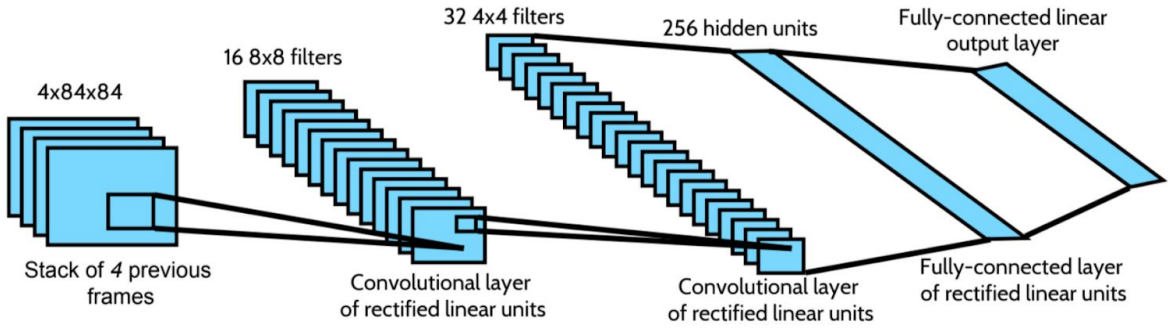
Advances in Robotics and Control

Assignment 2

DQN

Nivedita Rufus, 2019702002



Introduction

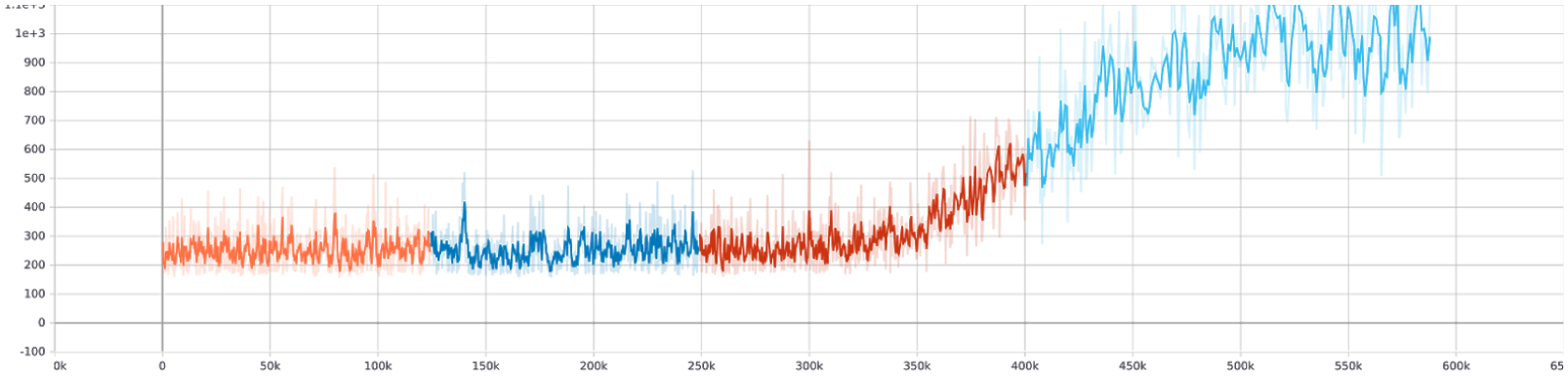
The task given to us is to observe the environment(Atari games) in which an agent interacts with it in a sequence of actions and rewards. At each time-step the agent selects an action a_t from the set of legal game actions, $A = \{1, \ldots, K\}$. The action is passed to the emulator and modifies its internal state and the game score. The network was trained for two of the Atari games namely Breakout(Q4) and Pong(Bonus) and the analysis of the mean reward with the past episodes were done. The performance of the network was also analysed after changing some hyper parameters for the Breakout game.

Q 4.1

A plot showing the performance of your DQN. x and y axis should indicate the number of time-steps and mean reward for the past 30 episodes respectively. You may also add a video/gif link showing your networks final performance.

