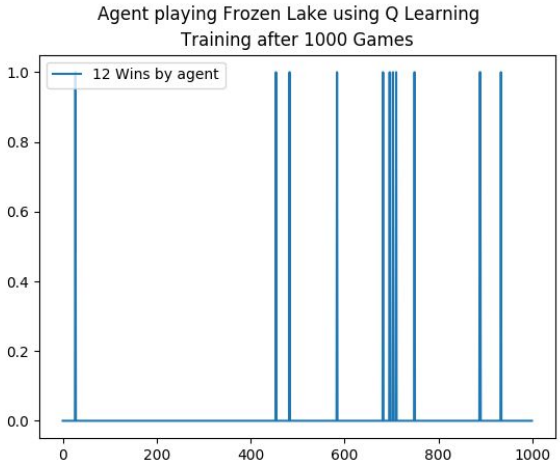
Implementing Q Learning Algorithm in Frozen Lake Environment

Q learning is implemented on the Frozen lake environment using only one Q table for choosing the action and updation. The agent is made to play the game for noOfGames times, and
Updation of Q learning is done using the following formula -
$Qnew(st,at) = (1-\alpha).Qold(st,at) + \alpha.(r + \gamma.\max Q(st+1,a))$
where α is learning rate and γ is discount factor

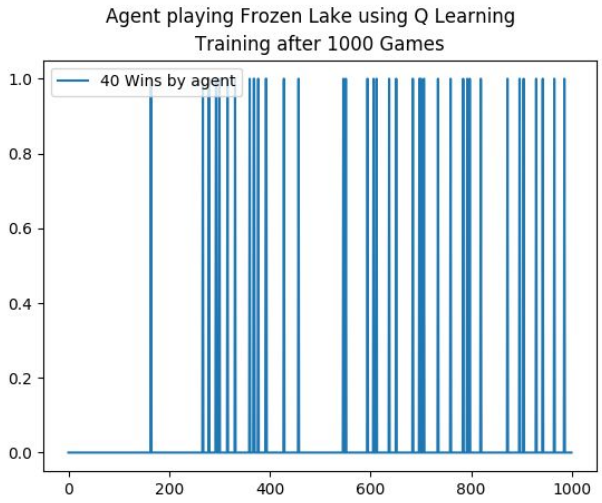
Variation in learning rate (which tells how much you accept the new value vs the old value). The bots performs a slightly better in cases of learning rates in the middle (0.6 etc) than the learning rates at the extremes (0.2,0.9)

Variation in rewards, we see the variation in the total no of wins by the agent.

- Normal given reward system

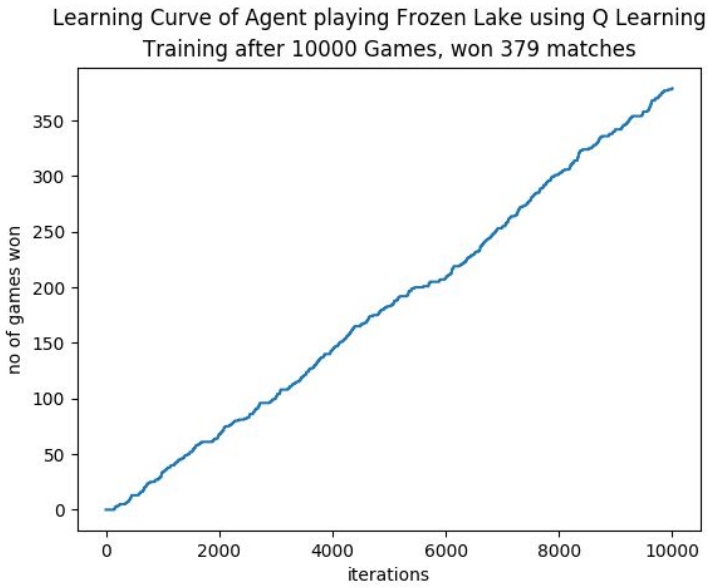


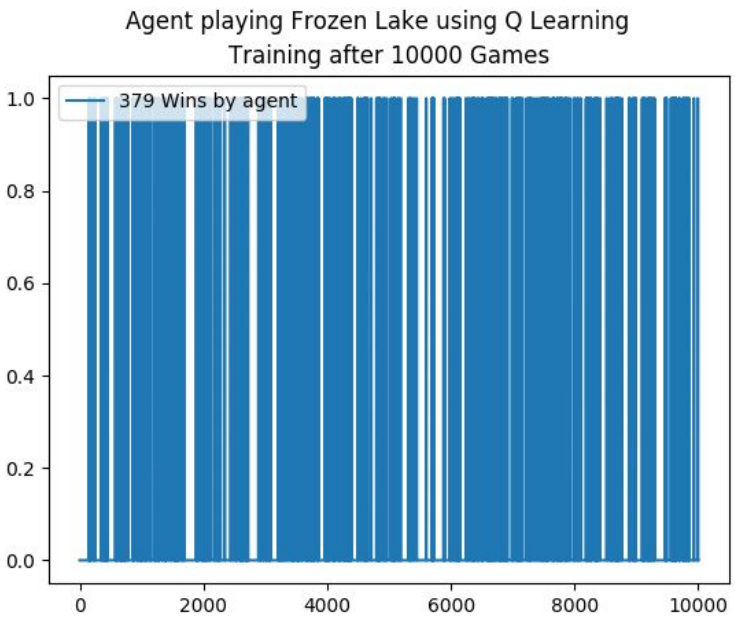
- Negative reward (-1) on loosing



Epsilon is also used to add exploration aspect to action selection using a random function. The 0.4 value of epsilon(close to middle ones) works significantly better than the extreme values.

For 10000 iterations, the learning curve and the Winning History graphs are -




Agent playing Frozen Lake using Q Learning
Training after 10000 Games
379 Wins by agent
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000