AI PROGRAMMING ASSIGNMENT-2 REPORT

Q1.) **Assumptions:**

- Termination condition taken is when all ants take the same path which will occur when the no. of edges in the graph is equal to the number of cities.
- Also, the distance matrix which contains the distances between cities is being initialized with random distances in the range 1 to 100.
- Here I have assumed that each city is connected to every other city in the graph.

Observations:

No. of cities	No. of ants	Alpha	Beta	Pheromone Evaporation Level	Iterations
20	40	1	1	0.5	131
20	40	1	2	0.5	122
20	40	2	1	0.5	19
20	40	1	1	0.1	598
20	40	1	1	0.9	46
20	60	1	1	0.5	146
20	20	1	1	0.5	55
20	40	1	4	0.5	31
20	40	4	1	0.5	17

- The parameters alpha, beta denotes how much importance we are giving to the pheromone level and to the length of path respectively.
- The parameters alpha, beta and the no. of ants taken play a significant role in the no. of iterations after which all ants start following the same path.
- On choosing alpha equal to beta the no. of iterations required are comparatively higher than if we chose beta greater than alpha.
- However, there is a kind of randomness when change the no. of ants, sometimes the no. of iterations are increasing and sometimes they get reduced.

Results:

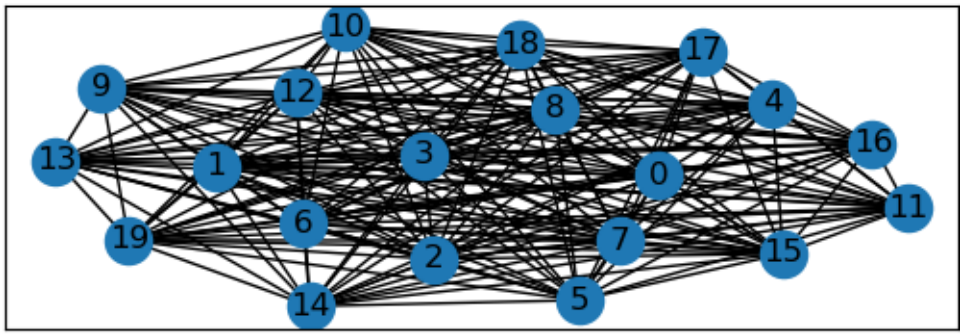
The graphs drawn using networkx for 2 cases are shown below.

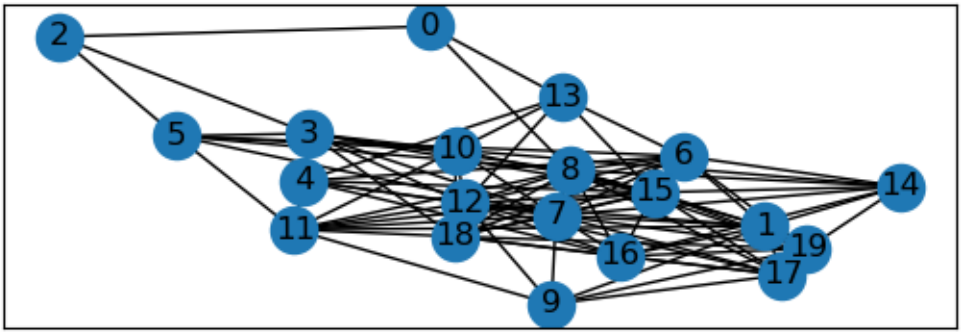
It can be seen as the iterations progress, more and more ants start following similar paths, as the no. of edges in the graph are decreasing.

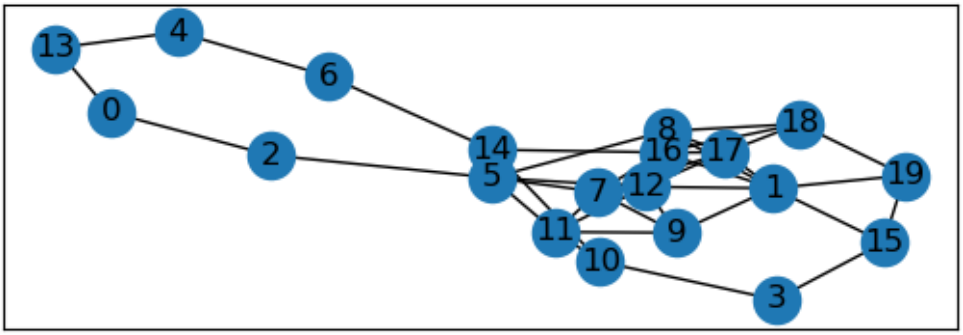
On the last iteration all ants are following the same path.