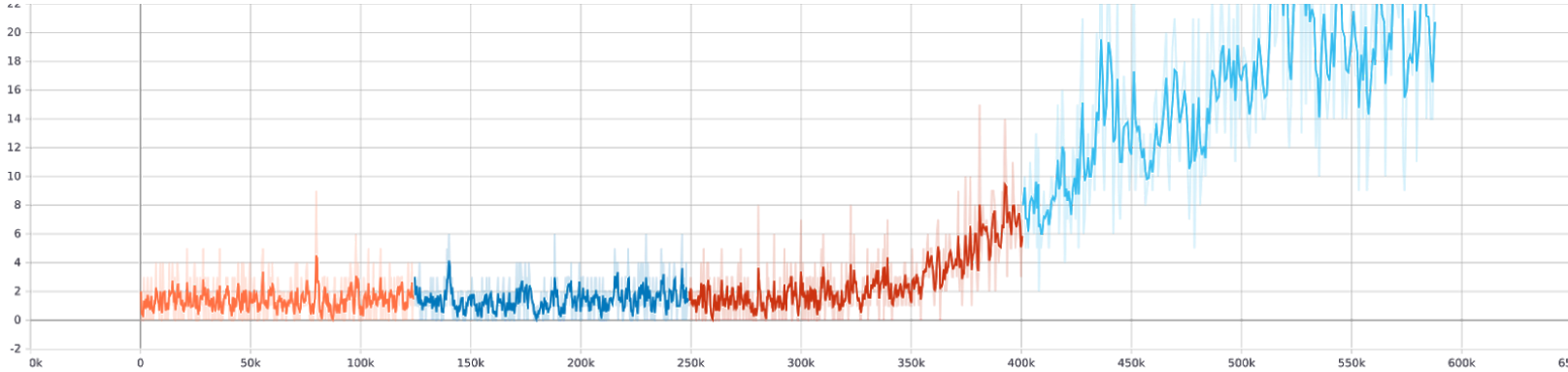
Length of episode

The figure below given below shows that steps taken in the game gradually become more and more, i.e the agent was able to play better without losing the lives in the game. In technical terms, this shows that the mean episode length increases as the number of steps increases. This also indicates that the performance of the network became better and better as the number of steps increases as we took more time to lose our lives which marks the end of an episode.



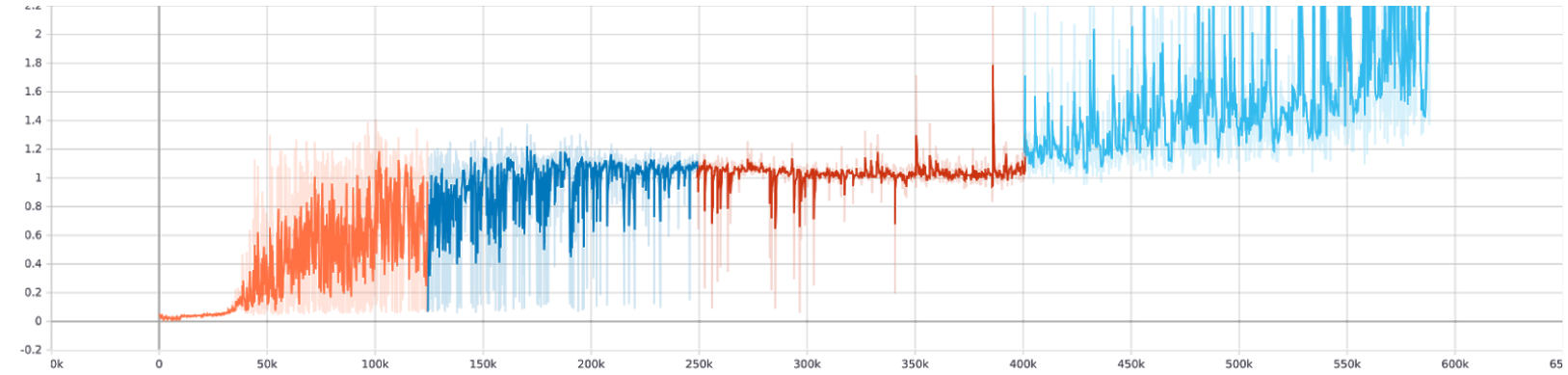
Reward of episode

This figure below shows the reward i.e. score we received after training for that many time steps. It can be noted that the reward always increases as the no. of steps increases. This only justifies the inference of the previous figure that the agent learns to play better as the no. of time steps increases. Thus we are able to play better without losing our lives.



