CSE 574: Introduction to Machine Learning

Fall 2018

Submitted by: Sai Kalyan Katta (UB Person Number: 50292522)

1. Introduction

Brain

A neural network is built which is having a three-layered structure with two hidden dense layers and one output layer. The first dense layer has the input dimensions as 4 and the output dimensions as 128. The second dense layer has the output dimensions as 128 and the output layer as the input size 128 and output 4. This helps to train tom about the path it should choose. The neural network helps the agent to explore all the grid by storing the experiences in a buffer and help it choose the optimal step as the next step.

```
model.add(Dense(output_dim =128,activation='relu',input_dim = self.state_dim))
model.add(Dense(output_dim =128,activation='relu'))
model.add(Dense(output_dim =self.action_dim,activation='linear'))
```

The activation functions for the 2 dense layers are "**relu**" and for the output layer it is "**linear**".

Exponential-decay Epsilon

The exponential-decay epsilon value helps to introduce the random selection of the next step by our agent. It also helps the agent to explore the environment completely.
The formula for epsilon:

$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) * e^{-\lambda |S|}$$

where $\epsilon_{min}, \epsilon_{max} \in [0,1]$
λ - hyperparameter for epsilon
$|S|$ - total number of steps

Q-Function

The Q-Function is defined as follows

$$Q_t = \begin{cases} r_t, & \text{if episode terminates at step } t+1 \\ r_t + \gamma max_a Q(s_t, a_t; \Theta), & \text{otherwise} \end{cases}$$

Q value is equal to the reward value if the next step is the goal state and if the next step is not the goal state we add the reward value with $\gamma \max_a Q(s_t,a)$ where $\max_a Q(s_t,a)$ is the maximum Q value of all the next possible steps.

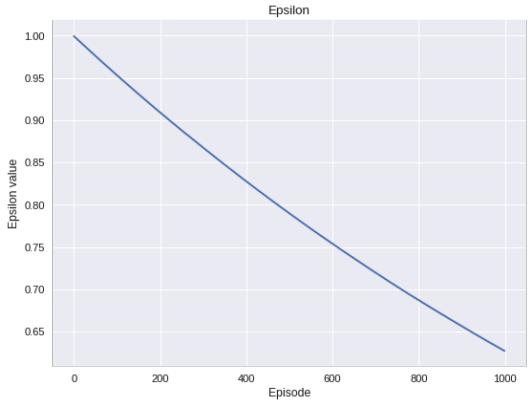
Because the future state Q value on prior, the agent keeps roaming around the same block again and again increasing the time taken. We can avoid this by using double deep Q learning.

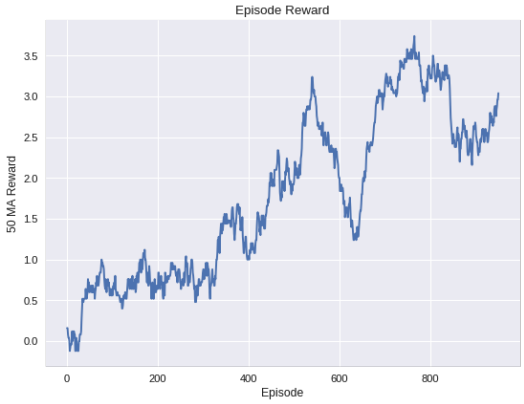
Depending the random states that the agent choses, the time taken by the agent to learn was at least ~400ms.

Analysis

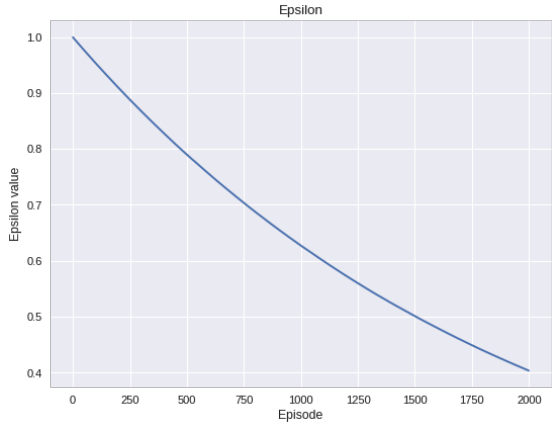
Keeping the Max_Epsilon, Min_Epsilon, Lambda at 1, 0.05, 0.00005 we change the number of episodes.

