Identification of strategies in multiplayer games using Reinforcement Learning (RL)

Department of Mathematics,IIT Madras
Guide: - Dr. Sivaram Ambikasaran
Co-Guide: - Dr. Srivallabha Deevi, Tiger Analytics
Submitted by: Tumpa Jalua

March 2, 2024

Abstract

- Online interactions of customers with any app can be modeled as a two-player game.
- Reinforcement Learning is a suitable method to teach agents to play two-player games.
- In this project, we created an RL agent that learned to play the game of Tic-Tac-Toe.
- Three strategies were compared: 1. Two random players, 2. The first player is a computer player, and the second player is a random, 3. The first player is random, and the second player is the computer.
- We used the Q-learning algorithm to train and test the three strategies.
- Calculating the winning probability of both players in these three game strategies.

Introduction

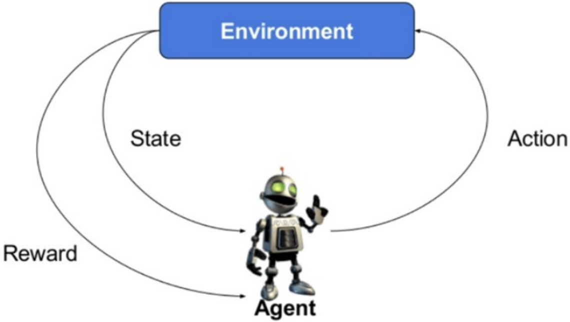
- Reinforcement Learning is a machine-learning model to train agents to play multi-step games.
- After each step, the machine receives a reward that is reflected, whether the step was good or bad in terms of achieving the target goal.
- By exploring its environment and exploiting the most rewarding steps, it learns to choose the best action at each stage.

Tic-Tac-Toe

- Used a 3x3 board, comprising a total of 9 cells.
- Computer (Player 1): Plays X and Random Player (Player 2): Plays O.
- The board will be represented with symbols: 0 for available positions, 1 for Player 1's moves, and -1 for Player 2's moves.
- Trained RL agent by trial and error using Q learning algorithm.
- The outcome of each game, i.e., whether it results in a win, loss, or tie, and assign corresponding rewards of 1, -1, or 0, respectively.

RL components

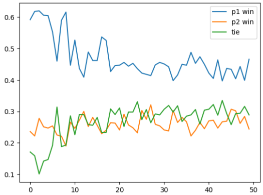
- State
- Environment
- Agent
- Action
- Reward



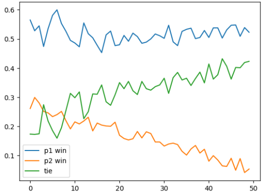
Q-Learning

- **Exploration:** Agents work on gathering more information to make the best overall decision.
- **Exploitation:** Agents make the best decision based on current information.
- **Epsilon greedy strategy:** To balance exploration and exploitation by choosing between exploration and exploitation based on a threshold.
- **Action Selection:**
- Available position on the game board.
- Random Action Selection (Exploration).
- Highest Expected Reward (Exploitation).

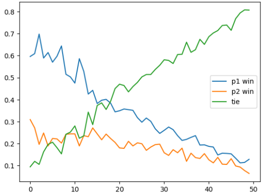
RL training



(a) Both players P1 and P2 trained with constant epsilon



(b) One player trained with decay epsilon and other player trained with constant epsilon



(c) Both players P1 and P2 trained with decay epsilon

Output

Two random players play each other and learn policies - P1 learns first player policy, P2 learns second player policy:

Player	Winning Probability
Random Player (P1)	57.40%
Random Player (P2)	43.60%

Computer player (P1) plays against a Random player (P2):

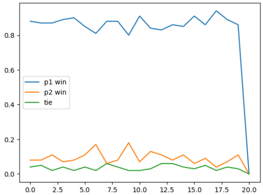
Test	Description	P1 win prob.	P2 win prob.	Tie Prob.
1	Both players trained with constant epsilon	96.90%	3.10%	0.00%
2	One player trained with decaying and other player trained with constant	98.40%	0.00%	1.60%
3	Both players trained with decaying	99.10%	0.00%	0.90%

Output

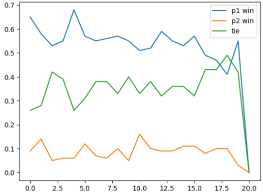
Random player (P1) plays against a Computer player (P2):

Test	Description	P1 win prob.	P2 win prob.	Tie Prob.
1	Both players trained with constant epsilon	49.20%	46.00%	4.80%
2	One player trained with decaying and other player trained with constant	49.50%	46.40%	4.10%
3	Both players trained with decaying	52.30%	44.10%	3.60%

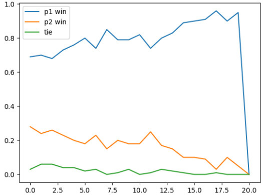
Pre-Neural network training



(a) Both players P1 and P2 trained with constant epsilon

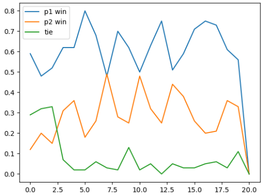


(b) One player trained with decay epsilon and other player trained with constant epsilon

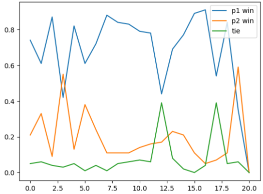


(c) Both players P1 and P2 trained with decay epsilon

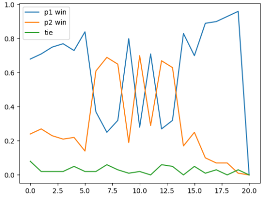
RL Neural network training



(a) Both players P1 and P2 trained with constant epsilon



(b) One player trained with decay epsilon and other player trained with constant epsilon



(c) Both players P1 and P2 trained with decay epsilon

Output used NN

Computer player (P1) plays against a Random player (P2):

Test	Description	P1 win prob.	P2 win prob.	Tie Prob.
1	Both players trained with constant epsilon	80.00%	10.30%	9.70%
2	One player trained with decaying and other player trained with constant	80.20%	12.40%	7.40%
3	Both players trained with decaying	76.20%	17.50%	6.30%

Output used NN

Random player (P1) plays against a Computer player (P2):

Test	Description	P1 win prob.	P2 win prob.	Tie Prob.
1	Both players trained with constant epsilon	34.30%	52.70%	13.00%
2	One player trained with decaying and other player trained with constant	41.80%	51.20%	7.00%
3	Both players trained with decaying	66.60%	25.00%	8.40%