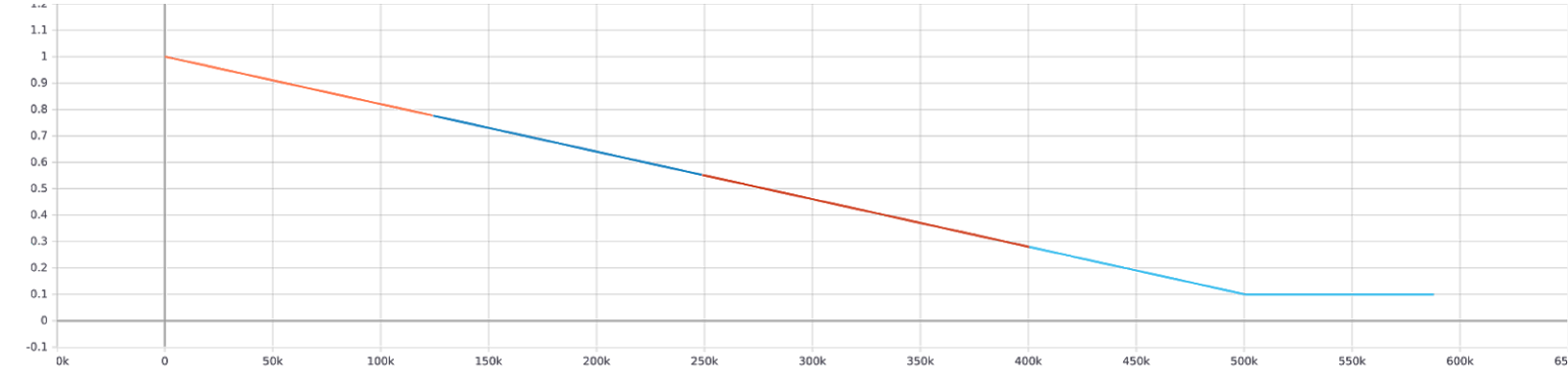
Max Q-values

The figure below shows max Q-values which gradually increases as the number of steps increases. This is due to the fact that we have more favourable state-action pairs after a lot of training steps. This also strongly justifies our inference of the previous figures that the performance gets better and better after enduring more no. of steps.



Value of Epsilon

This figure shows the change in epsilon which is set to decay from 1.0 with a tunable decay rate allowing more scope for exploration. The value is then changed to 0.1 after 500000 steps.



Link: [Link to the video of the game after training using the network.]

Q 4.2

After the network is trained, show 3-5 screenshots of the game, corresponding input to network, Q-values for each action corresponding to that input. Briefly explain your observations.

Atari Action set:

- *0 (noop)*
- *1 (fire)*
- *2 (left)*
- *3 (right)*

Screenshots of the game:

Action: 2 (going left to hit the ball)

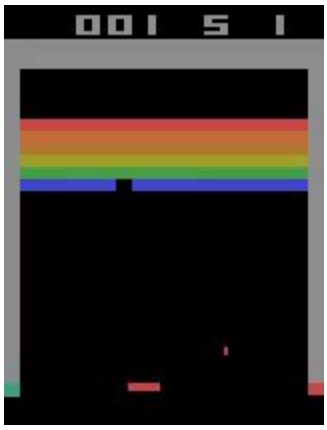
Frame: 19

Q-Value = 1.65

Action: 3 (going right to hit the ball)

Frame: 61

Q-Value = 1.39



Action: 3 (going right tracking the ball after hitting it)

Frame: 295

Q-Value = 1.77

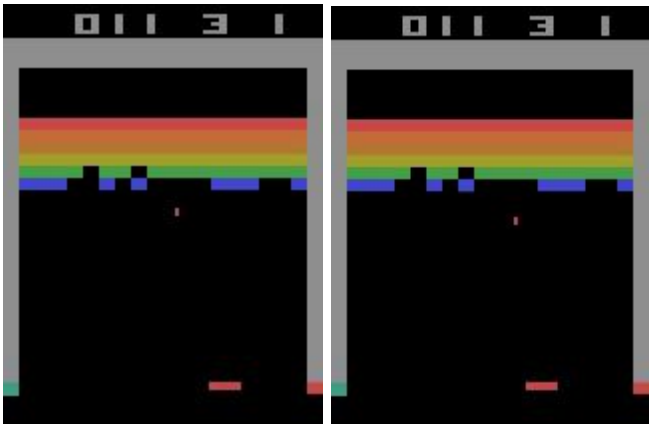
Action: 0 (No operation after hitting the ball)