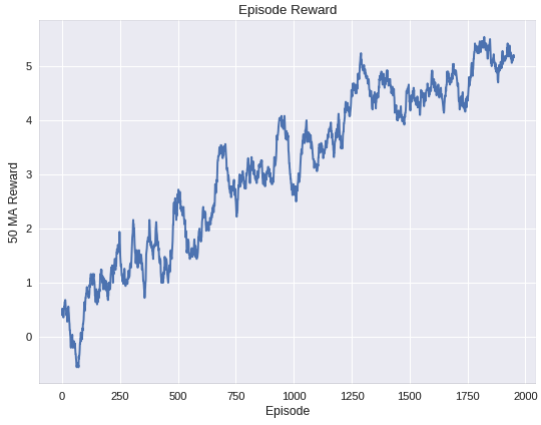
For 1000 episodes



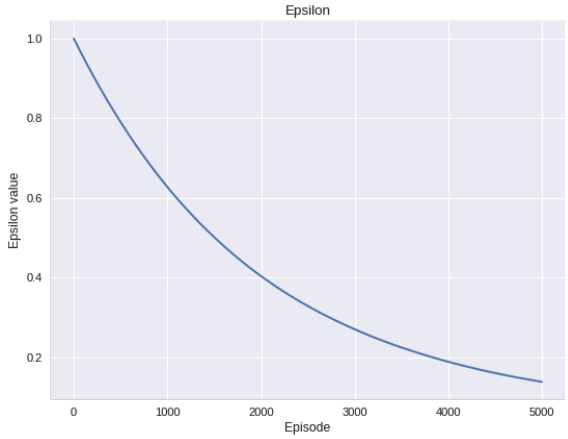


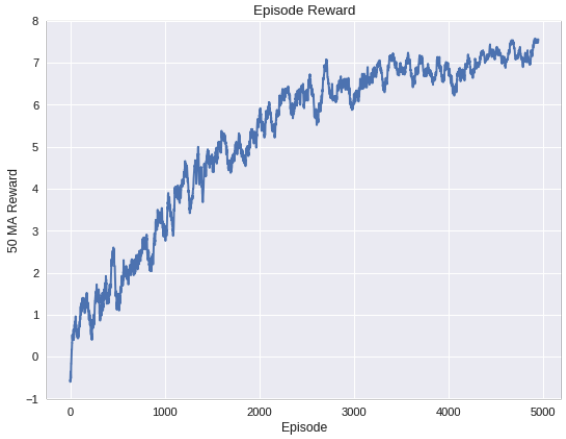
For 2000 episodes



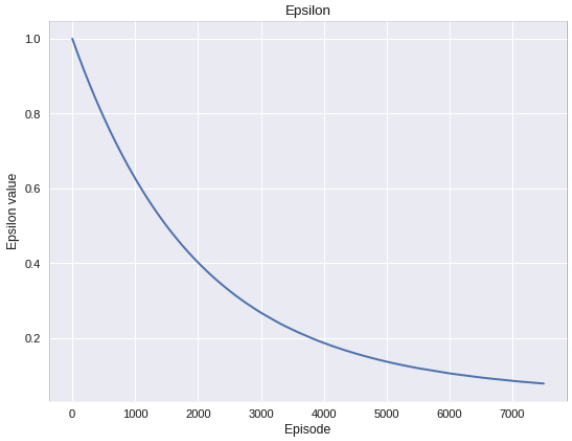


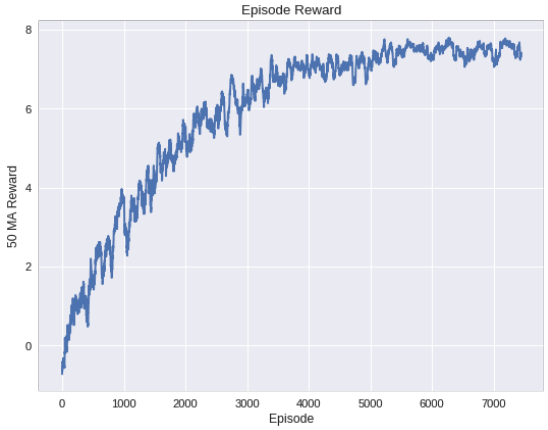
For 5000 episodes



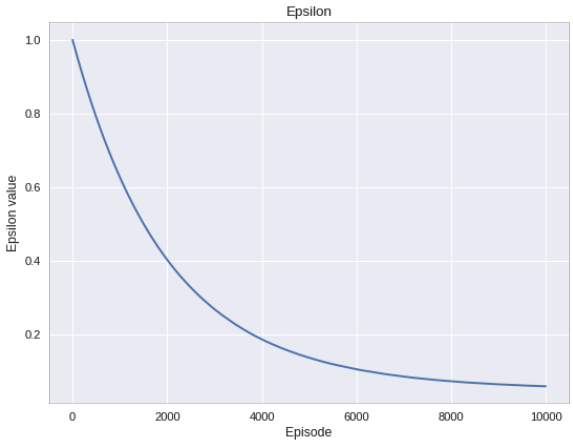


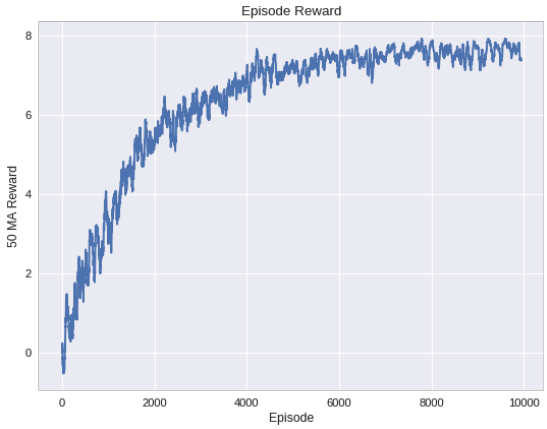
