Università della Svizzera italiana

Institute of Computational Science ICS

Deep Learning Lab **2019**

Student: Sanchit Chiplunkar

Report for Assignment 4

Due date: 20 December 2019, 23:59

Answer 1.4.1

FrameStack: In the DQN paper the observation space is of the last 4 images so as to give the information about the dynamics of object and this wrapper helps in making the stacking of images easier.

MaxAndSkipEnv wrapper combines the repetition of action during K frames and pixels from two consecutive frames.

ScaledFloatFrame wrapper converts observation data to floats and also convert all the pixel value in the range of [0,1].

ClipRewardEnv: This wrapper clips the reward to +1 ,0 or -1 value depending on the sign of reward achieved.

Answer 1.4.2

The Authors main intention of introducing target network was to improve the stability of the Q-learning when using neural networks(function-approximation) as they are prone to instability. As author mentions in standard online learning an update which increases $Q(s_t,a_t)$ can also increase $Q(s_{t+1},a)$ for all a hence increasing the target y_j and thus leading to oscillations or divergence of the policy. In the author's word generating the targets using an older set of parameters adds a delay between the time an update to Q is made and the time the update affects the targets y_j, making divergence or oscillations much more unlikely.

Answer 1.4.3

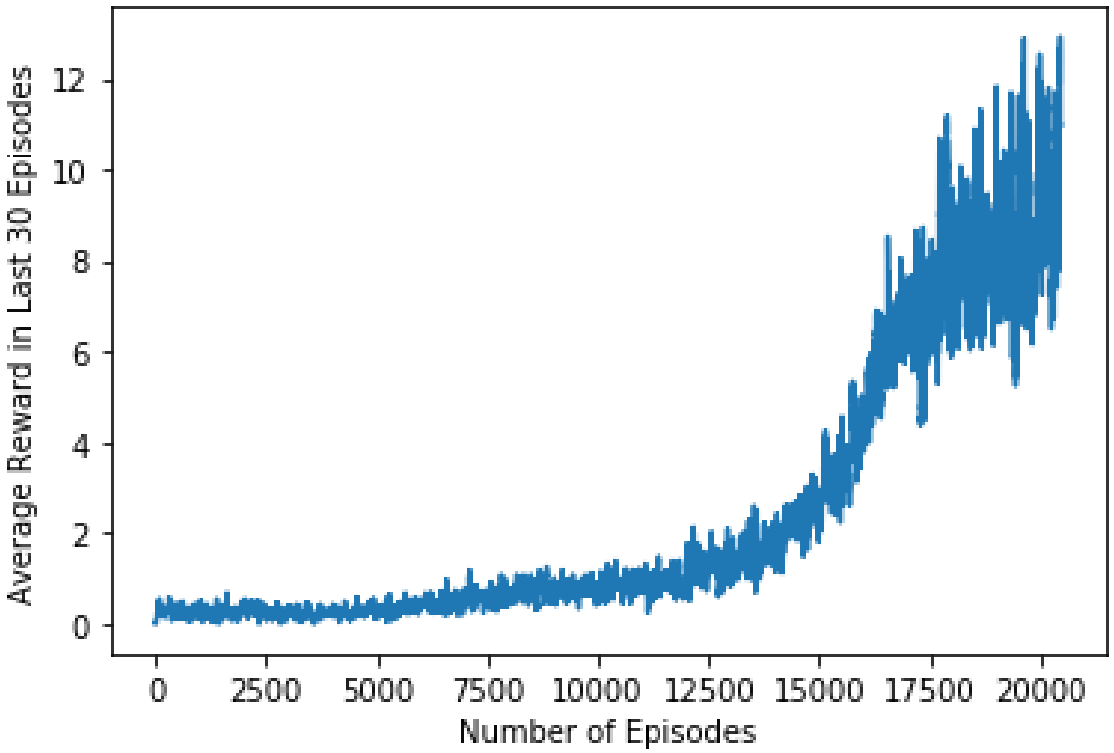
The ε-greedy policy helps us in giving trade-off in Exploitation and exploration.

