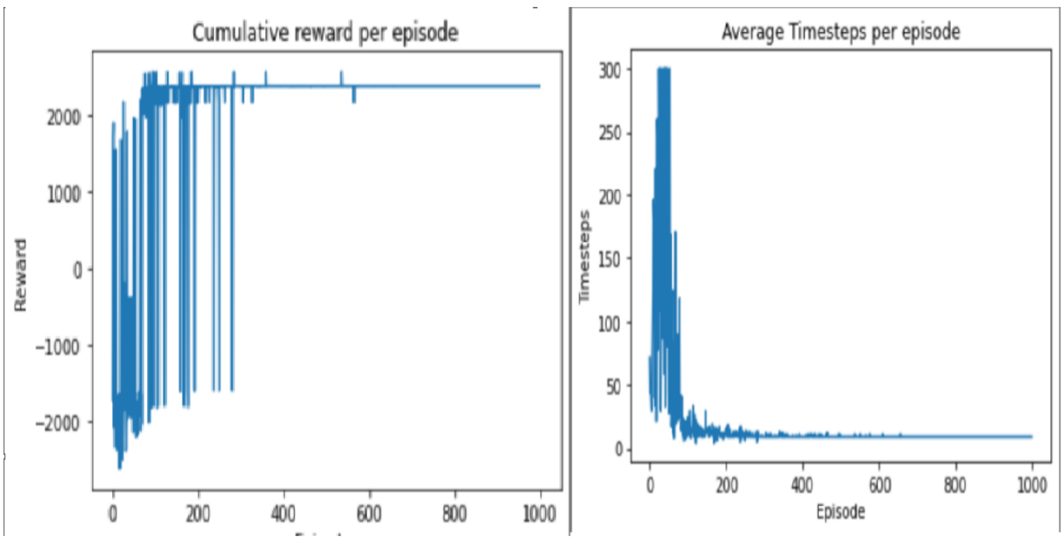

Figure 26: Q Learning performance metrics

10.2 Double Q Learning performance metrics

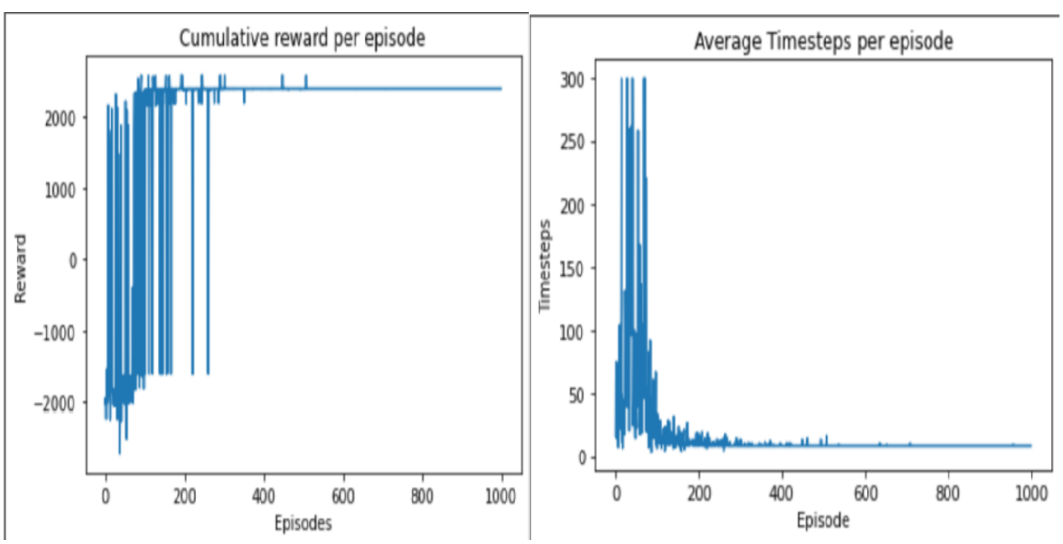

Figure 27: Double Q Learning performance metrics

It is evident from the plots that Double Q Learning converges faster than Q Learning