For 7500 episodes





For 10000 episodes

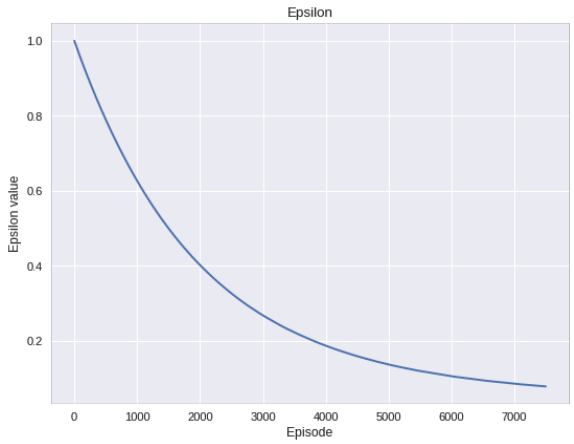


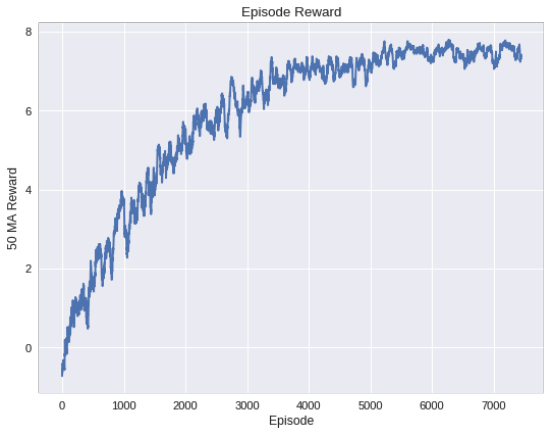


With the increase in the number of episodes, the epsilon decay takes the shape of an exponential curve and the mean reward keeps increasing exponentially. The time taken to train also increases with the increasing number of episodes.

