# BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

# PILANI

Assignment

BITS F444/CS F407 – Artificial Intelligence

## Q-learning

By

| | |
|---|---|
| Bhoomi Sawant | 2017A7PS0001P |
| Harpinder Jot Singh | 2017A7PS0057P |

Submitted to

Dr. Navneet Goyal

## INTRODUCTION

Q-learning is a simple and easy to implement reinforcement learning algorithm. This assignment applies Q-learning to the game of dots and boxes. The objective is to train a Q-learning agent against random, simple and other Q-learning agents with different experience. The performance of Q-learning agents were analyzed for several different combinations of learning rate, training set size, exploration coefficient and discount factor. Detailed discussion, result tables and labelled plots are included in this report.

## OBJECTIVE

The objective of this project is to implement a Q-learning based solution for a 2 player game. We have chosen the game of dots and boxes for this assignment. Using the Q-learning algorithm, we have to train 8 agents by making them play several thousands games against random agents, simple agents and even with other agents trained from Q-learning. The scenarios for training 8 agents are summarized in the table below:

| First player | Second player | Name of agent produced |
|---|---|---|
| Q-learning agent | Random agent | Q1 |
| Q-learning agent | Q-learning agent | Q2 |
| Q-learning agent | Simple agent | Q3 |
| Q1 | Q-learning agent | Q4 |
| Q2 | Random agent | Q5 |
| Q4 | Q5 | Q6 |
| Q3 | Q1 | Q7 |
| Q1 | Q2 | Q8 |

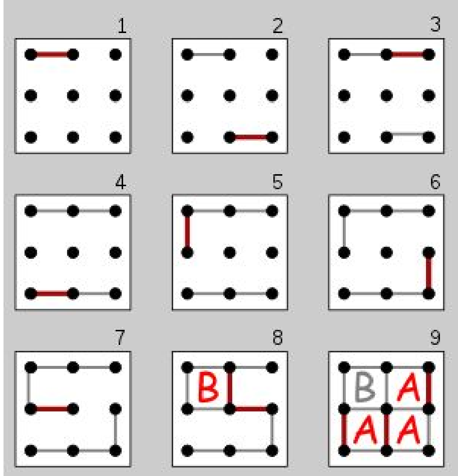
We have to analyze the performance of the agents by varying the parameters, learning rate α, discount factor γ, exploration quotient ε and draw conclusions about Q-learning from the experiments by comparing different agents.

## DOTS AND BOXES

Dots and boxes is a pencil-and-paper, usually played by two players, also known as la pipopipette. Typically there are 9 squares, that is, 16 dots, but the game board can be formed using any number of dots and boxes.

In the beginning of the game, there is an empty grid of dots. Two players play their turns one after another by drawing a line vertical or horizontal between two adjacent points, provided the line is not already drawn. If a player completes all four lines of a square and finishes a small box, one point is recorded for that player. After this step, the same player has to draw another line. The game ends when all the boxes in the grid are completed and the winner is the player with the most points.

An example of 9 dots and 4 boxes is shown in the figure. Two players A and B played the game and A won.



## REINFORCEMENT LEARNING

Reinforcement learning is learning what actions to take in a situation, or mapping a state or situation to actions, in a way so as to maximize a numerical total reward. The agent should be able to understand the surroundings and take actions to modify the current state.

The reward system is drawn from the findings in Neuroscience about how the human brain makes decisions, by devoting to the dopamine system.

RL is different from supervised learning in a sense that in RL, we just tell the agent that if you take a particular action in some state, how much reward you get, but we do not inform whether it is a best action or not.

Basically, it deals with learning via interaction and feedback, or in other words learning to solve a task by trial and error, or in other-other words acting in an environment and receiving rewards for it.

## Q-LEARNING

Q-learning is a simple and easy to implement reinforcement algorithm. State means the current game configuration. Action refers to the transition from one state to another. An action generates a reward (or penalty) from the environment to estimate the goodness of an action.

There are two main parameters involved: learning rate alpha and discount factor gamma. The learning rate describes how much we are changing the Q-value with each step. The discount factor is used to adjust the importance of future feedback.

