**Homework # 4 Report**

Gautam Balachandran

UID : 116079495

In this project I have implemented a program that can learn to play the Dots-And-Boxes game. The learning can take place in one of two ways :

1. **Through the use of Q-Tables** :- A Q-table is a table that stores the Q-value for every state and action in the game. The Q-value is computed and updated to the table every time a box is completed, or the game is won. A reward of +1 is assigned every time a box is completed and +5 every time a game is won.
2. **Through a Function Approximation** :- This is a function or model that closely tries to match a target function or model based on the task assigned to it. In this project, I implemented a Radial Basis Function (RBF) network using stochastic gradient descent using the Scikit-Learn machine learning package from python.

The training part has been done for 100, 1000 and 10000 iterations, and the testing was done for a standard 100. The following win percentages were observed :

1. **2x2 game** :

* 100 Iterations : 37%-win rate for Q-Tables and 42%-win rate for function approximation.
* 1000 Iterations : 45%-win rate for Q-Table and 63%-win rate for function approximation.
* 10000 Iterations : 67%-win rate for Q-Table and 82%-win rate for function approximation.

1. **3x3 game** :

* 100 Iterations : 44%-win rate for Q-Tables and 53%-win rate for function approximation.
* 1000 Iterations : 54%-win rate for Q-Table and 65%-win rate for function approximation.
* 10000 Iterations : 73%-win rate for Q-Table and 88%-win rate for function approximation.

As can be observed from the data above, with increase in the number of iterations in training, the success percentage increased correspondingly for both Q-Tables and Function approximation. But, there is a noticeably high increase in the performance of the code that using function approximation – an upto 40% increase in performance. Such a large increase in performance would mean that function approximation is a more efficient and effective means of machine learning than using Q-Tables. I also ran into several out of memory issues while using Q-Tables, so the function approximation method is more memory efficient as well.