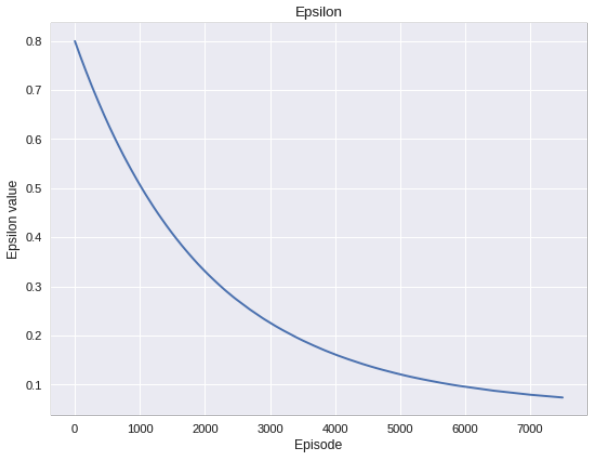
Keeping the number of episodes, Min_Epsilon, Lambda at 7500, 0.05, 0.00005 we change the Max_epsilon.

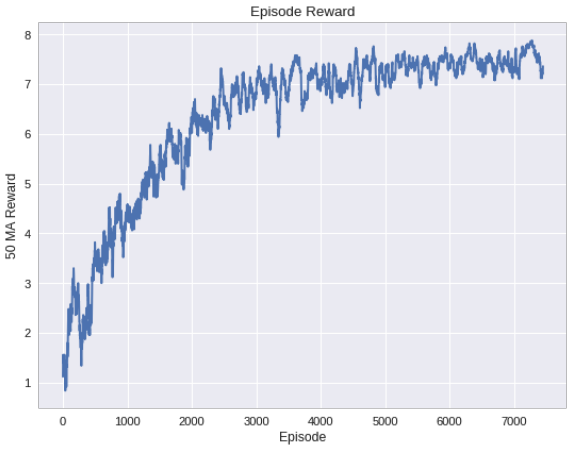
For Max_epsilon 1



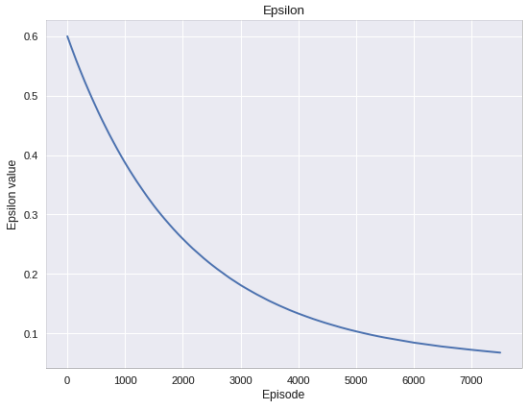


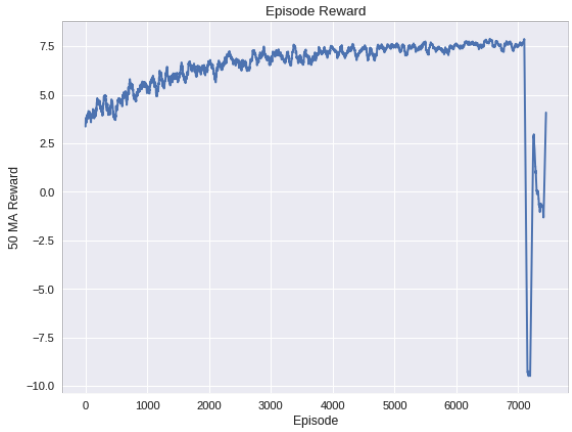
For Max_epsilon 0.8





