NAIVE REINFORCEMENT LEARNING ON ATARI GAMES

Mentor :- Dr. Manish Kumar Bajpai (Assistant Professor, CSE Dept. IIITDM Jabalpur)

Pravesh Vajpayee (2018194)
Kshitij Kumar Singh (2018124)
Raj Agrawal Mohit (2018201)

UPGRADATION REPORT

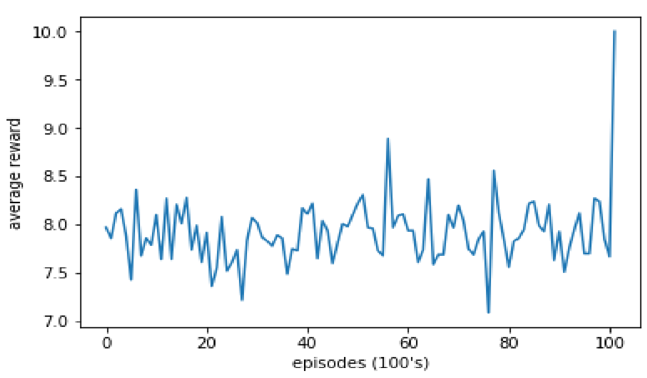
BRIEF INTRODUCTION AND PROBLEM STATEMENT

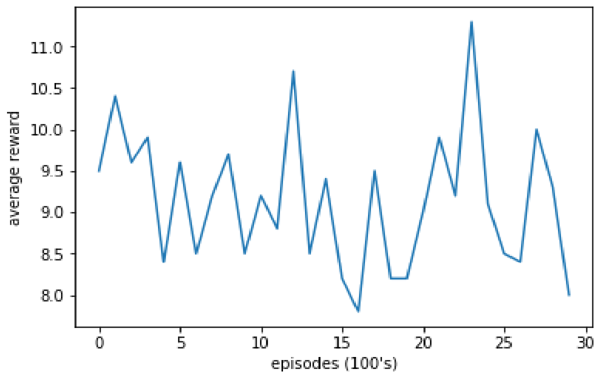
In this project , Two RL algorithms we have selected to compare their performance in this project are SARSA, naive Q-learning on different Atari games, and analyze them in order to identify factors which might influence the performance of these algorithms on different types of environments.

RESULT BASED ON OUR PREVIOUS IMPLEMENTATION

In the previous Report submitted we have concluded that SARSA seems to converge much faster than Naive Q in this case. This difference in convergence rate can be explained by the effect of epsilon-greedy implementation, where SARSA converges faster as it takes the epsilon probability of random action into account when updating its policy.

With either algorithm, there does not seem to be any difference between their initial performance and final performance. Our hypothesis is that the problem may lie in the non-markovian nature of this environment, as our implementation only takes into account the current state when deciding which action to take.