Frame: 359

Q-Value = 1.87

Action: 3, 2 (oscillation i.e. indecisive behaviour, "*i am confused!!!!*")

Frame: 580, **Frame:** 581



Q 4.3

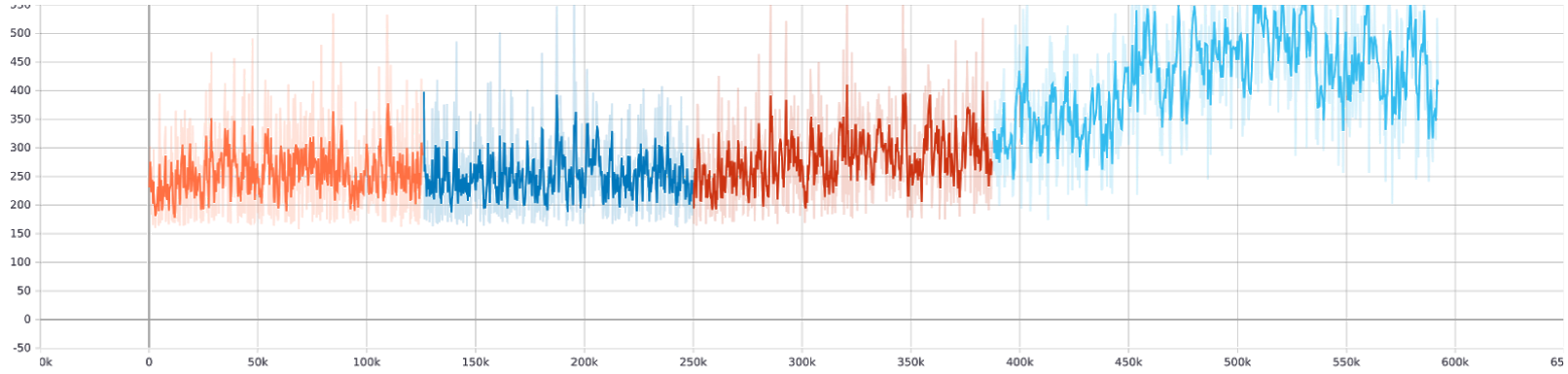
Choose one hyper-parameter (loss function, learning rate, input representation, exploration policy parameter, .. etc) that you expect to affect the performance of Q-network. Run at-least two more experiments by varying this hyper parameter and comment on the performance of the network with plots. Mention your reasoning for the choice of hyper parameter.

Hyperparameter chosen: Replay buffer size

Reason: The reason for choosing this particular hyper parameter is to observe the performance of the the network after the same no. of steps as the vanilla method. Ideally the observation should be that we would learn slower if we reduce the size of the replay buffer, i.e. the network is more *'forgetful'.* This is what can be very clearly observed in the graphs shown below.

Length of episode

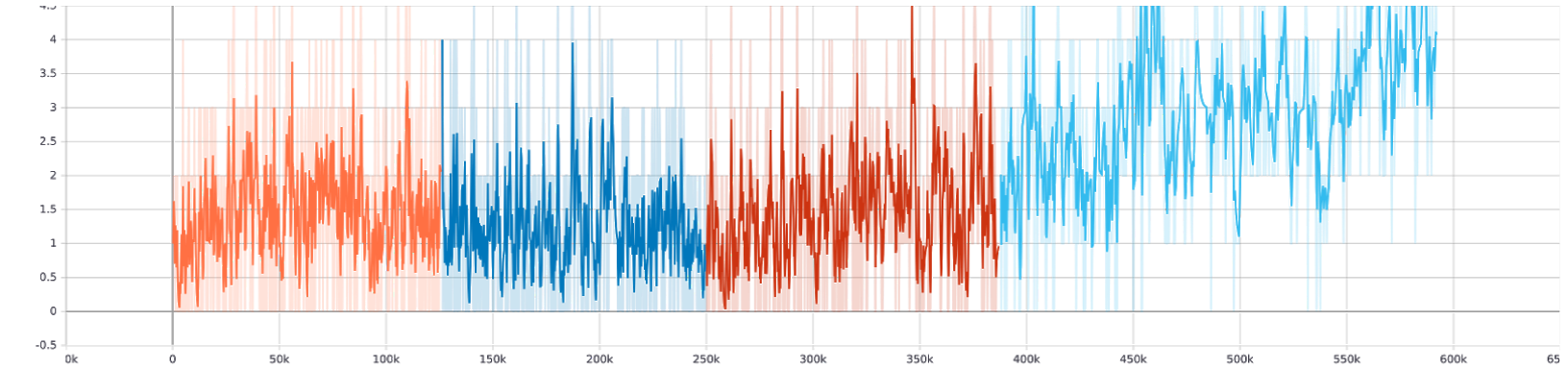
The figure below given below shows that steps taken in the game gradually become more and more, i.e the agent was able to play better without losing the lives in the game. In technical terms, this shows that the mean episode length increases as the number of steps increases. This also indicates that the performance of the network became better and better as the number of steps increases as we took more time to lose our lives which marks the end of an episode. The difference between this plot and the same plot of the vanilla network is that as the training approaches the the end the agent was able to play only 900 steps/moves in the vanilla network till it lost the game whereas here it could play only upto 500steps/moves.