For Max_epsilon 0.6

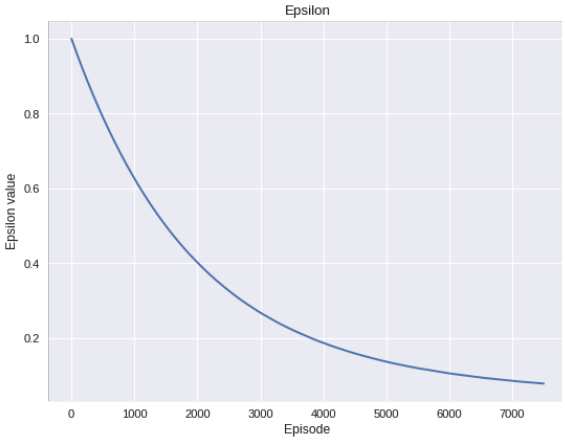


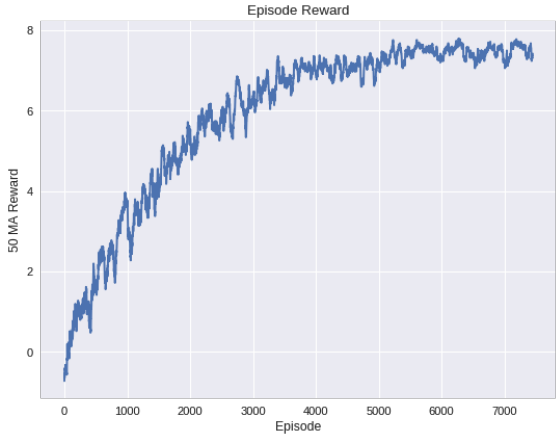


With the decrease in the Max_epsilon, the mean reward keeps decreasing.

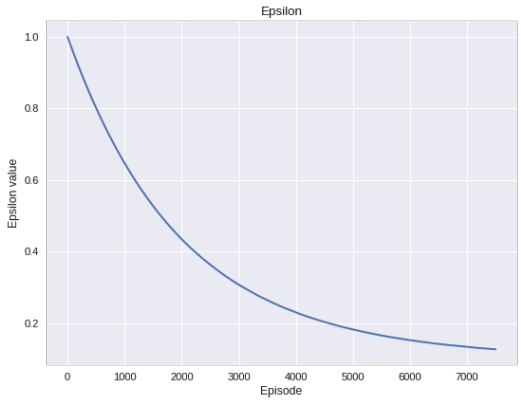
Keeping the number of episodes, Max_Epsilon, Lambda at 7500, 1, 0.00005 we change the Min_epsilon.

