Reinforcement Learning Model for Games

Team WJNKCW: Caleb Halter, Wafi Hussain, Brian Little, Nikki Meyer, James Peters, Kyle Petrie
Sponsored by: Dr. Mark Albert, University of North Texas
Student Mentors: Chengping Yuan and Ty Washburn

Project Overview

This project is based on using reinforcement learning to create artificial intelligence for board games.

Players engage with an AI created through reinforcement learning. The project show the decision-making process of the AI. The AI agent makes decisions based on a function of the potential rewards and tries to achieve the best outcome.

League play shows the AI's long-term decision-making process while trying to achieve a cumulative reward. Players can bet with the AI and see how it handles risk and reward.

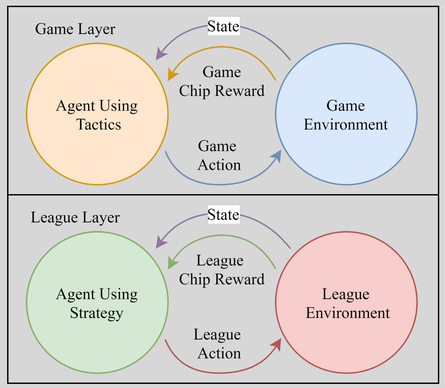


Figure 1. A chart displaying the manner in which an AI agent interacts with its environment using Reinforcement Learning

Motivation

The machine learning market is seeing phenomenal growth. This project is designed to be a demonstration of machine learning principles, especially to new students of various levels who might be interested in entering the field in the future. Interested students can see the fun applications of AI and understand how the AI thinks and behaves.

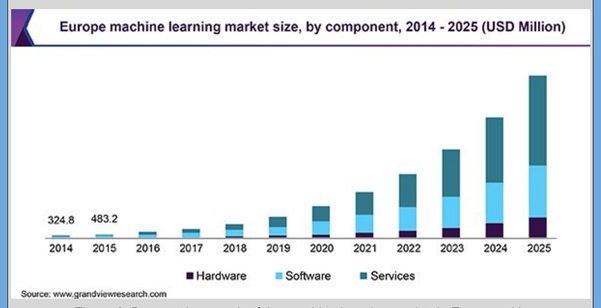


Figure 2. Data on the growth of the machine learning market in Europe, with projections for the future.

Game Features

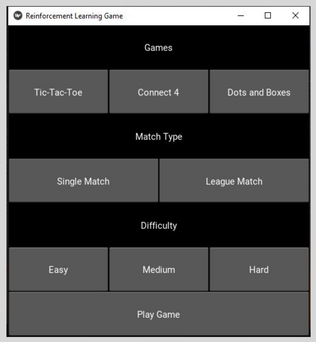


Image 1. The main menu of the project, with options for selecting game, match type, and difficulty.

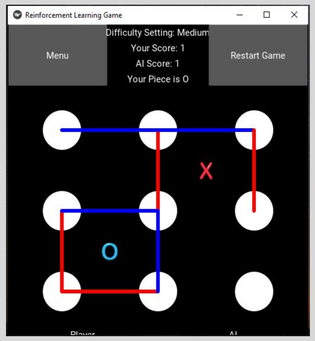


Image 2. Dots and Boxes

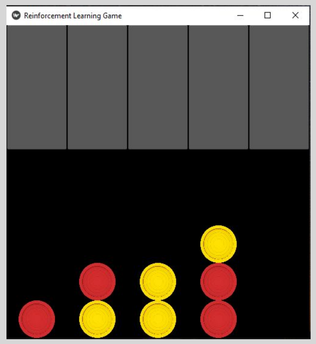


Image 3. Connect Four

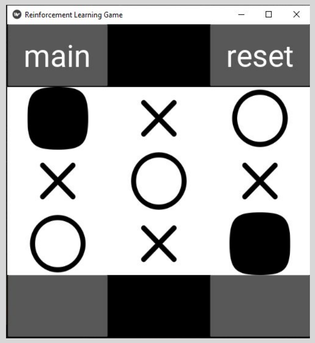


Image 4. Tic-Tac-Toe

- Three games
  - Tic-Tac-Toe
  - Connect Four
  - Dots and Boxes
- Three AI difficulties
  - Easy, Medium, Hard
- Two game modes
  - Single Match
  - League Play with betting

Technologies Used

- Python
- Kivy
- GitHub
- PyInstaller
- GitHub Actions

Reinforcement Learning

- Reinforcement learning teaches artificial intelligence through repeated exploration of options, finding the best course of action in a situation. It trains agents to achieve the best possible reward over cumulative decisions.

Figure 3. This flowchart displays the most basic aspect of reinforcement learning.

- The model gives values to different states based on reward.
- The table of these states and actions is called a q-table.
- Q-learning, a type of reinforcement learning, teaches an AI agent to pick the best action in a given situation using a function to optimize and predict rewards.

A Sample Q-table Demonstrating how the AI thinks

Q-Table	Actions				
		Up	Down	Left	Right
States	0	1.4	1.8	1.6	1.4
	1	1.6	2.5	2.2	3.2
	2	2.3	3.2	2.1	1
	3	0.2	2.3	3.1	0.4

Figure 4. This samples q-table demonstrates how actions and states intersect to hold weights or values to predict a future reward.

Acknowledgements

This project is the work of several groups of students carrying on each other's efforts. Special thanks to Chengping Yuan and Ty Washburn as student mentors and to Jakob Smith, Daniel McGartland, and Anthony Solorio as the previous student group working on the project.

League Play

League play is a key feature of this project and the demonstration of the reinforcement learning. The league play demonstrates the AI's behavior in an environment with potential loss and reward.

The user bets "chips" against the AI before playing a game. The AI can bet with the player, call, or quit the game if they decide the risk is not worth it. It displays the AI's long-term decision-making process, how it seeks for rewards, and how it avoids risk.

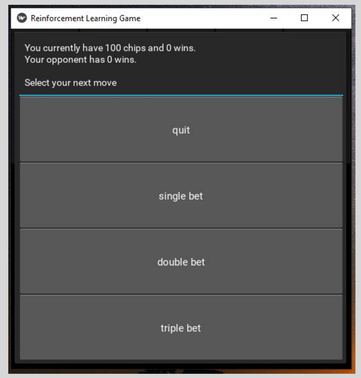
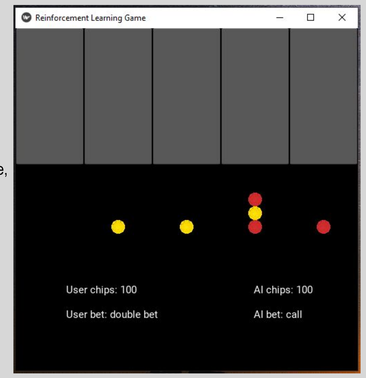


Image 5. The player is able to "gamble" with the AI to see it decide on long-term reward.

Image 6. The games are the same, but the AI will make betting decisions to achieve reward.



League play is the clearest example of the AI's reward-seeking and risk-avoiding behavior.

Contact information

- To contact the team
  - BrianLittle@my.unt.edu
  - bwlittle10@gmail.com