Lunar Lander DQN Report

Overview of DQN

Deep Q- networks (DQN) are a reinforcement learning technique where a deep neural network is used to approximate the Q-value function. This function estimates the expected future rewards for taking a specific action in a given state. For this project I implemented a DQN agent to solve the lunar lander environment from gymnasium. In the DQN model there are multiple parts to it.

DQN algorithm techniques

- Q – network with fully connected layers

In Q learning a Q table is used to store the expected value of taking an action in a specific state but that does not work with a continuous environment like the lunar lander. For that specific reason we use a neural network to approximate the Q – function. The Q network is a fully connected neural network that will take the current state as the input and outputs the Q-value for each possible action.

- Target network

The next network that was used was the Target network which is a second neural network that works the same as the main Q-network and is used to compute the target Q-values during training but the difference with this one is that it is updated less frequently. It is needed because without it the Q-values could become unstable since the target and the network would shift simultaneously.

- Experience Replay Buffer

The experience replay buffer is fixed-sized memory that stores past experiences as tuples in order improve sample efficiency. It does this by collecting experience over time through the replays and then randomly samples a small number of past episodes for training.

- Epsilon-greedy exploration

The other technique that was implemented was the Epsilon-Greedy Exploration which is a strategy for balancing exploration like trying new actions and exploitation such as choosing the best-known action. The way it works is that it with a probability e it will take a random action. Then with probability 1 – e, takes the best action according to the Q-network. The probability starts high and gradually decreases over time to favor exploitation later in training.

- Online training loop

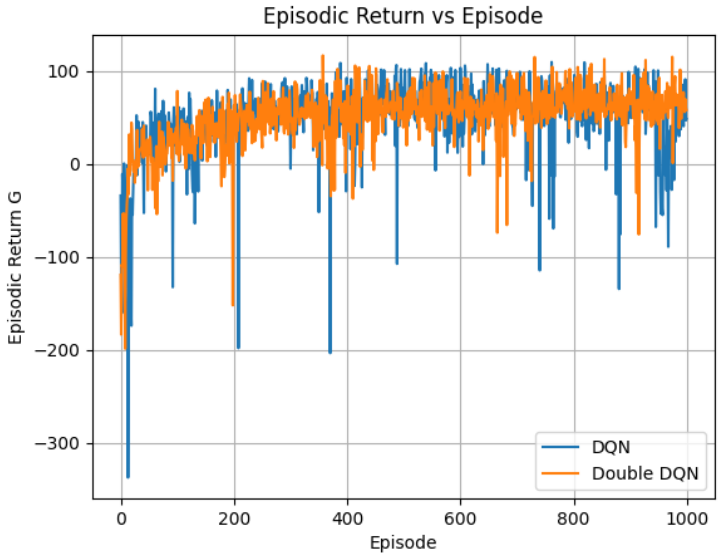
The training loop consists of the following steps. First it will select an action using the epsilon greedy exploration, then it steps through the environment with that action and stores this exploration in the replay buffer. From this buffer it will sample a small number of experiences to then update the Q-network.

DQN Extension

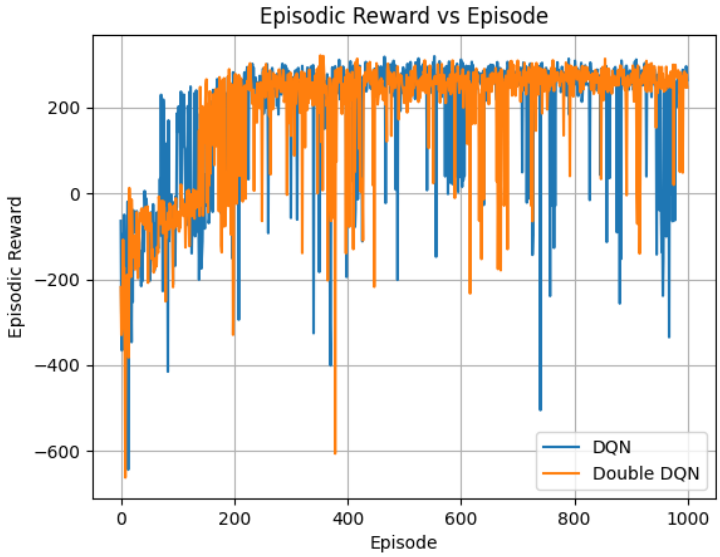
The extension that I chose for this was the Double DQN. The importance of Double DQN is that It helps in the overestimation of the Q values in regular DQN. In DQN the maximum predicted Q-value to calculate the target value which could lead to overestimation especially with the lunar landing which has high variability. DQN uses the target network to both select and evaluate the maximum actions while Double DQN separates these two where the online network will select the next best action and the target network will evaluate that action by providing a stable estimate to how good that action is which usually results in more stable learning.