Reward of episode

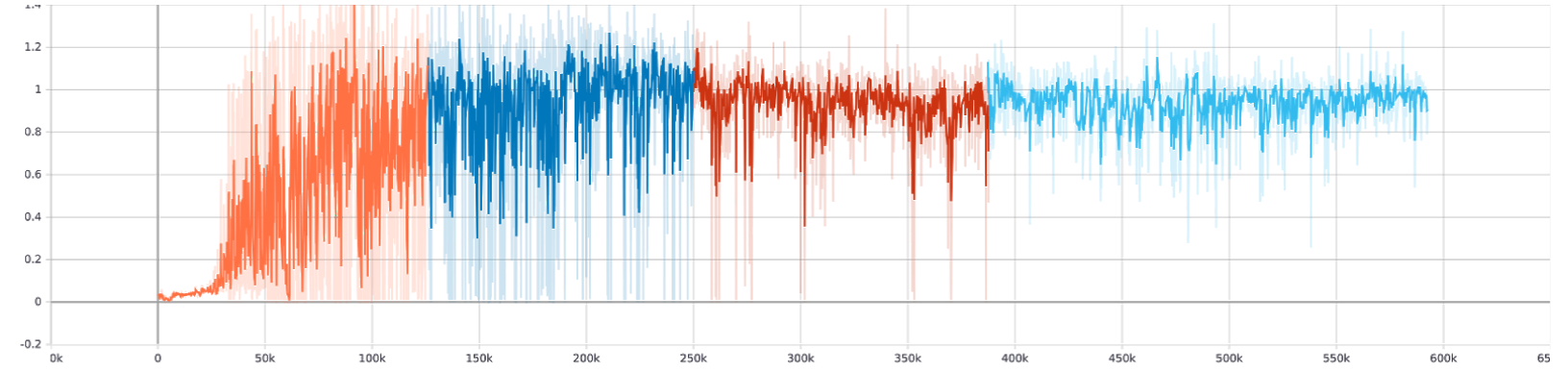
This figure below shows the reward i.e. score we received after training for that many time steps. It can be noted that the reward always increases as the no. of steps increases. This only

justifies the inference of the previous figure that the agent learns to play better as the no. of time steps increases. Thus we are able to play better without losing our lives. We were able to get upto 20 as a reward in the vanilla network but here we were able to reach only upto 5 for the same number of steps.



