# UNIVERSITY OF TEHRAN


**COLLEGE OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**


# NEURAL NETWORK & DEEP LEARNING

EXTRA ASSIGNMENT


SIAVASH SHAMS

810197644

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF TEHRAN


***May. 2022***

# 1 CONTENTS

## 2 LUNAR LANDER

# A.

State Space: The state space is 8-dimensional and (mostly) continuous, consisting of the X and Y coordinates, the X and Y velocity, the angle, and the angular velocity of the lander, and two booleans indicating whether the left and right leg of the lander have landed on the moon.

Action Space: In the discrete version, the agent can take one of four actions at each time step: [do nothing, fire engines left, fire engines right, fire engines down].

Reward System: The agent gets a reward of +100 for landing safely and -100 for crashing. Episode finishes if the lander crashes or comes to rest. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. It receives negative rewards for moving away from the landing site, increasing in velocity, turning sideways, taking the lander legs off the moon and for using fuel (firing the thrusters). The best score an agent can achieve in an episode is about +250.

# B.

***The model does not converge (reaching 200 average rewards) with given parameters within 250 episodes, so we change the learning rate to 0.0005 and epsilon decay to 0.99 so the model converges within 250 episodes.**

Mathematically, the regret is expressed as the difference between the payoff (reward or return) of a possible action and the payoff of the action that has been actually taken. If we denote the payoff function as u the formula becomes:

$$regret = u(possible\ action) - u(action\ taken)$$

Figure1 represents an intuitive explanation of regret

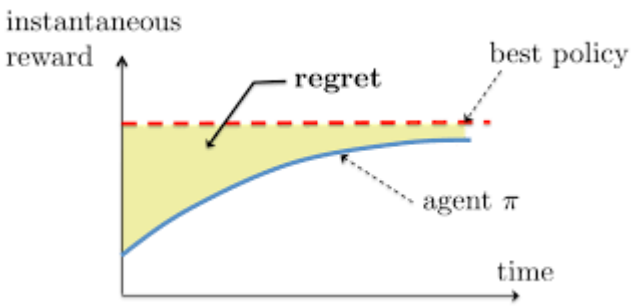


Figure1. visual representation of regret

Orange line represents the average reward needed to solve the environment. We sketched it to compare regret and convergence speed.