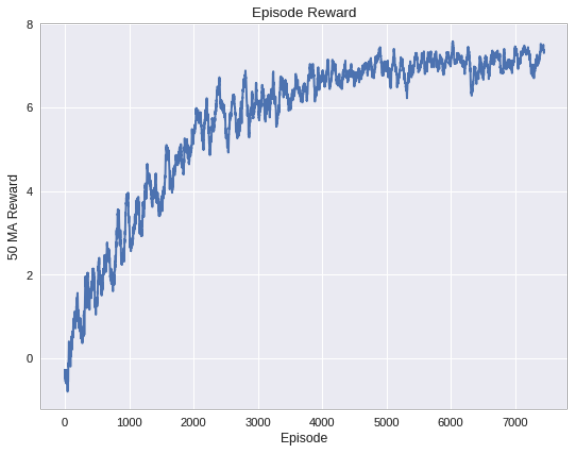
For Min_epsilon 0.05



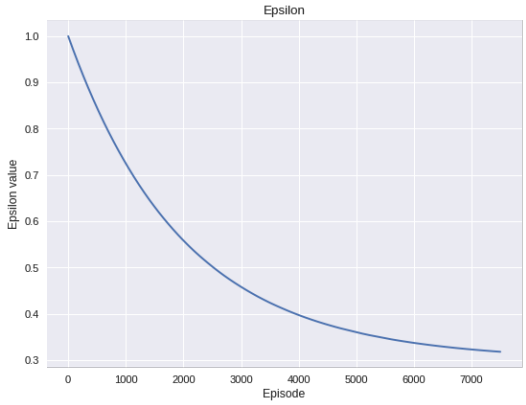


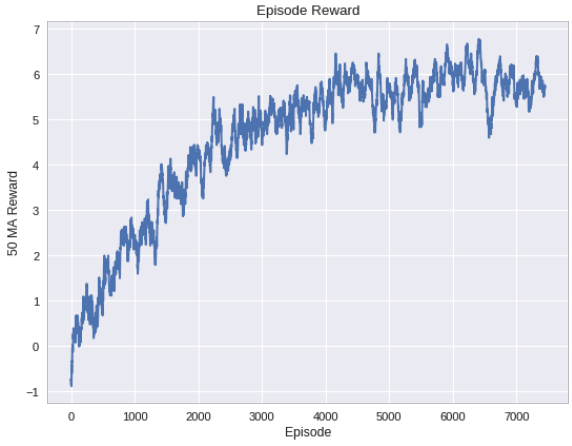
For Min_epsilon 0.1





For Min_epsilon 0.3

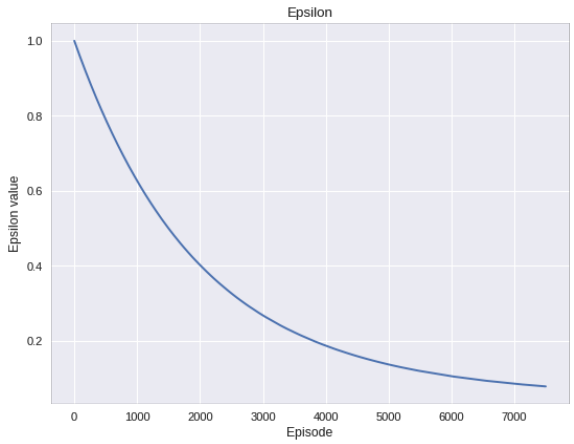


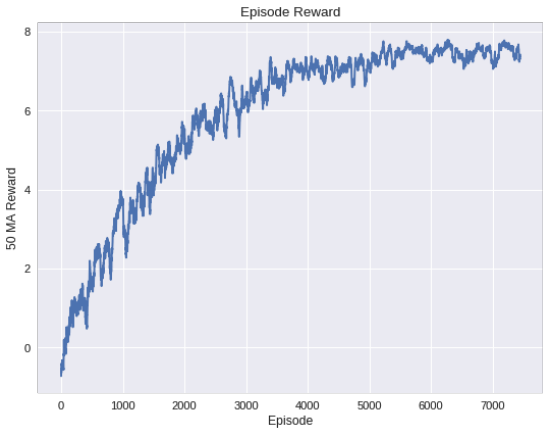


With the increasing Min_epsilon value, the mean reward value keeps decreasing.

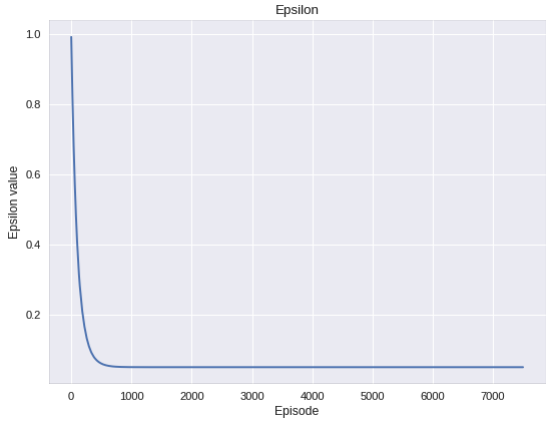
Keeping the number of episodes, Max_Epsilon, Min_epsilon at 7500, 1, 0.05 we change the lamba.

