Implementing DQN

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

1 Introduction

Deep Q-Network is one of the effective ways to train the reinforcement learning agent. It is highly effective when there are too many states and too many actions. It predicts the action on unseen state by the experience gained from the seen state using the Deep learning techniques.

$$Q(s, a, w) \approx Q^{\pi}(s, a)$$

2 Benefits

Replay in DQN :- with increase in size of memory, the learning experience of the agent increases.
Target Network in DQN :- Moving target is one of the biggest concern in DQN. The results fluctuate rapidly with moving target. Hence the concept of Target Network gives better results.

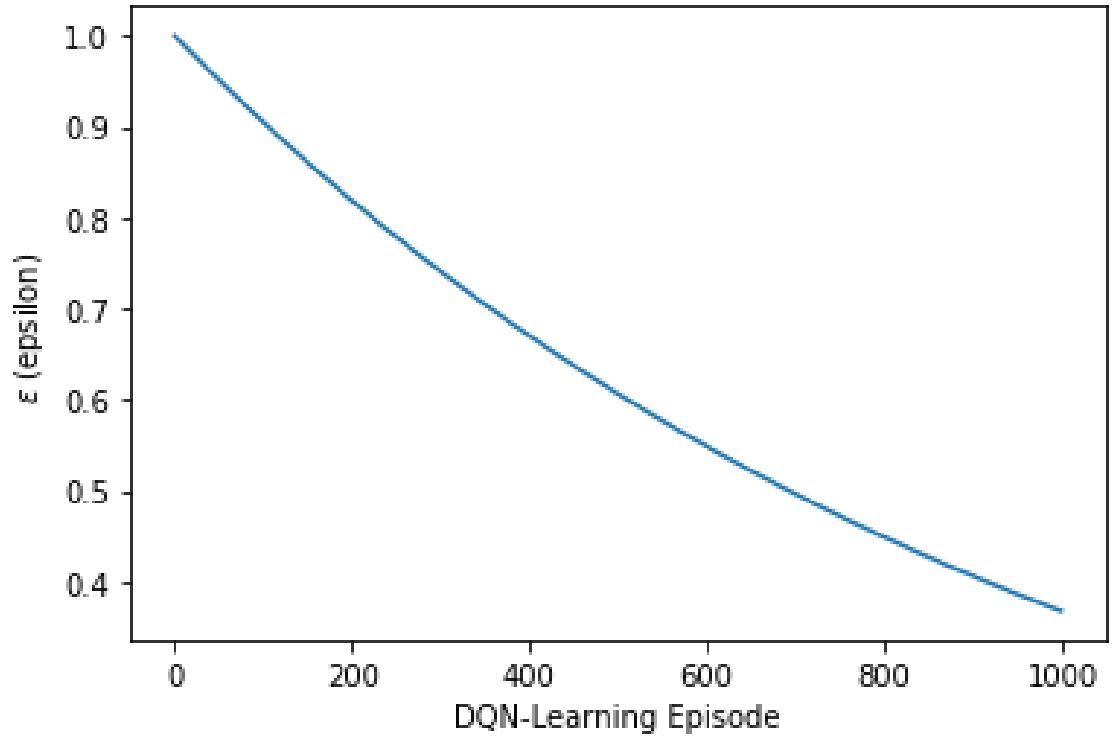
$$\mathcal{L}(w) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}} \left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w) \right)^2 \right]$$

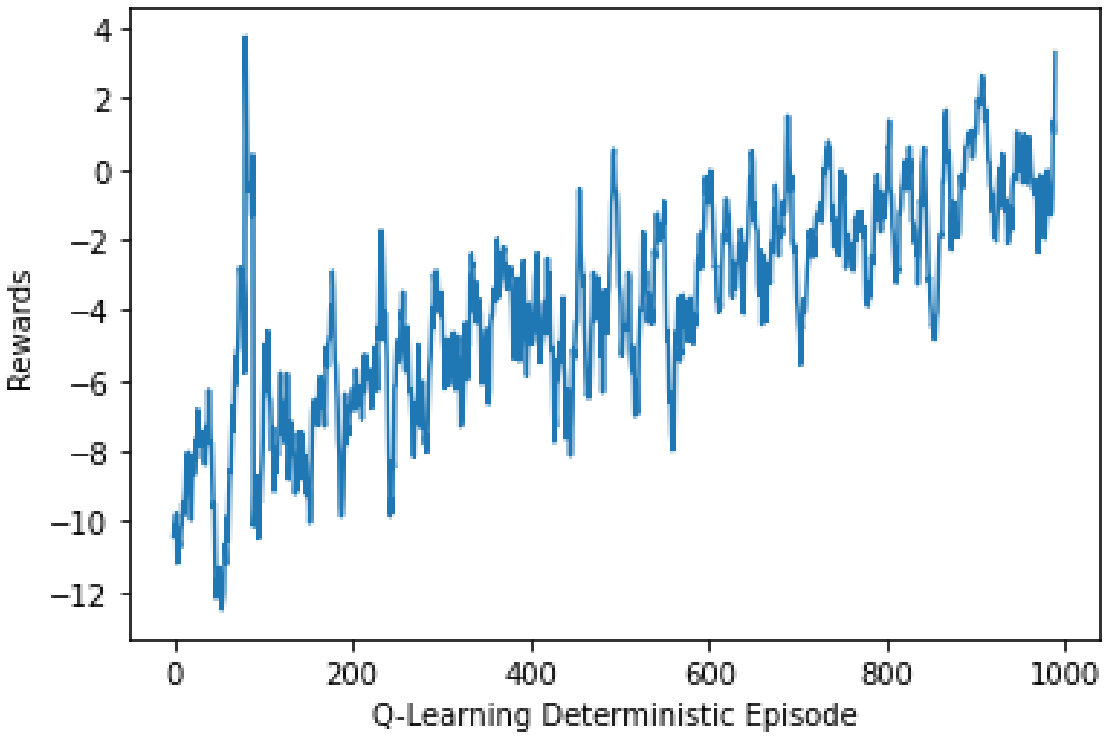
3 Grid World

we have chosen 6 by 6 grid world. It is a deterministic environment with 4 possible actions (Left,Right,Up,Down). Reward distribution is as follows
the first row :- 0.1,0.2,0.3,0.4,0.5
the last col :- 0.5,0.6,0.7,0.8,1
rest all are given -1
the 5,5 position is the target position with reward of 1
Objective :- The main Objective is to reach the end corner position (5,5)

4 Results

The entire environment is implemented with learning rate of 0.6 and discount factor of 0.9. We implemented deep Neural Networks of two layers each with 16 nodes. We used relu activation function, Adam optimizer and MSE loss function.


1.0
0.9
0.8
0.7
0.6
0.5
0.4
ε (epsilon)
0
200
400
600
800
1000
DQN-Learning Episode


4
2
0
−2
−4
−6
−8
−10
−12
Rewards
0
200
400
600
800
1000
Q-Learning Deterministic Episode