The value for each state-action pair is stored in a matrix, thus this algorithm requires large memory. At every state, the best action is chosen (other actions also can be chosen depending on degree of exploration) based on the states it can go from the current state. The feedback helps to update the Q-value of the current state according to the following equation:

$$Q(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old Q}-\text{value}} + \underbrace{\alpha}_{\text{learning rate}} \times \Big[ \overbrace{\underbrace{r_{t+1}}_{\text{feedback}} + \underbrace{\gamma}_{\text{discount factor}} \underbrace{\max_a Q(s_{t+1},a)}_{\text{max future Q}-\text{value}}}^{\text{expected discounted feedback}} - \overbrace{Q(s_t,a_t)}^{\text{old Q}-\text{value}} \Big]$$

**Q-learning Agent**: The purpose of Q-learning agent is to use Q-learning algorithm to choose an action for a particular state.

**Random Agent**: As the name suggests, this agent is unpredictable. Given a state, it will choose random action. One advantage of this agent is that a lot of training data can be generated using this agent, which is used to train the Q-learning agent (as in round1 and round5).

**Simple Agent**: This agent always chooses the minimum numbered line which is not already drawn. The same move is chosen all the time, irrespective of the opponent. It gives its opponent easy and predictable moves.

### State Space size

The configuration of a game at any point defines its state. For the dots and boxes game, we have points in grid and lines connecting them, each line is either drawn or not. For a 3 x 3 grid, i.e. 16 points, we have 24 lines possible, each drawn or not leads us to 2^24 possible configurations.

Drawing a line is termed as an action taken by the agent. On average, we could draw 12 lines in a state, so that leads us to 12x2^24 combinations possible for state and action pairs.

# RESULTS

## Q1: Q-learning agent trained against a Random agent

The share of wins and Q learning score for Q-learning agent when trained with a random agent on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.8 | 0.818680 | 637360 |
| 0.4 | 0.8 | 0.816370 | 632740 |

| | | | |
|---|---|---|---|
| 0.6 | 0.8 | 0.814610 | 629220 |
| 0.8 | 0.8 | 0.813490 | 626980 |

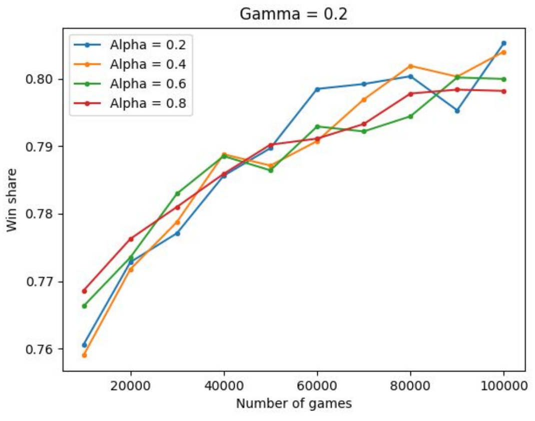
We observed that the **best results were obtained for learning rate 0.2 and discount factor 0.8**

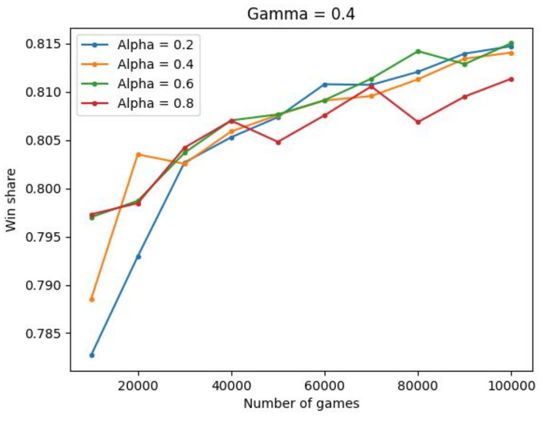
The share of wins and Q learning score for Q-learning agent when trained with a random agent on 200000 games for different learning rate and discount factor is summarized in the table below:

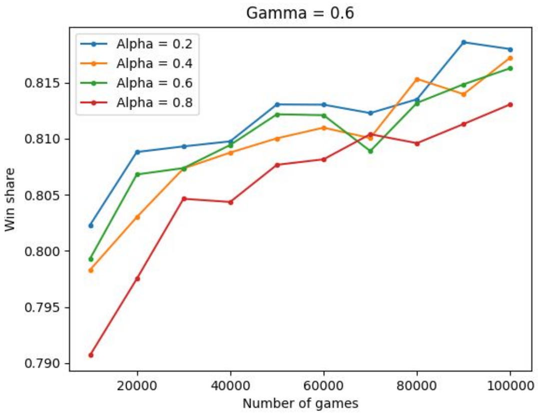
| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.8 | 0.818415 | 1273660 |
| 0.4 | 0.8 | 0.817955 | 1271820 |
| 0.6 | 0.8 | 0.817265 | 1269060 |
| 0.8 | 0.8 | 0.816065 | 1264260 |

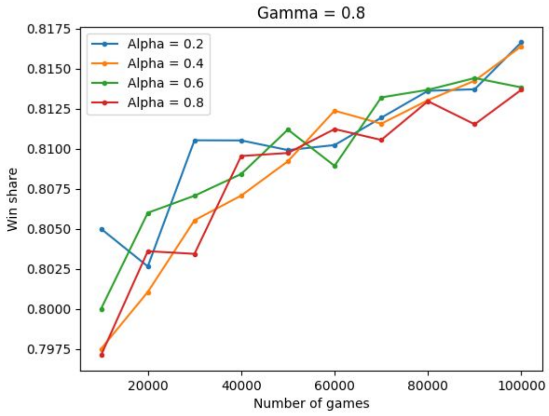
We again observe that **learning rate 0.2 and discount factor 0.8 gives the best results.** Hence, these parameters were chosen for Q1.

The figures below are the plots of win share with respect the number of games played for multiple combinations of the parameters alpha and gamma. It can be seen in all four figures that as we increase the number of games played, the share of wins of Q-learning agent increases. In each figure, gamma is kept constant and different colors are used for each value of alpha.

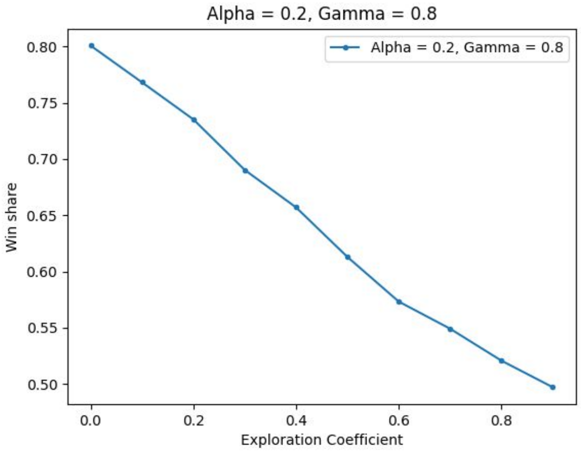








We tried to observe the change in win share by varying the exploration coefficient.The results are summarized in the graph below:

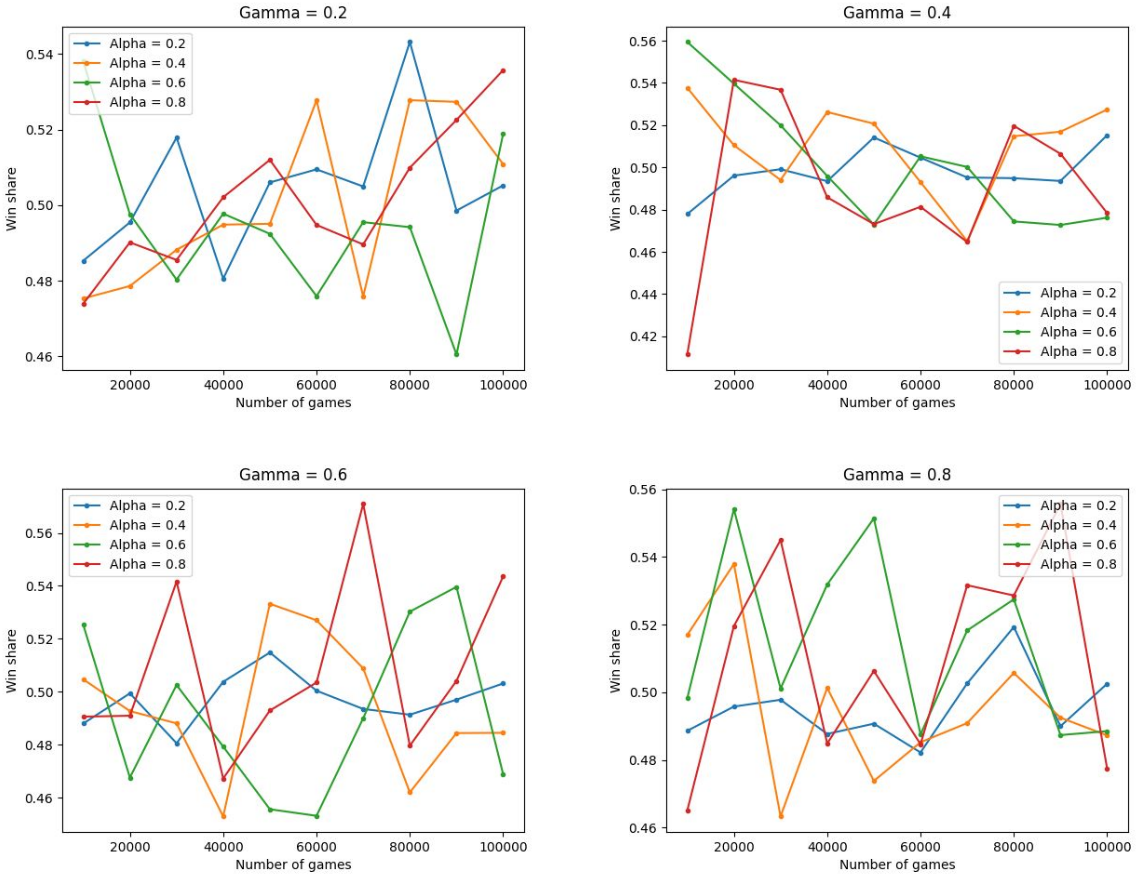


## Q2: Q-learning agent trained against another Q-learning agent

The share of wins for both Q-learning agents when trained against one another on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins for agent 1 | Share of wins for agent 2 |
|---|---|---|---|
| 0.2 | 0.8 | 0.499990 | 0.500010 |
| 0.4 | 0.8 | 0.527920 | 0.472080 |
| 0.5 | 0.8 | 0.507900 | 0.492100 |
| 0.8 | 0.8 | 0.473350 | 0.526650 |

**We observed that for each pair of learning rate and discount factor, share of wins were almost the same, close to 0.5**.



The above plots show that when a fresh Q-learning agent is trained against another fresh Q-learning agent, both achieve a share of wins close to 0.5, no clear trend is seen with the increase in the number of games played, as expected. Both models try to perform better against each other.

## Q3: Q-learning agent trained against a Simple agent

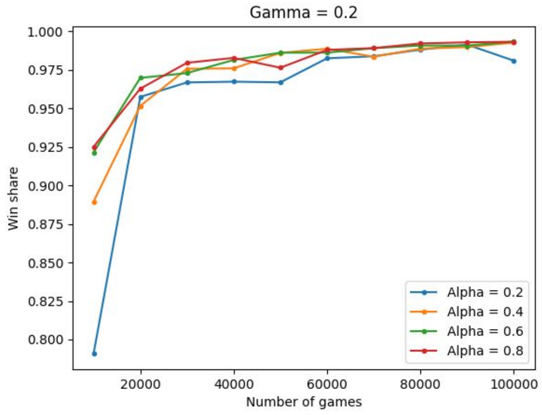
While training a Q-learning agent against a simple agent, the Q-learning got a simple and predictive competition, hence as expected, this round took least time for training. The share of wins and Q-learning score for Q-learning agent when trained with a simple agent on 100000 games for different learning rate and discount factor is summarized in the table below:

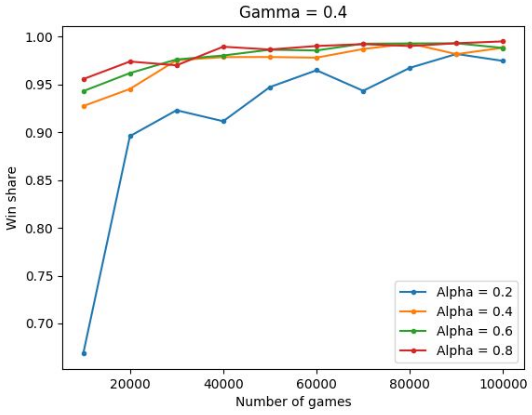
| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.988930 | 977860 |
| 0.2 | 0.4 | 0.976340 | 952680 |
| 0.2 | 0.6 | 0.899380 | 798760 |
| 0.2 | 0.8 | 0.889190 | 778380 |

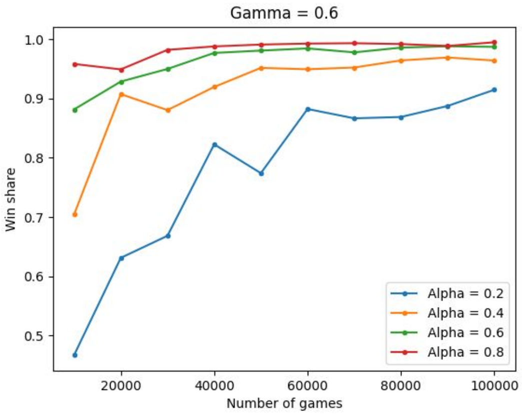
**We observed that the best results were obtained for learning rate 0.2 and discount factor 0.2**

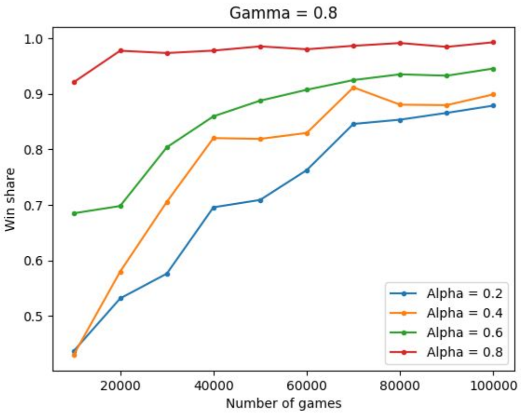
The share of wins and Q learning score for Q-learning agent when trained with a simple agent on 200000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.994905 | 1979620 |
| 0.4 | 0.2 | 0.995330 | 1981320 |
| 0.6 | 0.2 | 0.996100 | 1984400 |
| 0.8 | 0.2 | 0.995890 | 1983560 |









**We observe that, for all the values of gamma, alpha 0.8 is the best**. For gamma 0.2, all the values of alpha give win of shares more than 0.9. **Thus we concluded that for Q3, alpha 0.8 and gamma 0.2 is the best.**

## Q4: Q1 trained against a new Q-learning agent

The share of wins and Q learning score for Q1 trained agent when trained with a Q-learning agent on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.564330 | 128660 |
| 0.2 | 0.4 | 0.576260 | 152520 |
| 0.2 | 0.6 | 0.512080 | 24160 |
| 0.2 | 0.8 | 0.546780 | 93560 |

The share of wins and Q learning score for Q1 trained agent when trained with a Q-learning agent on 200000 games for different learning rate and discount factor is summarized in the table below and **we observe that learning rate 0.2 and discount factor 0.4 gives the best results in this case**.

