

DOTS AND BOXES #HW4
ENPM808F
Abhishek Kathpal
114852373

TASK:

The task is to implement Dots and Boxes game using Reinforcement learning. In this game, players take turns drawing lines to complete boxes. A player is allowed to take another turn after completing a box. The game concludes when no more lines can be drawn and all boxes are claimed.

The task is broken down into following step:-

1. Implementing the game with a 2x2 grid using Q-learning.
2. Implementing the same technique with 3x3 grid.
3. Replacing the Q-table with neural network to find the probability of actions.

A reward of +1 is assigned for completion of a box, +5 for a win, and 0 otherwise.

train itself to play through self-play and Q-learning.

IMPLEMENTATION:

1. For the implementation using Q-Table, two players are used. Human player class acts as a Greedy player which chooses the actions randomly except when it can capture a box.

Q-Player class implements the Q- learning algorithm. It consists of methods for updating and saving the Q-Table.

I have implemented the training for different number of games from 100 to 100000. The accuracy for winning games improves from 30% to 48.3%.

2. The same algorithm is then implemented for 3x3 board size. Average time for running the algorithm increased a lot and it requires a lot of games to get the accuracy as good as 2x2. It is shown in the notebook file. The outputs of Q-Table are stored as pickle file for both grid sizes.

Another approach for implementing on 3x3 grid is to divide the grid into four 2x2 grids and implement Q-learning on that and then select the maximum Q-value among them and take the respective action.

3. Now, instead of selecting action from Q-table by selecting the action which has maximum q-value in state-action table, I replaced the Q-table with the neural network implementation using PyTorch. The implementation output is shown in jupyter notebook.

RUN:

Dots-Boxes-NN → Run this for the 3x3 neural network implementation

Dots-Boxes-Qtable → Run this for Q-Table implementation. Change the grid size as per as requirement. The wins and losses are printed.