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. I implemented DQN, Double DQN and Dueling DQN on 6*6 Grid World,Cartpole and LunarLander.

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

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.

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

2 Dueling DQN

In Dueling DQN, we separate the estimator layer of neural network into two elements, using two new streams. One estimates the state value V(s) and other estimates the advantage for each action A(s,a) and combines back into a single Q-function at the final layer.

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \right)$$

3 Double DQN

The Double DQN is similar to Double Q-Learning. It has two Q estimators, Q1 estimates the best actions and Q2 evaluates Q1 for the selected action.

$$Q_1(s, a) \leftarrow Q_1(s, a) + \alpha(\text{Target} - Q_1(s, a))$$

Q Target: $r(s, a) + \gamma \max_{a'} Q_1(s', a')$

Double Q Target: $r(s, a) + \gamma Q_2(s', \arg\max_{a'}(Q_1(s', a')))$

4 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)

5 Cartpole

Cartpole has two actions 0 to push left and 1 to push right. It gets a reward of 1 for every step taken, including the termination step. The threshold is 475.
Objective :- The objective is to make the pole stand for as long as possible.

Num	Observation	Min	Max
0	Cart Position	-2.4	2.4
1	Cart Velocity	-Inf	Inf
2	Pole Angle	~ -41.8°	~ 41.8°
3	Pole Velocity At Tip	-Inf	Inf

Figure 1: Catrpole Observation

6 LunarLander

Landing pad is always at coordinates (0,0). Coordinates are the first two numbers in state vector. Reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If lander moves away from landing pad it loses reward back. Episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame.
Objective :- The main Objective is to land on the launch pad in between the flags.