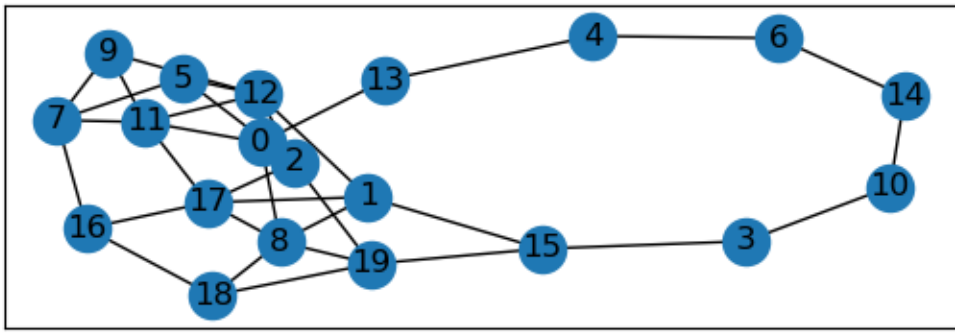
1.) Cities = 20, Ants = 40, Alpha = 6, Beta = 4, Evap rate = 0.9

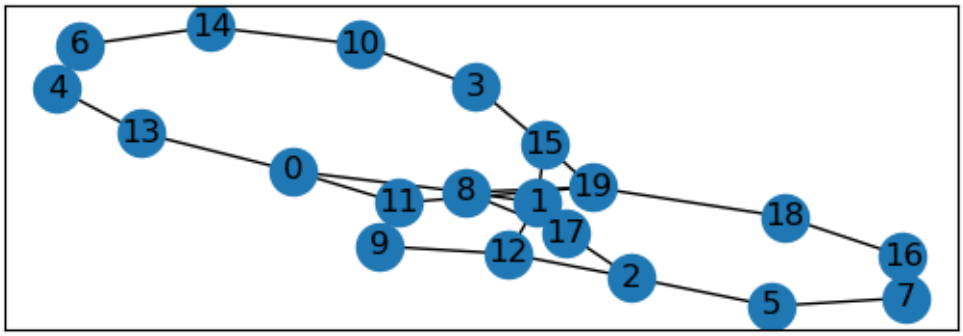
Iterations = 4

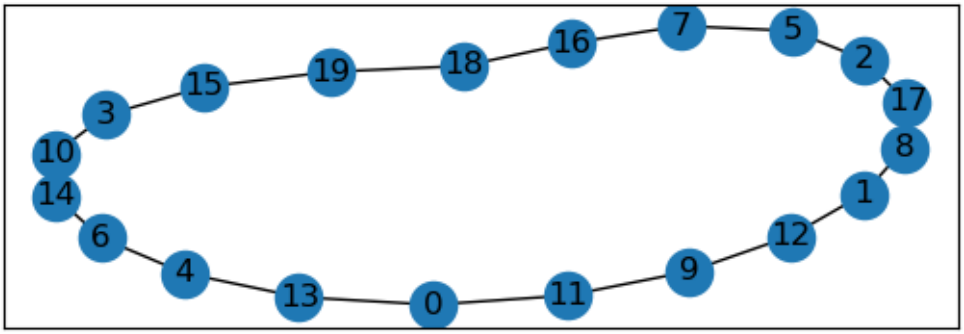




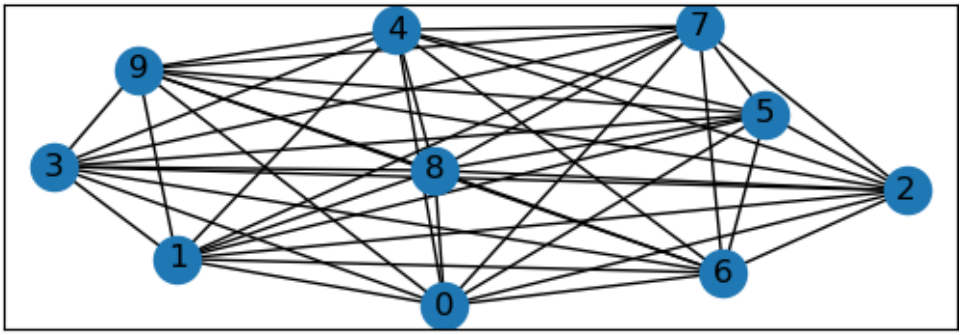


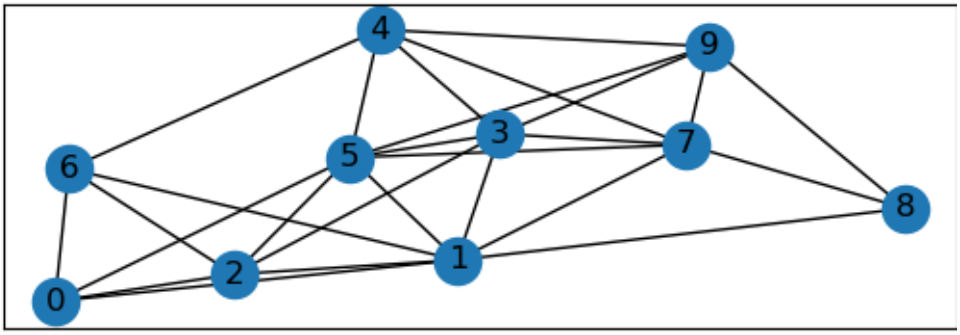

9
5
12
13
4
6
14
7
11
0
2
10
17
1
16
8
15
3
19
18

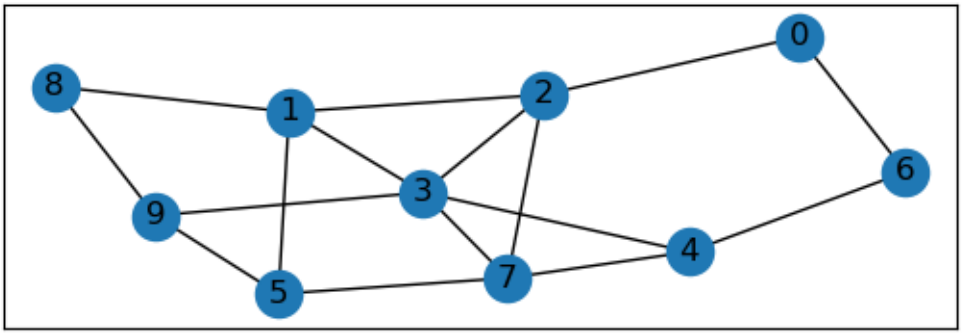

6
14
10
4
3
13
15
0
11
8
1
19
18
9
12
17
16
2
5
7

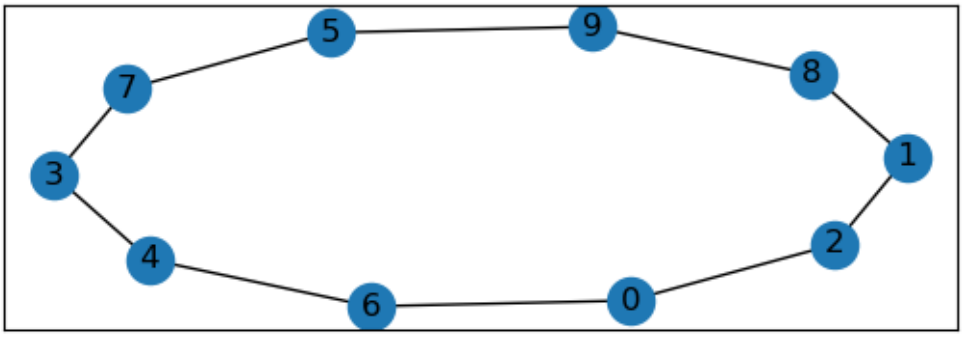

7
5
16
2
18
19
15
3
17
10
8
14
1
6
12
4
9
13
0
11

2.) Cities = 10, Ants = 20, Alpha = 6, Beta = 4, Evap rate = 0.8
Iterations = 4









Methodology used:

- Here at the beginning of each episode the probability matrix is populated.
- In the first episode the ants move randomly.
- From second episode onwards the ants move according to the probability distribution.
- To select the city using probability distribution the concept of roulette wheel has been used.
- At the end of episode, the pheromone matrix is updated.