Max Q-values

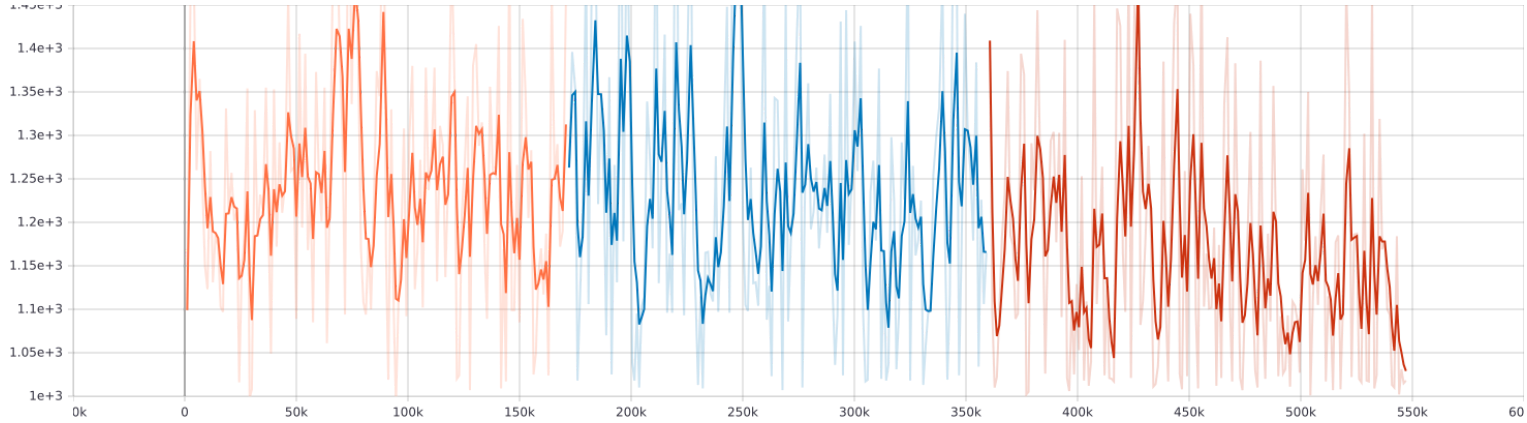
The figure below shows max Q-values which gradually increases as the number of steps increases. This is due to the fact that we have more favourable state-action pairs after a lot of training steps. This also strongly justifies our inference of the previous figures that the performance gets better and better after enduring more no. of steps. We also observe that the max Q-values are significantly much lower than that of the vanilla network.



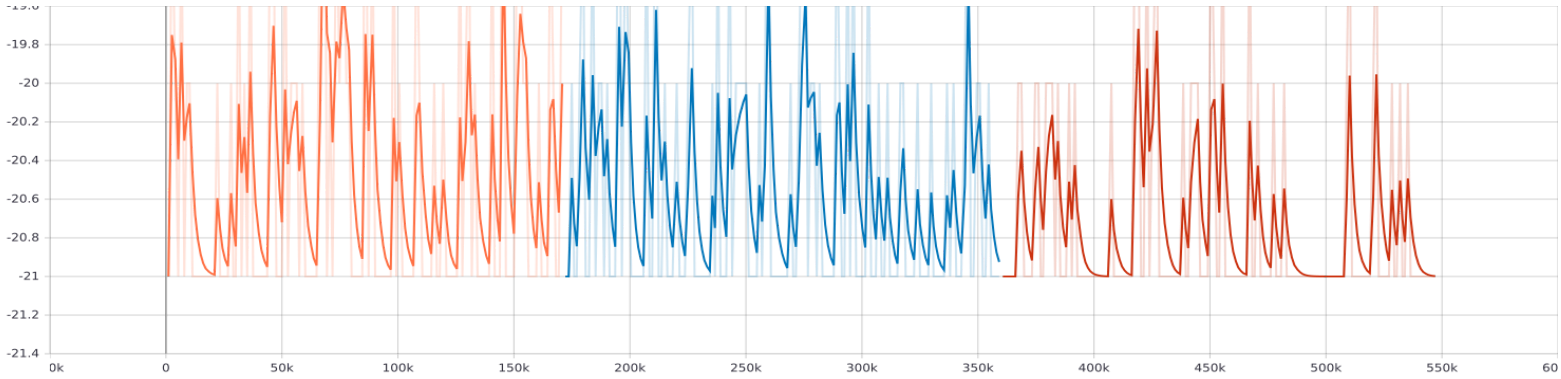
Bonus

Use the network trained on Breakout to evaluate on Pong. Report it's performance after 0, 100, 500 episodes of training. Show average score in the report. Show its performance plot after full training.