7 Results

We have taken same epsilon decay algorithm. The minimum epsilon is 0.1

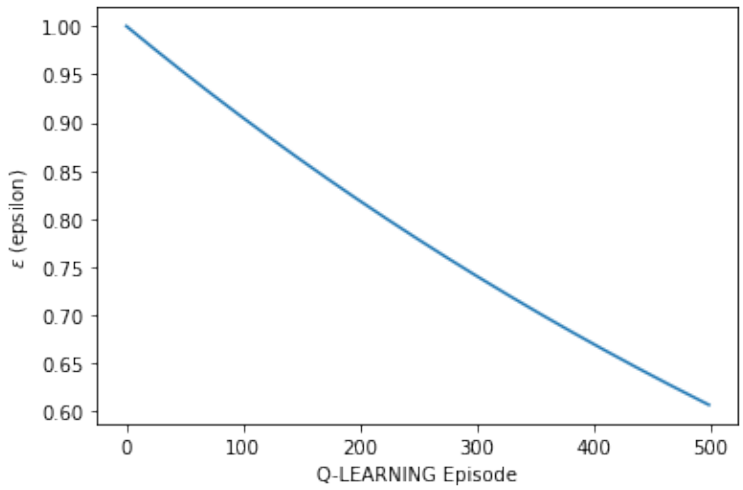

Figure 2: Epsilon decay for all Environments

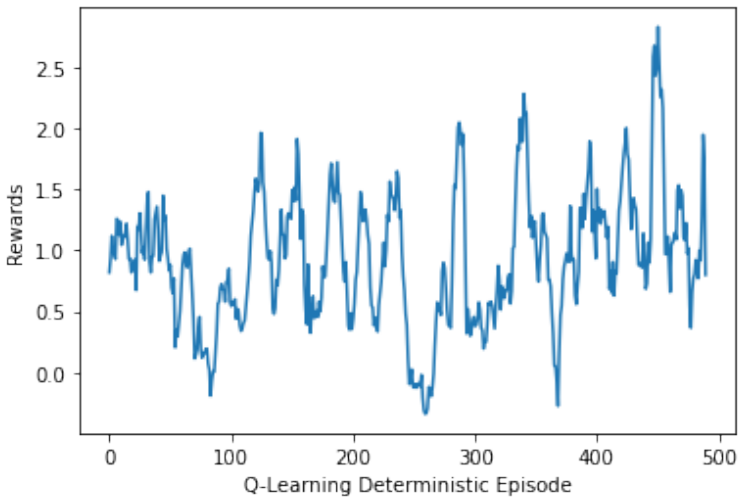

Figure 3: Gridworld DQN

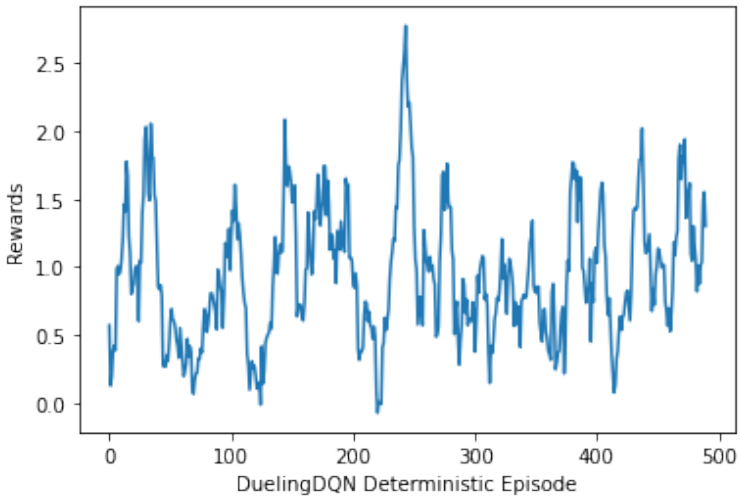

Figure 4: Gridworld Dueling DQN

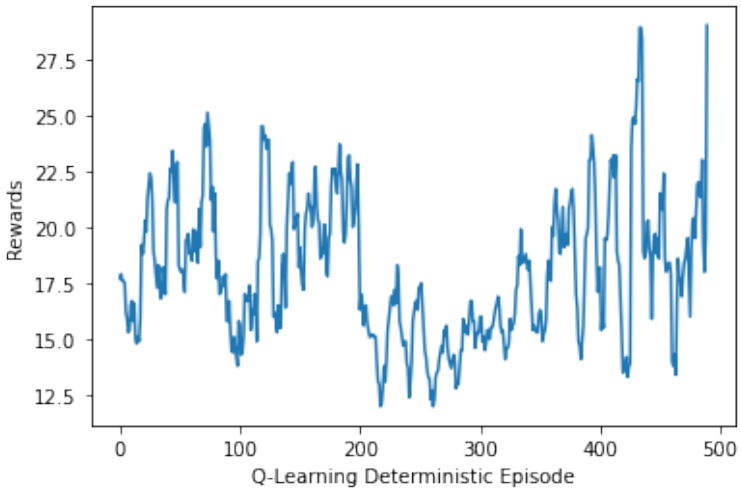

Figure 5: Cartpole DQN

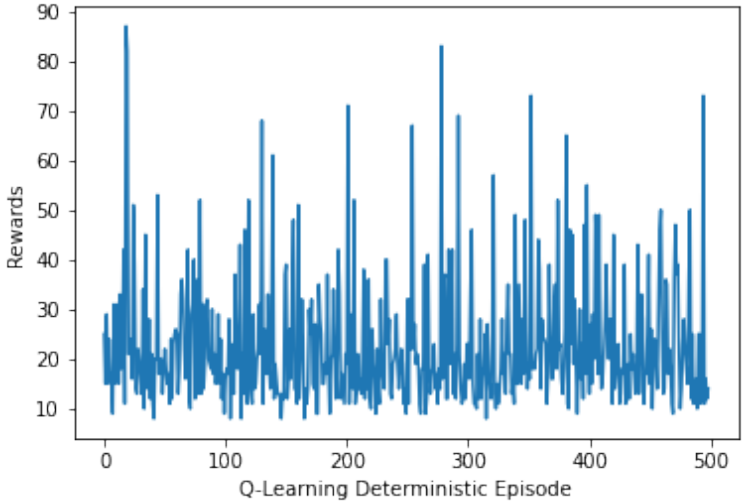

Figure 6: Cartpole Dueling DQN

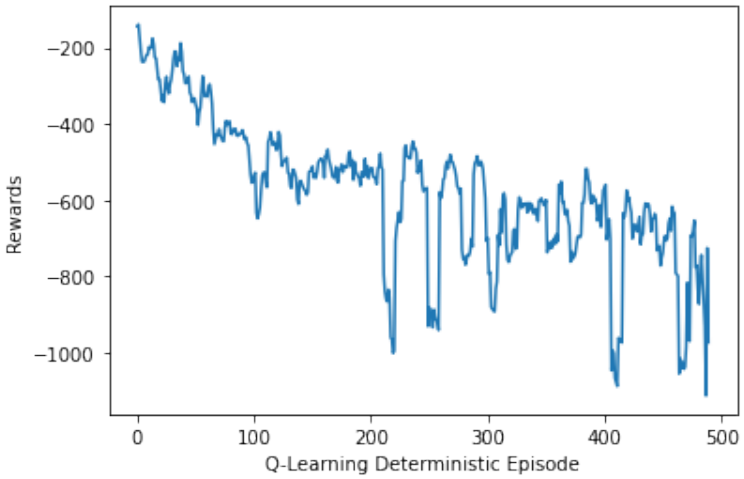

Figure 7: LunarLanding DQN

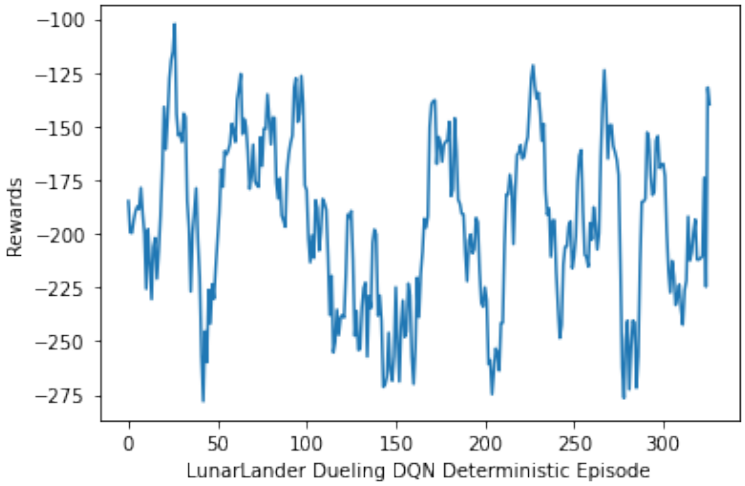

Figure 8: LunarLanding Dueling DQN