Q2.) The learning curve corresponding to various combination of the hyperparameters is shown below along with each combination:

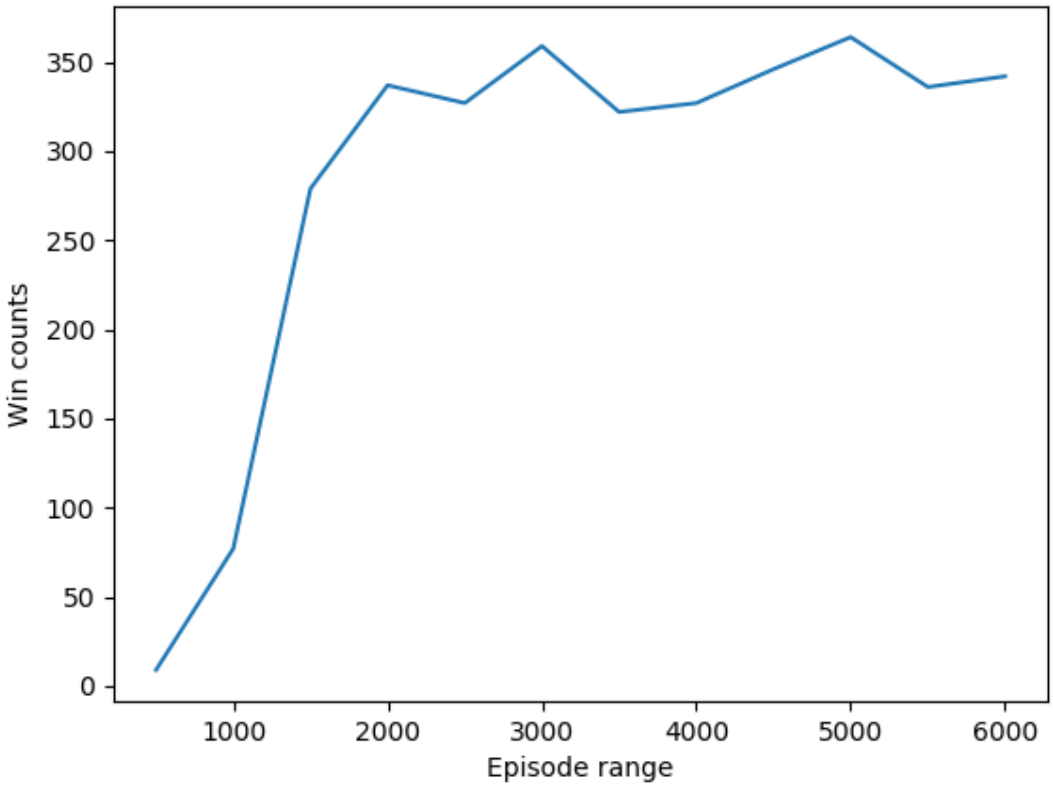
The hyperparameters varied over here are:

learning rate, discount factor, exploration rate

The learning curve here plots the counts of win per 500 episodes on the Y-axis and the episode range along the X- axis:

learning rate, discount factor, exploration rate

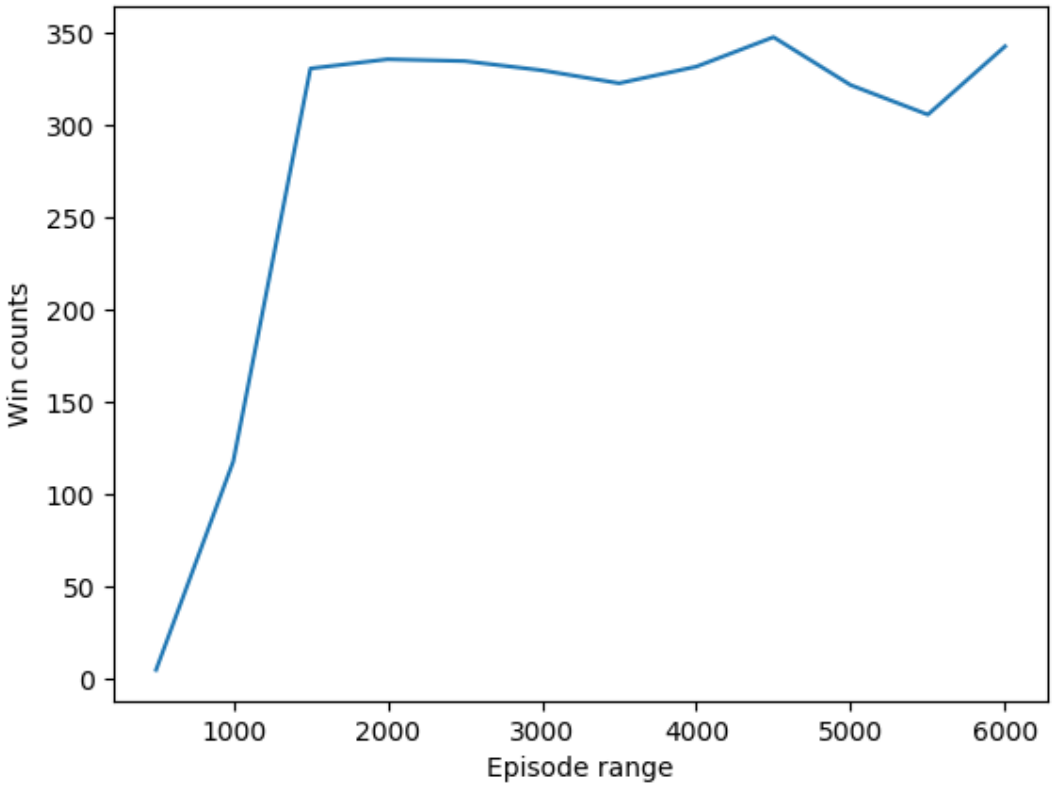
1.) learning rate = 0.1 , discount factor = 0.99 , exploration rate = 1