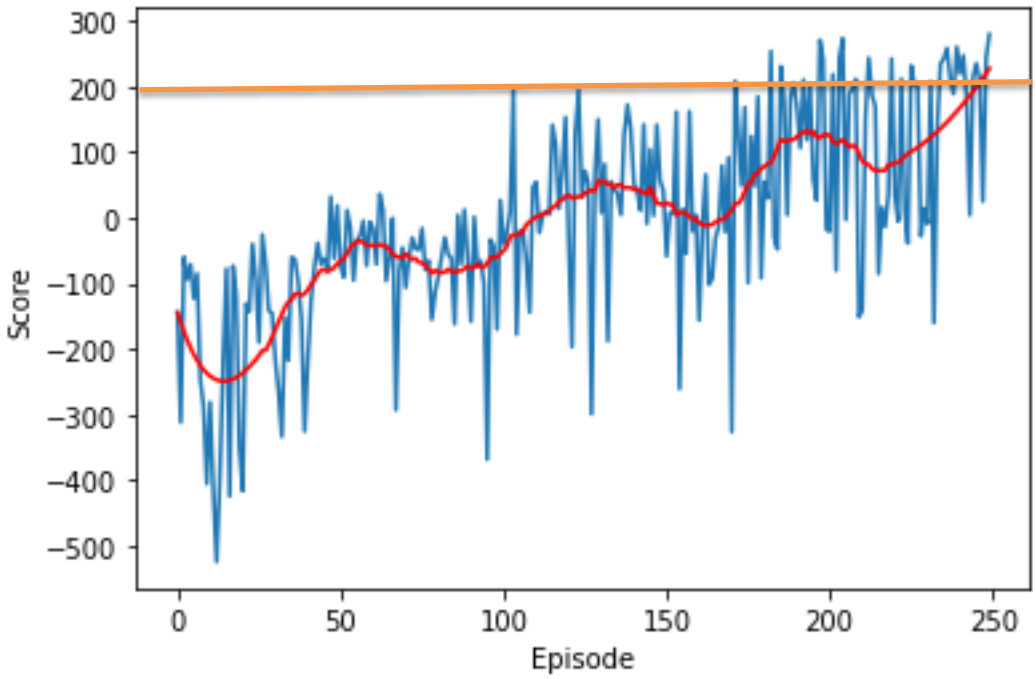


Figure2. Average score vs episode with 32 batch size

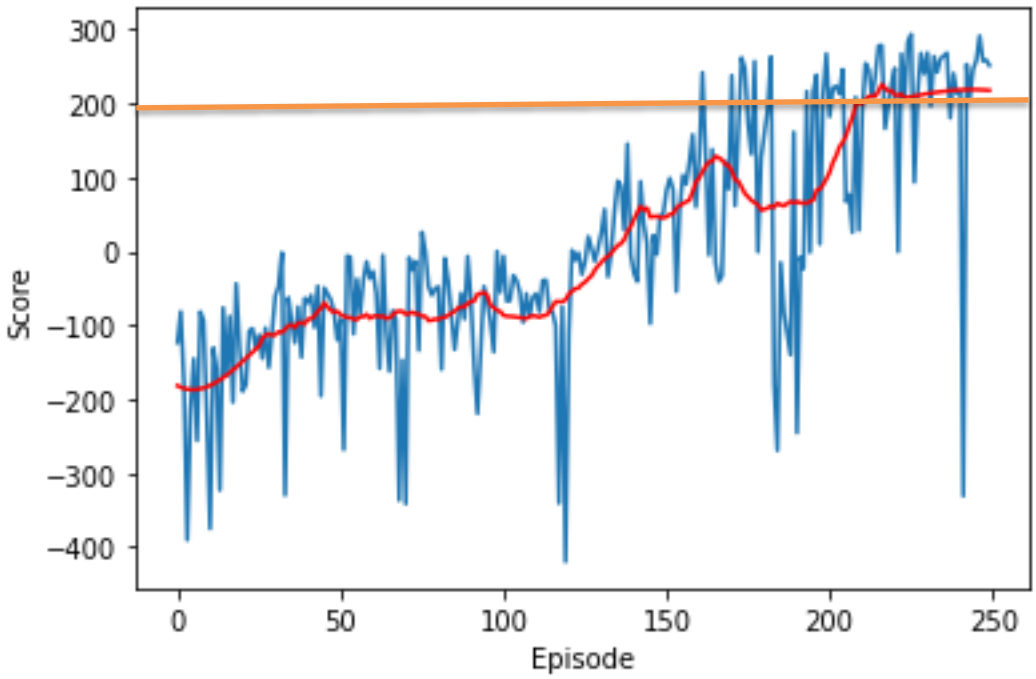


Figure3. Average score vs episode with 64 batch size

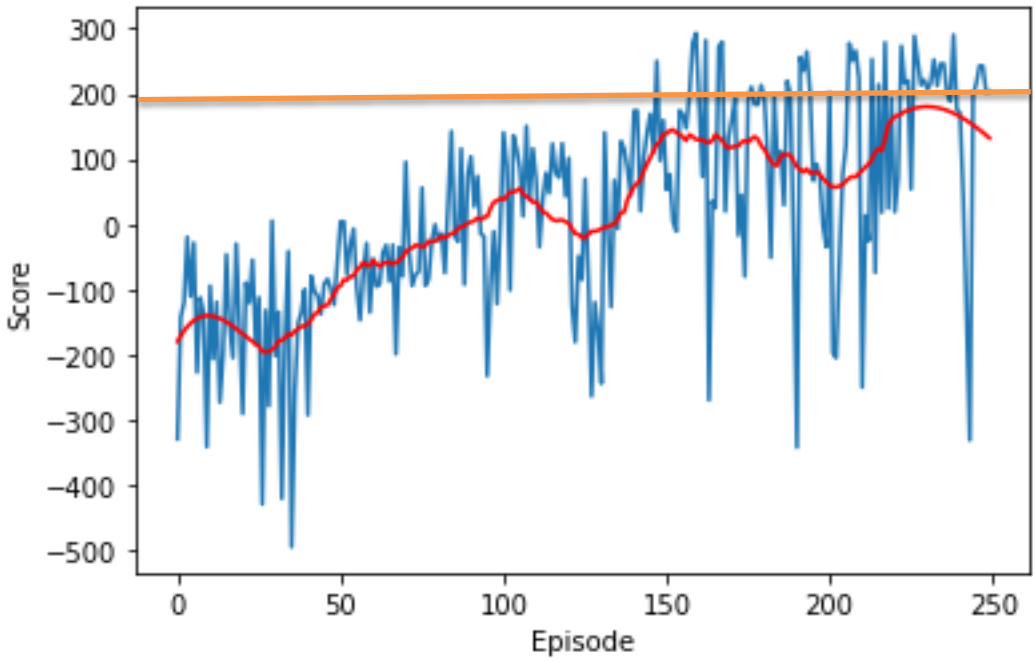


Figure4. Average score vs episode with 128 batch size

By looking at the figures we can see that with batch size 32 the environment is solved in about 250 epochs, with batch size 64 the environment is solved in about 210 epochs and with batch size 128 the environment does not converge within 250 epochs.

Also, if we take a look at the area between instantaneous rewards and maximum reward, we can see that the regret is the most with batch size 32 and the model with batch size 128 has the least regret

In overall, we choose a batch size equal to 64 because it converges the fastest and has reasonable regret.