11 Q Learning vs Double Q Learning in the Deterministic Environment

Theoretically, Double Q Learning is supposed to perform better than Q Learning but in the plots we see below there is a contradiction

11.1 Q Learning performance metrics

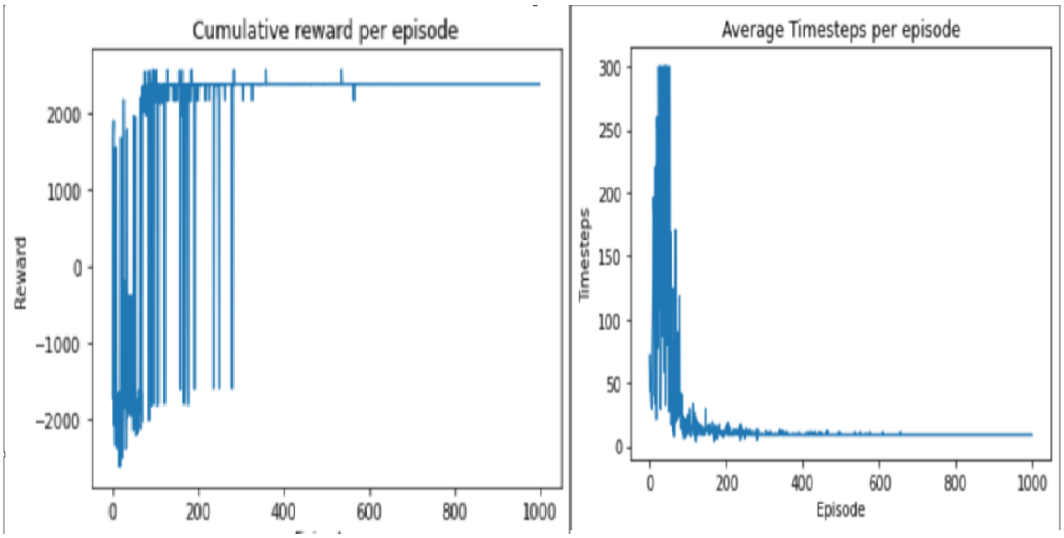

Figure 28: Q Learning performance metrics

11.2 Double Q Learning performance metrics

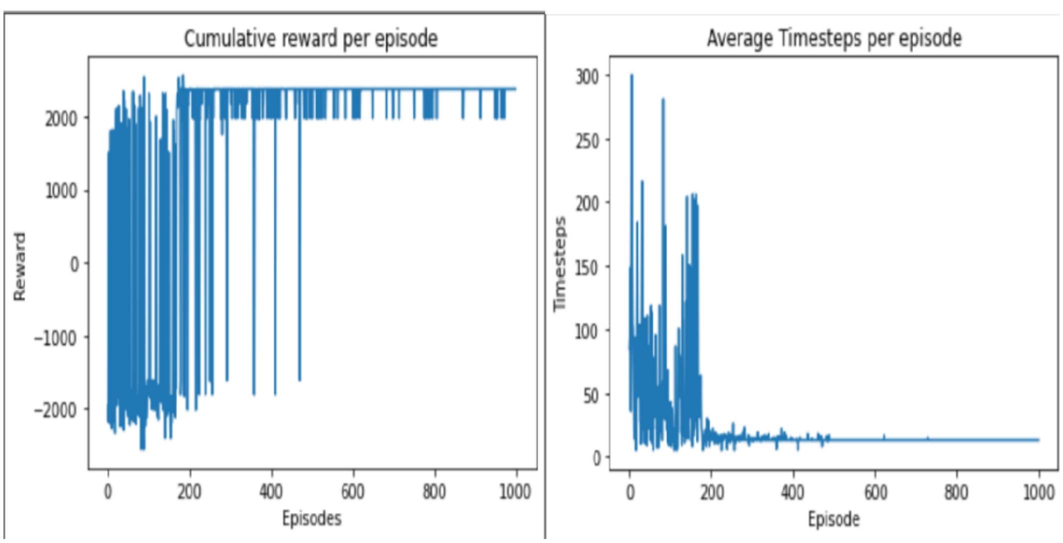

Figure 29: Double Q Learning performance metrics

It is evident from the plots that Q Learning is converging faster and it should also be noted that Double Q Learning oscillates in the higher region unlike Q Learning. It should be concluded that Double Q Learning performs better for as the complexity increases and this instance is an outlier

12 Hyper parameter Tuning

We consider the following hyper parameters and tweak them to observe how the algorithm responds and how the learning gets affected.

- Number of episodes
- Max Timesteps

12.1 Number of Episodes

Number of episodes used in the training plays a pivotal role in how the agent learns.

If the number of episodes are too less, the agent will have limited training and will not learn the optimal policy

On the other hand, if it is more than necessary, we would be wasting computational resources despite the optimal policy having been achieved.

We have used five values for episodes and the results are as follows