The win counts for each of the range of episodes is
[20, 126, 333, 334, 338, 326, 328, 350, 345, 348, 324, 318]

Above is the learning curve that we get for this choice of parameters.

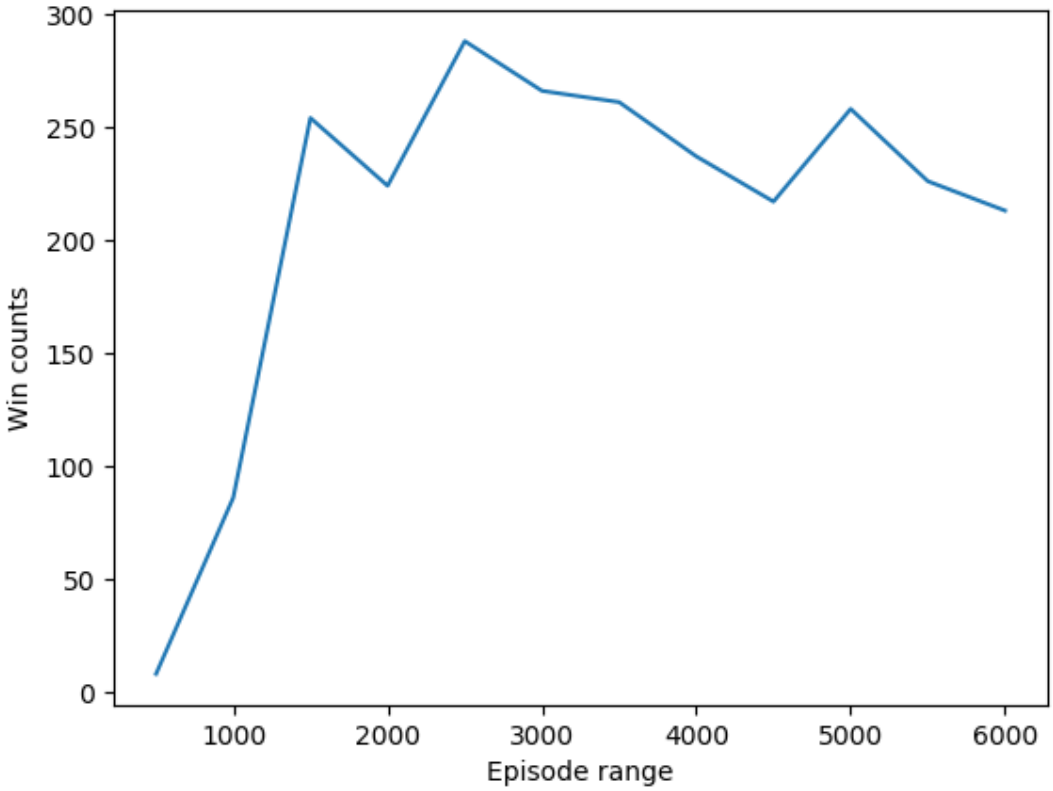
2.) learning rate = 0.2 , discount factor = 0.99 , exploration rate = 1



The win counts for each of the range of episodes is
[5, 118, 331, 336, 335, 330, 323, 332, 348, 322, 306, 343]

Not any significant change is seen on changing the learning rate from 0.1 to 0.2. A change we can see is that the no. of wins in the first 500 episodes has now decreased as compared to the previous case.