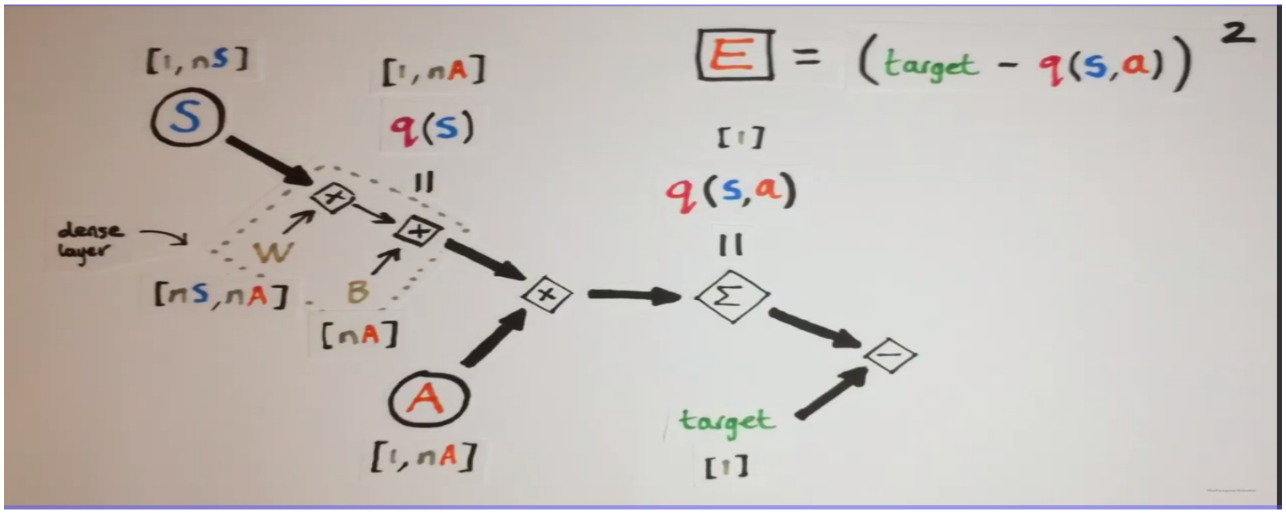


Q Learning Reward Vs Episodes plot **SARSA Algorithm Rewards Vs Episodes**

IMPROVEMENT

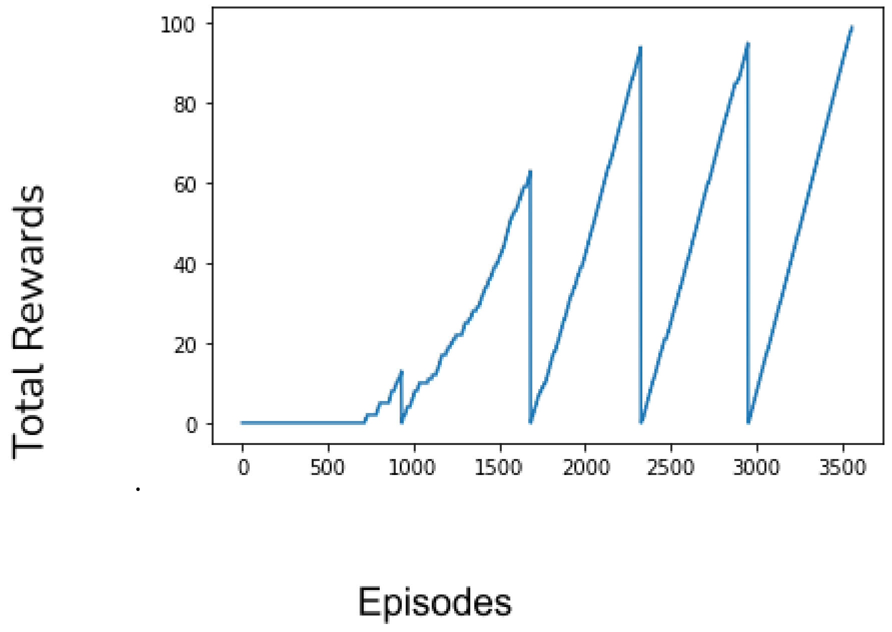
We can improve the performance by Deep Q-Learning using Experience Replay further supports our hypothesis, as using Experience Replay allows it to use a history of states to determine action, solving the problem of non-markovian property as mentioned in our previous report submitted.

Q learning with Neural Network in Tensorflow finds optimal policy a little bit faster than the other agent which uses the q table to store q values and it is more stable than normal Q learning approach.



E is loss function in above image

Total Rewards Vs Episodes in Q Learning with Neural Network



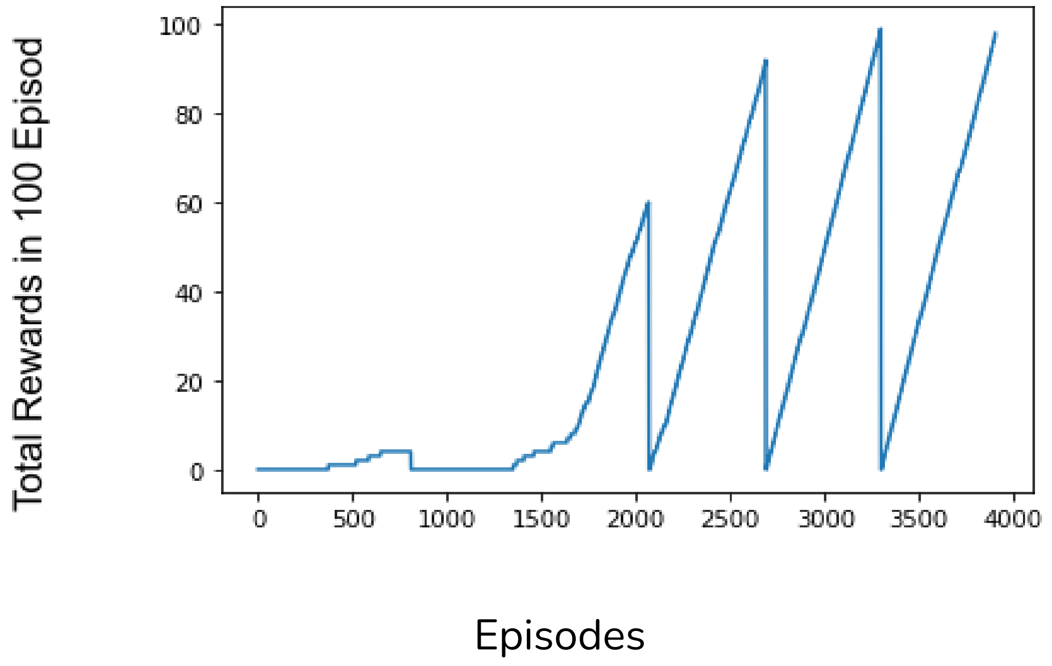
Now further improving it by using Fast Q Learning with Experience Replay with Neural Network which will further improve the efficiency of Training in Q learning.

Earlier we were using a single experience tuple only once and forget before getting the next experience tuple. So its drawback is that it creates bias towards training on common experiences and negatively affects the rare experiences which are later useful sometimes. So experience Replay gives the chance of training on rare experiences in case they are not seen in a while.

So we have created a deque with a fixed length 1000 and stored the experiences and sampling it in batches for further training . After we have trained then we push out the oldest experience to create space for new experience and similarly continue.

So we can see that our model learns in a **more stable** way then unlearning it at some point. So using Experience replay with Neural Network in Q learning helps to speed up the training efficiency .

Total Rewards Vs Episodes in Experience Replay in Q Learning with Neural Network



So we can see that it is an upgradation over Neural Network by speeding up the training efficiency

REFERENCES

[1] Andrew G.Barto Richard S.Sutton, Reinforcement Learning An Introduction, The MIT Press, 2018.

[2] “Implementing deep reinforcement learning models with tensorflow + openai gym,” 2018. [6] “Playing atari with deep reinforcement learning,” 2013.

[3] https://www.geeksforgeeks.org/

[4] www.cse.unsw.edu.au

[5] https://towardsdatascience.com/deep-q-network-dqn-ii-b6bf911b6b2c

[6] https://www.youtube.com/channel/UCUbeqkIVP808fEP0ae-_akw

[7]https://stackoverflow.com/