## C.

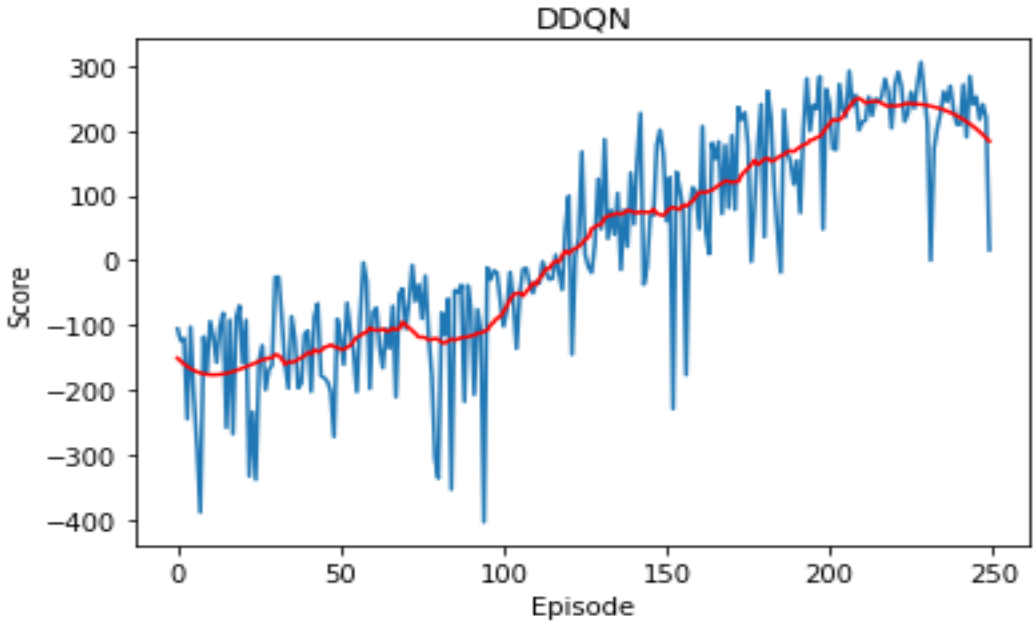


Figure5. Average score vs episode with 64 batch size DDQN

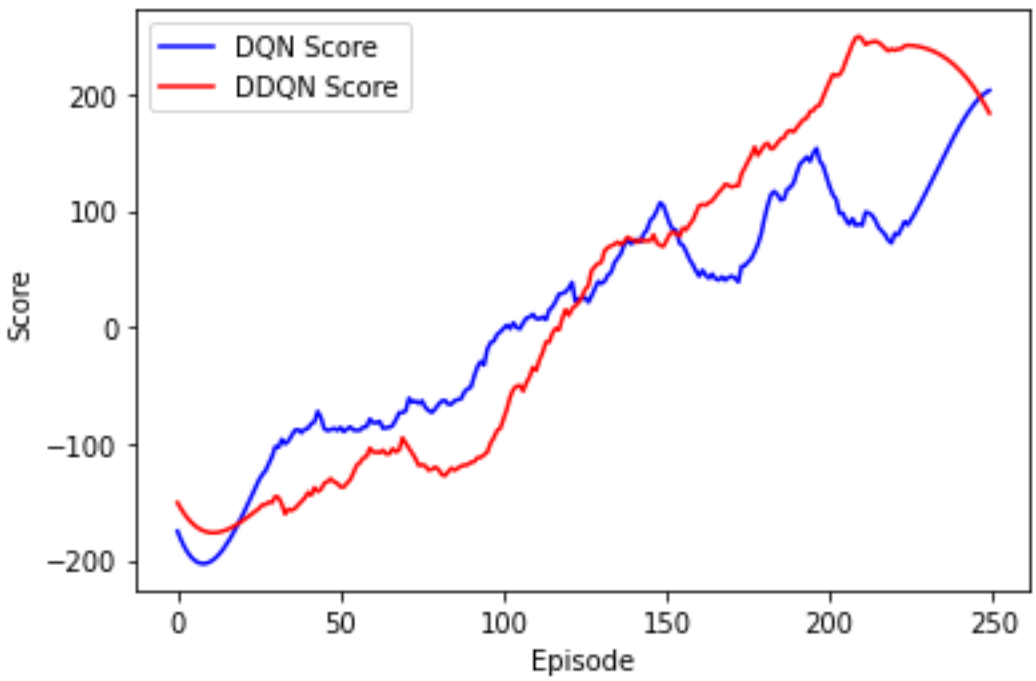


Figure6. DQN vs DDQN average scores against episode

As we can see from figure 6 DDQN reaches 200 average reward faster but DQN have lower regret. Also In the videos we can see that in epoch 100 DQN agent has better performance than DDQN agent