There are two ways to take an action either to choose it randomly or chose the action with highest Q value. Now initially when we start our training our model hasn't decided the strategy to maximise the reward it is better to explore different actions even if we don't get decent reward

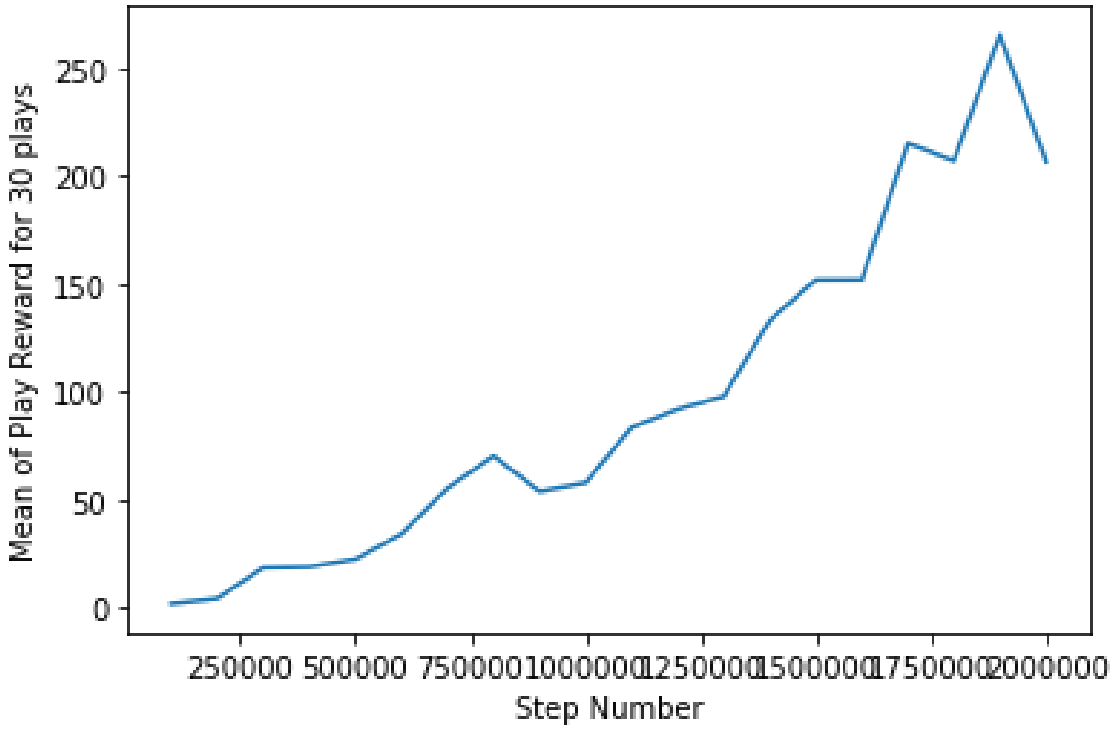
taking those action and as our training progresses our model learns the strategy it is better to explore less and exploit on our current strategy to reap better rewards which is what our $\varepsilon-$greedy policy does, initially our epsilon is set to high so as to increase exploitation and with every iteration is reduced by a small quantity till we reach the minimum epsilon decided as our hyper-parameter value. After minimum epsilon we train our model more to learn the best model to get a better reward which wasn't possible just by using the greedy policy.
Apart from this the epsilon method is even done during the testing/evaluation phase and the main reason for that was to minimise the possibility of overfitting.

Answer 1.4.4

Answer 1.4.4.a
Below is the plot for return per episode averaged over the last 30 episodes as was asked in the question:

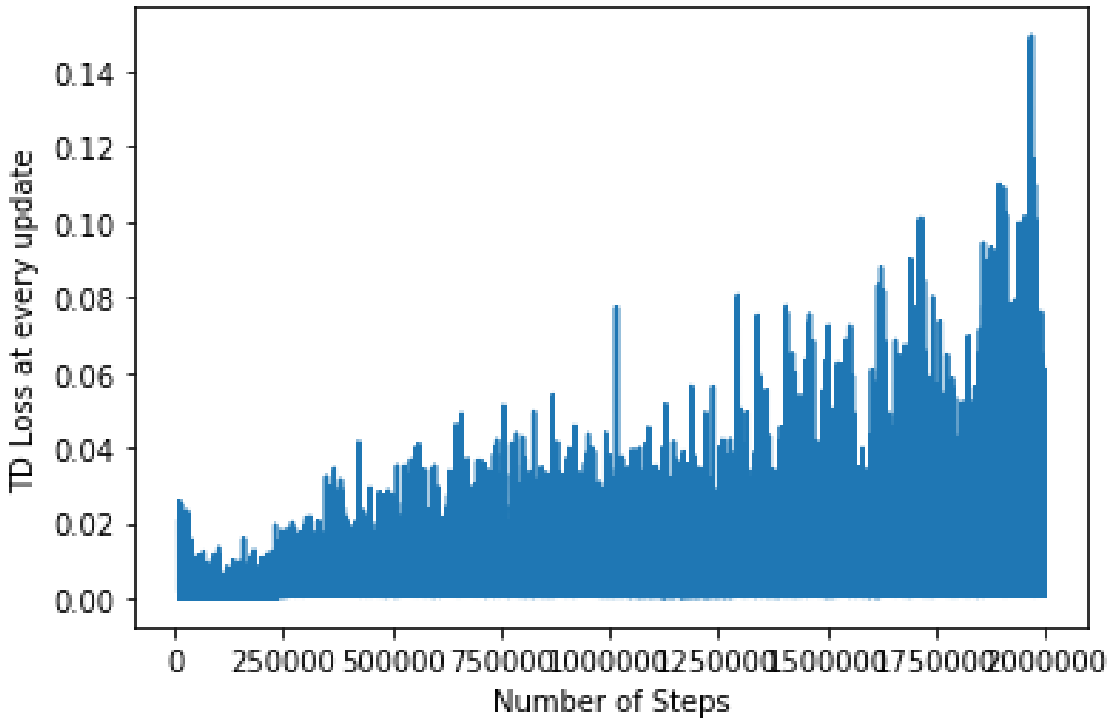


Answer 1.4.4.b
Below is the plot for return for average scores summed across as was asked in the question:



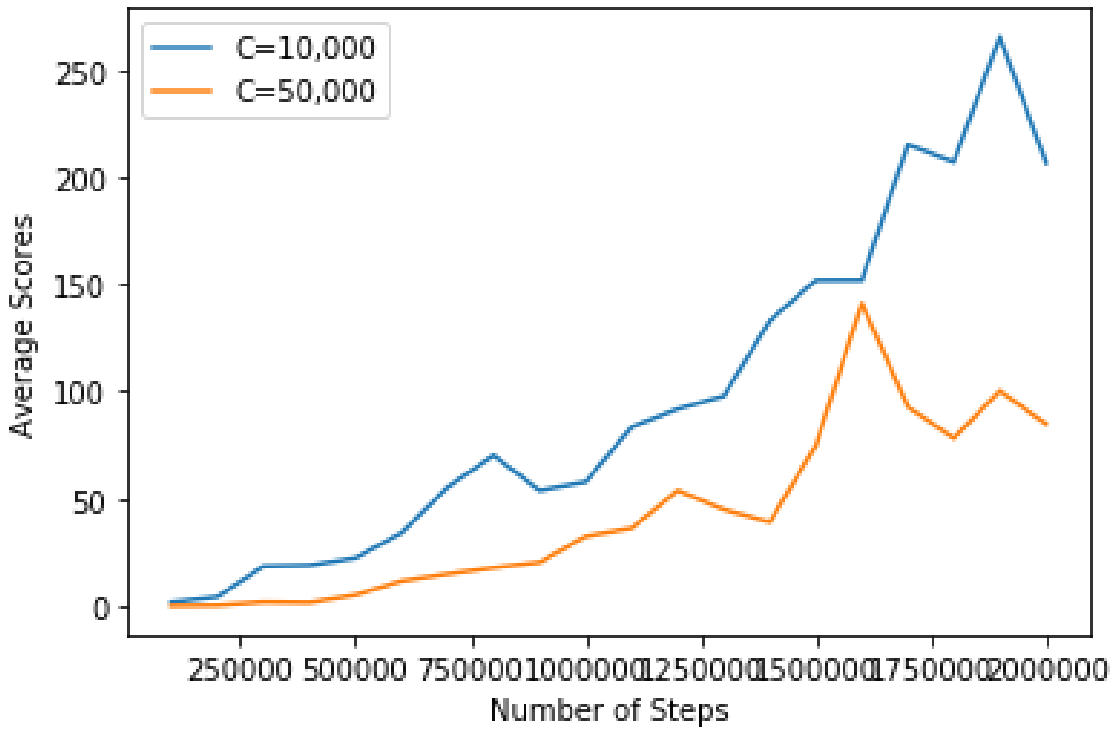
Answer 1.4.4.c

Below is the plot for return for Temporal Difference Error at each update step as was asked in the question:



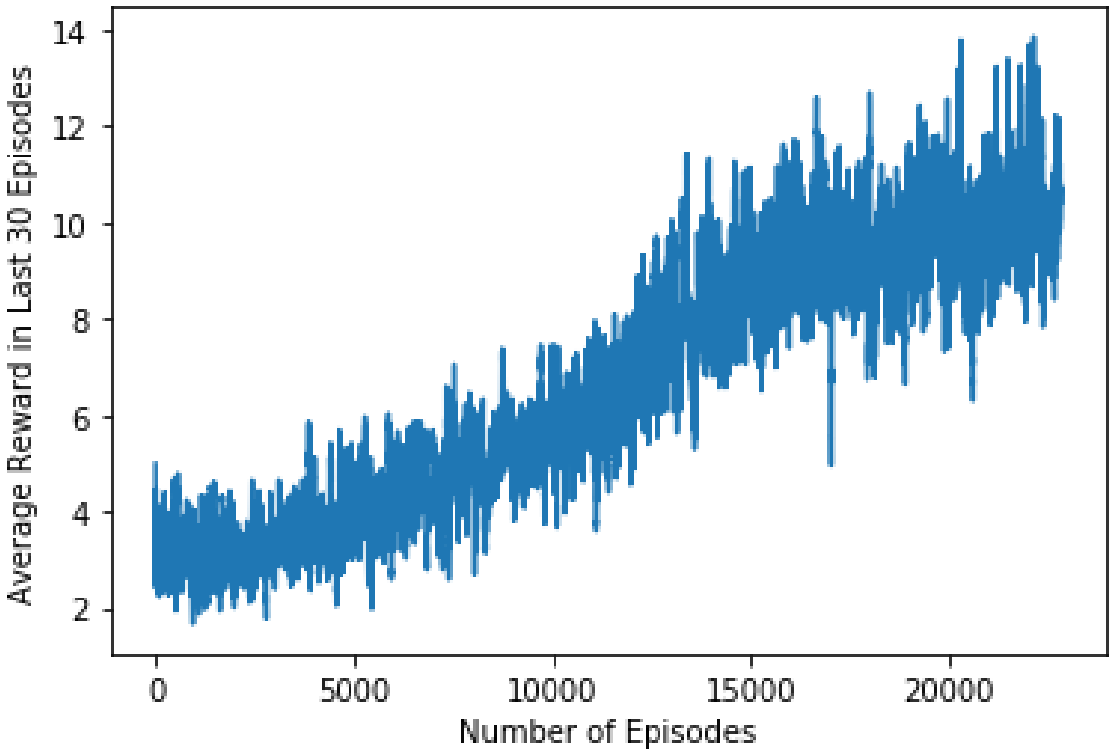
Answer 1.4.5 The mp4 file is already attached in the zip folder with the name monitor_Breakout.mp4. In this video as can be seen the agents tries to and succeeds in sending the ball on one of the side continuously so as to send the ball on the top of the stack and get maximum reward.