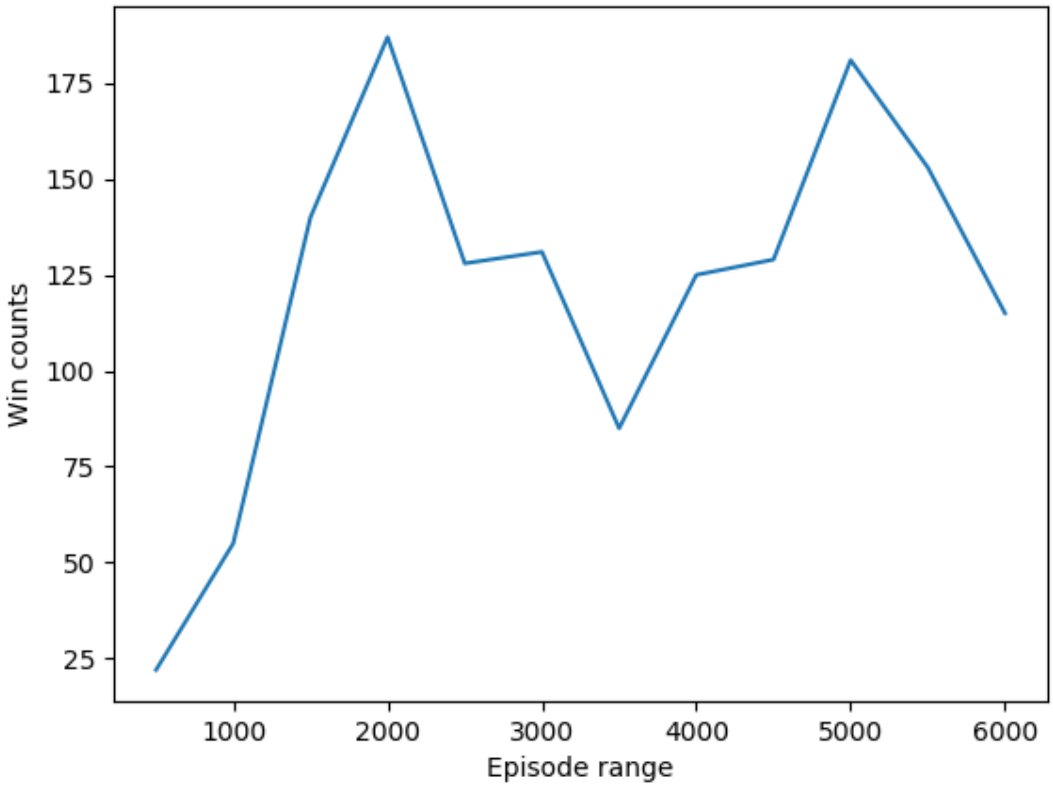
3.) learning rate = 0.1 , discount factor = 0.90 , exploration rate = 1



The win counts for each of the range of episodes are:
[8, 86, 254, 224, 288, 266, 261, 237, 217, 258, 226, 213]

So, we can see that on decreasing the discount factor and keeping all other parameters same, the no. of win counts of the agent per 500 episode slots decreases. We can see that for the episode range 5500-6000 the no. of wins here is 213 which is lower than what we get on keeping discount factor = 0.99

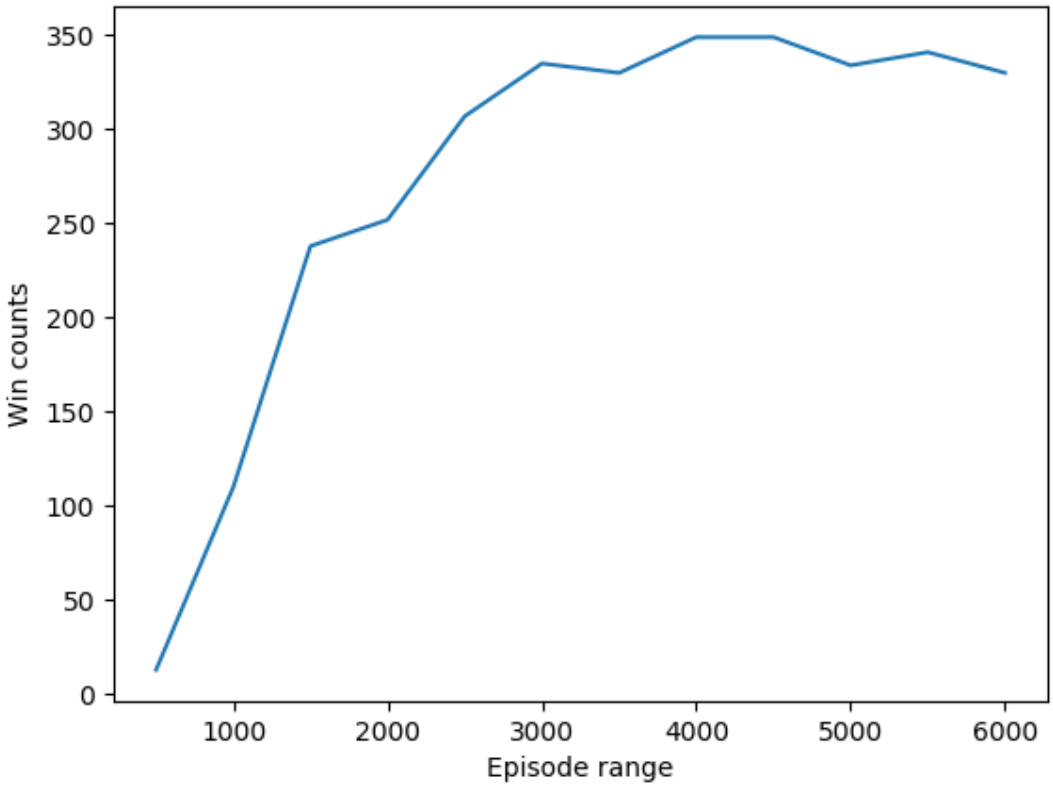
4.) learning rate = 0.1 , discount factor = 0.80 , exploration rate = 1