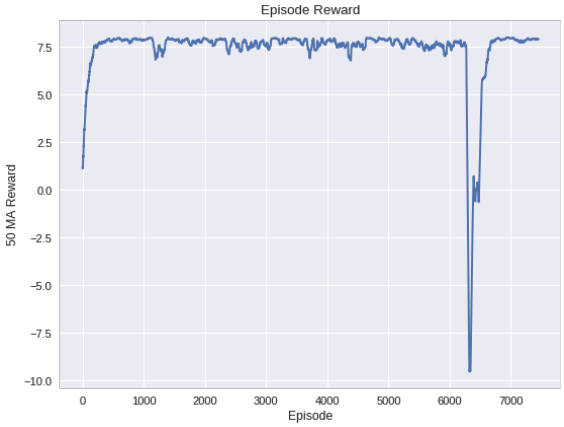
For lambda 0.00005





For lambda 0.001





With the increasing lambda value, the mean reward value keeps decreasing.

1.2 Writing Tasks

1.

If the agent always chooses the action maximizes the Q value, he chooses the path that he already chose to reach the goal state which is a greedy approach also known as the exploitation. It doesn't help the agent to get to know all the environment also known as the exploration. The approaches to avoid this are ϵ-greedy approach and Bayesian approach.

2.

As state S_4 is the final state, the Q values for all the possible actions is 0
$Q(S_4, U) = 0$
$Q(S_4, D) = 0$
$Q(S_4, L) = 0$
$Q(S_4, R) = 0$

By the Given equation $Q(S_t,a_t) = r_t + \gamma * \max_a Q(S_{t+1},a)$

For state S_3 the reward values are
$r_1 = -1$
$r_2 = 1$
$r_3 = -1$
$r_4 = 0$
$Q(S_3, U) = -1 + 0.99(1+0.99(1)) = 0.97$
$Q(S_3, D) = 1 + 0.99x0 = 1$
$Q(S_3, L) = -1 + 0.99\ (1+0.99(1)) = 0.97$
$Q(S_3, R) = 0 + 0.99(1) = 0.99$

For state S_2 the reward values are
$r_1 = -1$
$r_2 = 1$
$r_3 = -1$
$r_4 = 1$

$Q(S_2, U) = -1 + 0.99(1+0.99(1)+(0.99)^2(1)) = 1.94$
$Q(S_2, D) = 1 + 0.99x1 = 1.99$
$Q(S_2, L) = -1 + 0.99(1+0.99(1)+(0.99)^2(1)) = 1.94$
$Q(S_2, R) = 1 + 0.99x1 = 1.99$

For state S_1 the reward values are
$r_1 = 0$
$r_2 = 1$
$r_3 = -1$
$r_4 = 1$

$Q(S_1, U) = 0 + 0.99(1+0.99(1.99)) = 2.94$

$Q(S_1, D) = 1 + 0.99(1.99))= 2.97$
$Q(S_1, L) = -1 + 0.99(1+0.99(1)+(0.99)^2(1.99))= 2.9$
$Q(S_1, R) = 1 + 0.99(1+0.99(1)) = 2.97$

For state S_3 the reward values are
$r_1 = 0$
$r_2 = 1$
$r_3 = 0$
$r_4 = 1$

$Q(S_0, U) = 0 + 0.99(1+0.99(2.97))= 3.9$
$Q(S_0, D) = 1 + 0.99(1+0.99(1.99))= 3.94$
$Q(S_0, L) = 0 + 0.99(1+0.99(2.97))= 3.9$
$Q(S_0, R) = 1 + 0.99x2.97 = 3.94$

STATE	UP	DOWN	LEFT	RIGHT
0	3.9	3.94	3.9	3.94
1	2.94	2.97	2.9	2.97
2	1.94	1.99	1.94	1.99
3	0.97	1	0.97	0.99
4	0	0	0	0