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.4 | 0.587955 | 351820 |
| 0.4 | 0.4 | 0.529645 | 118580 |

|  |  |  |  |
|---|---|---|---|
| 0.6 | 0.4 | 0.561040 | 244160 |

## Q5: Q2 trained against a random agent

The share of wins and Q learning score for Q2 trained agent when trained with a random agent on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.843540 | 687080 |
| 0.2 | 0.4 | 0.845330 | 690660 |
| 0.2 | 0.6 | 0.843720 | 687440 |
| 0.2 | 0.8 | 0.841910 | 683820 |

The share of wins and Q learning score for Q2 trained agent when trained with a random agent on 200000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.4 | 0.835020 | 1340080 |
| 0.4 | 0.4 | 0.834005 | 1336020 |
| 0.5 | 0.5 | 0.833040 | 1332160 |

**We observe that learning rate 0.2 and discount factor 0.4 gives the best results in this case.**

## Q6: Q4 trained against Q5

The share of wins and Q learning score for Q4 trained agent when trained with Q5 on 100000 games for different learning rate and discount factor is summarized in the table below.

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.583570 | 167140 |
| 0.2 | 0.4 | 0.605420 | 210840 |
| 0.2 | 0.6 | 0.651650 | 303300 |
| 0.2 | 0.8 | 0.680500 | 361000 |

The share of wins and Q learning score for Q4 trained agent when trained with Q5 on 200000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.8 | 0.770705 | 1082820 |
| 0.4 | 0.8 | 0.998275 | 1993100 |
| 0.6 | 0.8 | 0.999780 | 1999120 |
| 0.8 | 0.8 | 0.999870 | 1999480 |

**We observe from the above table that the best win share is obtained by using alpha = 0.8 and gamma = 0.8.**

## Q7: Q3 trained against Q1

The share of wins and Q learning score for Q3 trained agent when trained with Q1 on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.986960 | 973920 |
| 0.2 | 0.4 | 0.982960 | 965920 |
| 0.2 | 0.6 | 0.931510 | 863020 |
| 0.2 | 0.8 | 0.818330 | 636660 |

The share of wins and Q learning score for Q3 trained agent when trained with Q1 on 200000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.8 | 0.903225 | 1612900 |
| 0.4 | 0.8 | 0.993040 | 1972160 |
| 0.6 | 0.8 | 0.997195 | 1988780 |
| 0.8 | 0.8 | 0.997325 | 1989300 |

**We observe from the above table that the best win share is obtained by using alpha = 0.8 and gamma = 0.8**

## Q8: Q1 trained against Q2

The share of wins and Q learning score for Q1 trained agent when trained with Q2 on 100000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.2 | 0.573020 | 146040 |
| 0.2 | 0.4 | 0.544310 | 88620 |
| 0.2 | 0.6 | 0.533440 | 66880 |
| 0.2 | 0.8 | 0.565840 | 131680 |

The share of wins and Q learning score for Q1 trained agent when trained with Q2 on 200000 games for different learning rate and discount factor is summarized in the table below:

| Learning Rate α | Discount factor | Share of wins | Q Learning score |
|---|---|---|---|
| 0.2 | 0.8 | 0.574040 | 296160 |

| | | | |
|---|---|---|---|
| 0.4 | 0.8 | 0.997360 | 1989440 |
| 0.6 | 0.8 | 0.999905 | 1999620 |
| 0.8 | 0.8 | 0.999910 | 1999640 |

**We observe from the above table that the best win share is obtained by using alpha = 0.8 and gamma = 0.8**

## CONCLUSION

This assignment talks about the Q-learning implementation for a popular 2-player board game dots and boxes. The Q-learning agent was trained against random, simple and other Q-learning agents. The performance was analyzed by varying learning rate, discount factor and exploration quotient. A total of 8 scenarios were considered and the results are reported in the form of tables and graphs. We observed that the more the number of games played by the agent, the better it could perform. We also noticed that a simple agent was the easiest competitor for a Q-learning agent. The model with those parameter values were chosen for further analysis which gave the most share of wins.

## REFERENCES

1. Q-Learning for a Simple Board Game, OSKAR ARVIDSSON and LINUS WALLGREN. Paper link

2. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, Pearson, 4e,Prentice Hall

3. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.

4. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011