Answer 1.4.6
Below is the comparison plot of Average scores when C=10000 vs when C=50,000:



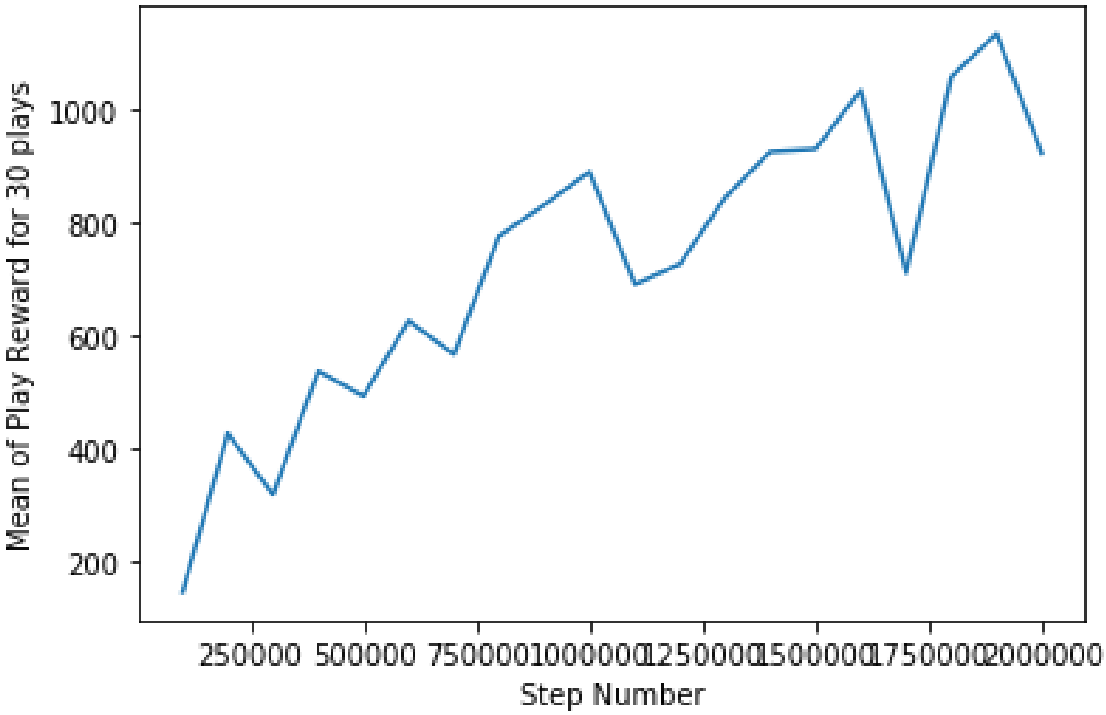
Answer 1.4.7 I have trained on the Atari game **Space Invaders** for this question

Answer 1.4.7.a
Below is the plot for return per episode averaged over the last 30 episodes as was asked in the question:



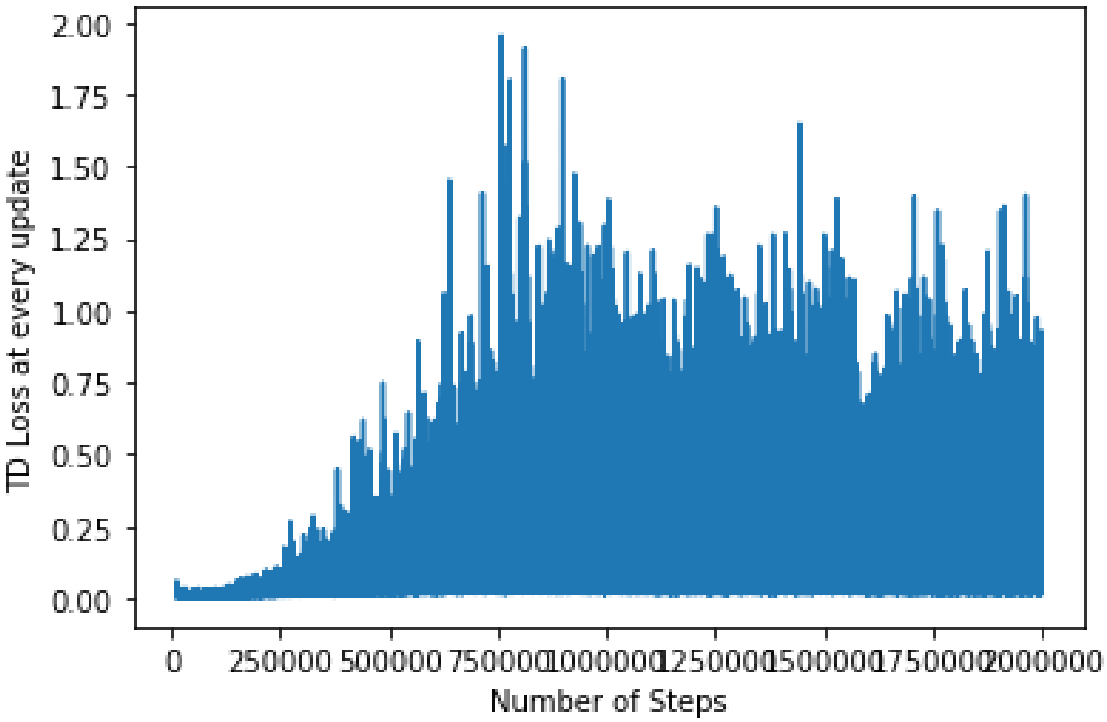
Answer 1.4.7.b

Below is the plot for return for average scores summed across as was asked in the question:



Answer 1.4.7.c

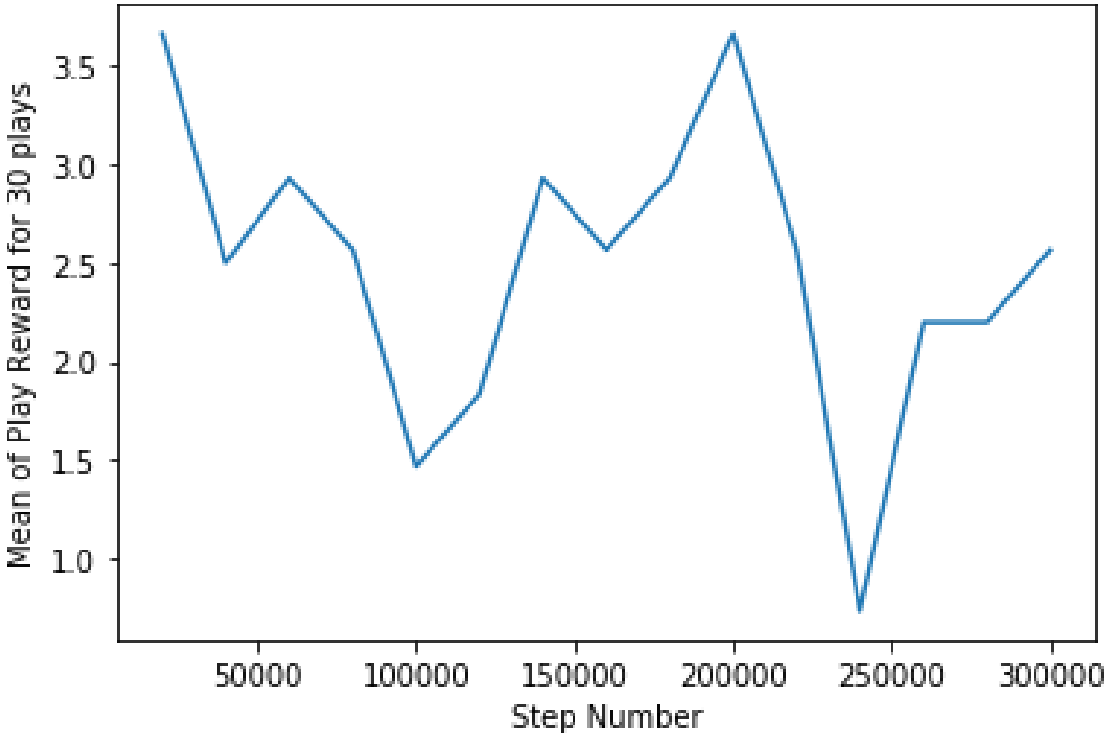
Below is the plot for return for Temporal Difference Error at each update step as was asked in the question:



Answer 1.4.7.d The mp4 file is attached in the zip folder with the name monitor_SpaceInvader.mp4. As can be seen from the video that agent even attempt and suceeds in killing the queen to get maximum reward.

Answer 1.4.8

Answer 1.4.9 Below is the plot for average scores evaluated at every 20,000 steps for running this code you will have to set self.bonus2=True and also have to provide the directory for the .npy file:


Mean of Play Reward for 30 plays
3.5
3.0
2.5
2.0
1.5
1.0
50000
100000
150000
200000
250000
300000
Step Number