The win counts for each of the range of episodes are:
[22, 55, 140, 187, 128, 131, 85, 125, 129, 181, 153, 115]

On further decreasing the discount factor to 0.8 we can see that the win Counts per 500 episodes decrease drastically.

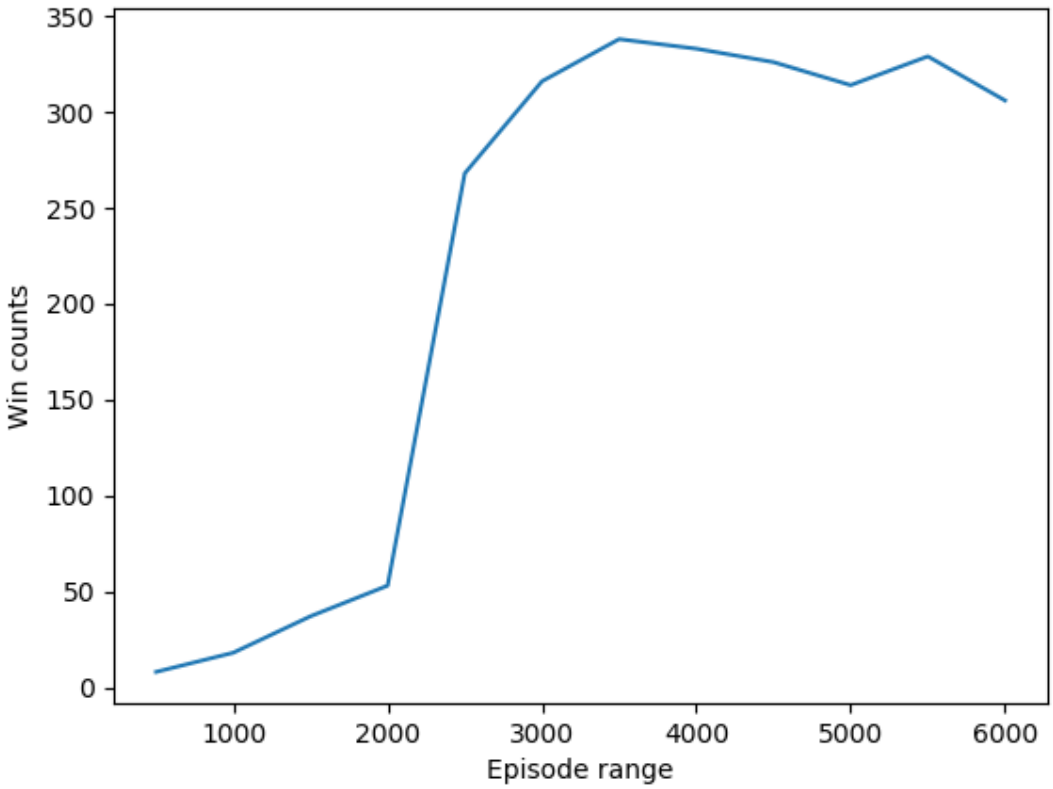
5.) learning rate = 0.1 , discount factor = 0.99 , exploration rate = 0.9



The win counts for each of the range of episodes are:
[13, 110, 238, 252, 307, 335, 330, 349, 349, 334, 341, 330]

On decreasing the exploration rate to 0.9 from 1 we can see that that the win counts per 500 episodes increase a bit slowly. That is the learning curve is less steep.