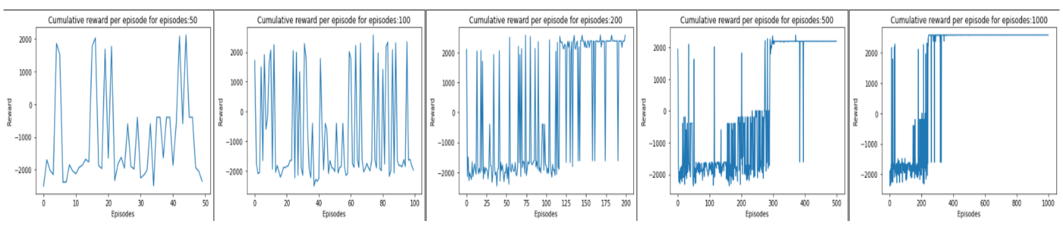

Figure 30: Cumulative rewards for varying episodes

It can be observed from the plots that for all the episode values less than 500, the cumulative rewards are inconsistent. The agent is trained around 400 episodes.

We can see that after 400 episodes the learning flat lines and we are wasting resources by running the algorithm further

12.2 Max Time steps

Max time steps dictates when the episode has to be terminated if the goal state has not been reached.

If the max time steps value is too less, the agent will have limited time steps to explore or exploit the environment in pursuit of goal and will not learn the optimal policy.

However, we would be wasting time steps, resources, and would not converge if it is more than required

We have used five values for max time steps and the results are as follows

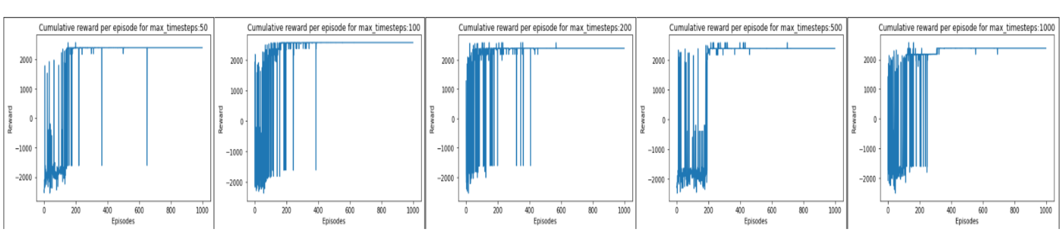

Figure 31: Cumulative rewards for varying max time steps

References

[1] Richard S. Sutton & Andrew G. Barto. Reinforcement Learning: An Introduction

[2] Alina Vereshchaka. CSE 546 Lectures & Slides

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, Dan Mané. Concrete Problems in AI Safety

[4] NIPS Styles (docx, tex)

[5] Overleaf (LaTex based online document generator) - a free tool for creating professional reports

[6] GYM environments