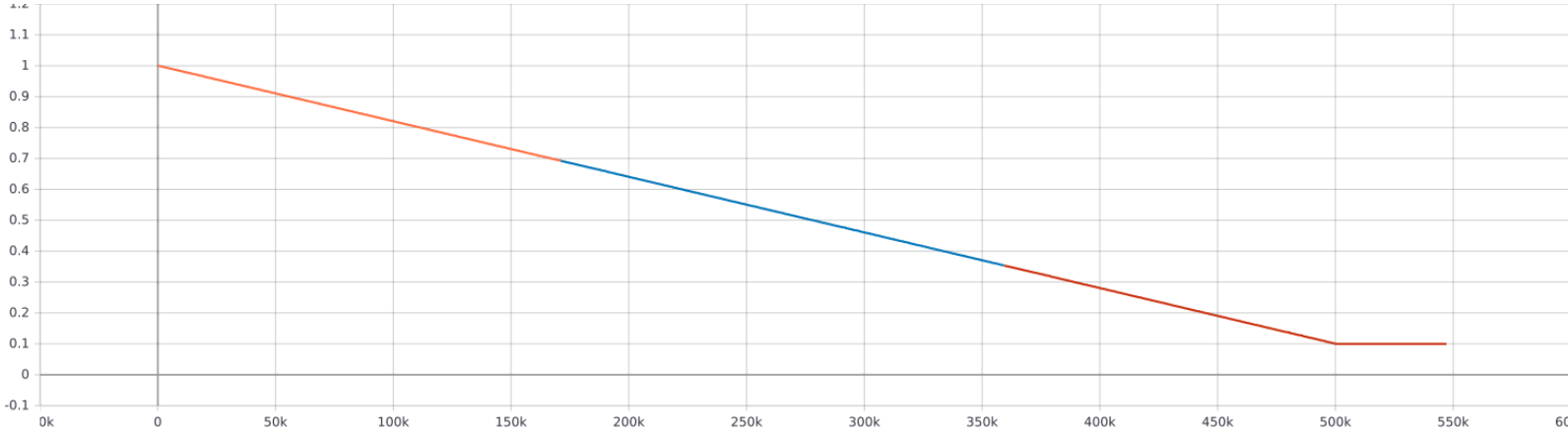
Length of episode



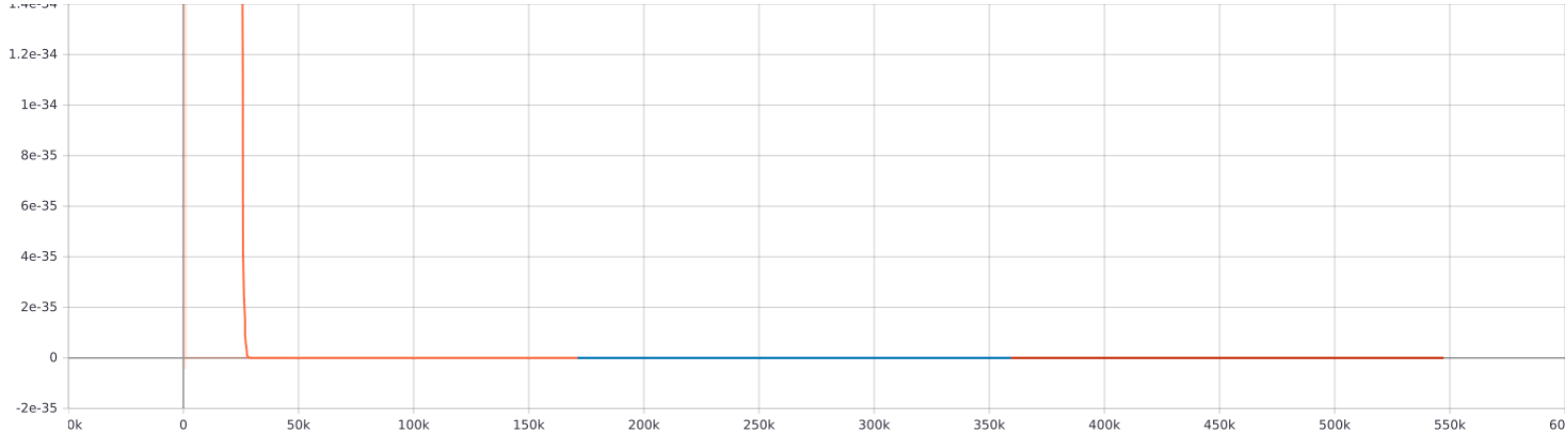
Reward of episode



Value of Epsilon



Max Q-value



All the plots above signify that the network we had did not train very well for Pong. This could be attributed to the fact that here, the agent must not only catch the ball but should hit it back in such a way that the defender misses it. Unlike Breakout it does not receive rewards for just touching the ball. It should definitely take more time-steps to train than it did for Breakout.

Screenshots of the game(PONG):