6.) learning rate = 0.1 , discount factor = 0.99 , exploration rate = 0.65



The win counts for each of the range of episodes are:
[8, 18, 37, 53, 268, 316, 338, 333, 326, 314, 329, 306]

On further decreasing the exploration rate 0.65the winning counts further decrease.

Inference:

We can therefore infer that keeping the exploration rate to be around 1 is a good choice. Also, a discount factor of near 0.99 seems good for this problem. Learning of around 0.1 or 0.2 is fine.

Q3.) Here for training the agent I have created 2 AI agents laying against each other. The agent which plays ‘O’ i.e my agent2 is the one which I am using testing part. Q-learning is used to train both the agents.

Also, as new states are generated as the episodes count progresses.

During testing the game is human i.e(‘X’) vs AI agent i.e (‘O’).

Observations:

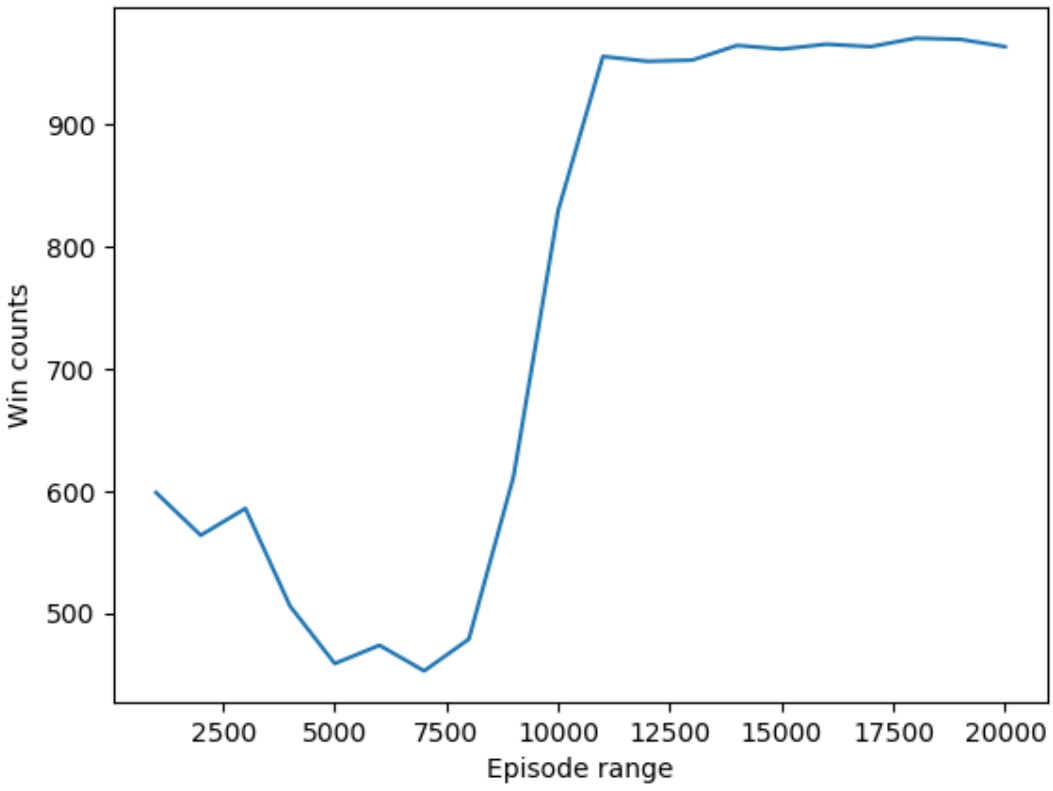
The below graph is plotted for win counts of agent 2 i.e. ‘O’ per 1000 episodes.

It can be seen that as episodes progress the win counts for the agent are increasing. Here for agent2 the values of parameters taken were:

Learning rate(alpha) = 0.1, discount factor(gamma) = 0.99

Epsilon = 1, epsilon decay = 0.0001, min epsilon = 0.01

No. of episodes for which trained = 20000



Inference

It can be seen that initially the agent2 i.e. ‘O’ does not win many games but as the episodes progress the winning counts of ‘O’ increase. Also as episodes progress more of exploitation is being done and thus the agent learns which move is the best one to take at a given state of the game.

Results:

In the testing phase the trained agent plays against human and tries to win the games.