Results and Insights

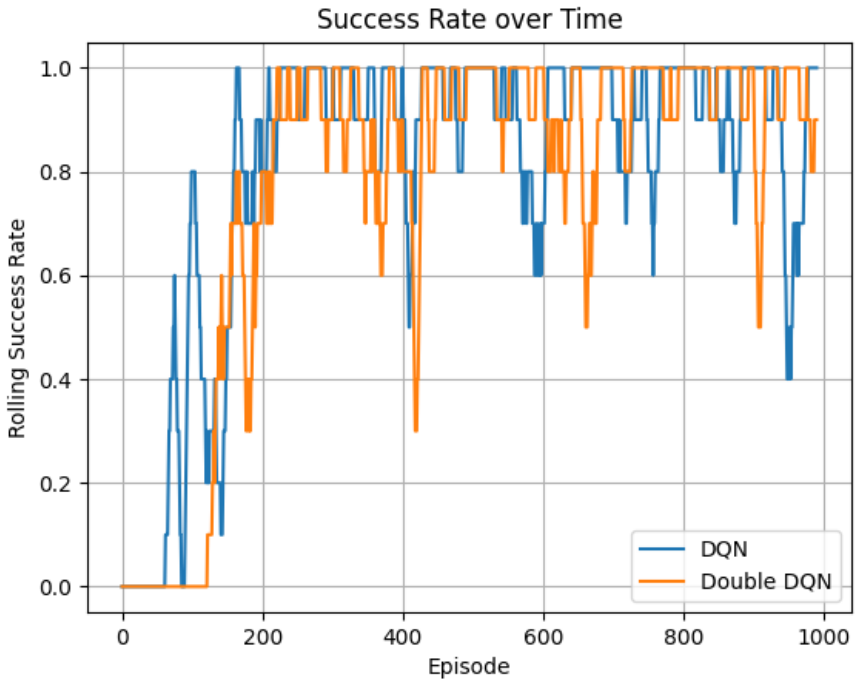
Episodic Return vs Episode



The above graph shows the episodic return vs episode. From this graph we can see that the DQN learns fast and stabilizes early one which is why it is more consistent throughout the episodes. The Double DQN on the other hand starts off more volatile but then stabilizes towards the end which is normal with Double DQN as it is avoiding overestimation.



The above graph shows the episodic reward vs return giving us an insight in how well the agents are performing after each episode through their rewards. We can see that DQN learns faster early on while double DQN struggles early. The both end up converging but DQN has lower dips meaning it's more volatile while Double DQN become more consistent and stable.



The above graph shows the success rate over time and as we can see DQN started going up a lot earlier that Double DQN but again it did not remain as consistent towards the end like Double DQN did.

Average over last 100 episodes:

Metric	DQN	Double DQN
Avg Reward	212.06	243.80
Success Rate (%)	86.0%	91.0%

The above table shows the summary of the last 100 episodes and we can see that the Double DQN has a higher success rate and avg reward that the regular DQN which just shows the importance of Double DQN and how it addresses the overestimation bias.

Overall though we have seen that DQN quickly achieves a high performance but then slows down and occasionally regresses. Double DQN at first starts off slow but ends up either matching or surpassing DQN's performance in both stability and success rate which shows how its reduced overestimation bias helps it in terms of performance.

Contributions : Emily Zapata

Github : https://github.com/EmilyAZ/lunar-lander-dqn.git