

Proposal: Reinforcement Learning in the Game "ConnectX"

LI JUNFENG(A0296986X, lijunfeng@u.nus.edu), ZHOU YIZHUO(A0162506L, e0134093@u.nus.edu), LI PEIRAN(A0289359H, lipeiran@u.nus.edu), ZHENG HANMO(A0307219H, e1391130@u.nus.edu)

Introduction

The proposal aims to develop a reinforcement learning (RL) model to play "**ConnectX**", a game where the objective is to align a certain number of checkers in a row, horizontally, vertically, or diagonally, before the opponent does. Reinforcement learning is a subfield of machine learning where agents learn to make decisions by interacting with an environment to maximize cumulative rewards. Notable achievements in Deep RL have been widely successful in game-playing scenarios. This project will involve creating an agent capable of learning strategies to win in **ConnectX** using various reinforcement learning techniques.

Objective

1. Develop a competitive RL agent as the baseline for "**ConnectX**" using models we have learned in the lecture and get a deeper understanding of these models.
2. Learn state-of-the-art RL techniques and try to develop an agent based on those techniques.
3. Get a deeper understanding of what we learn about RL from the lecture.
4. Try to optimise the model.

Methodology & Timeline

1. **Environment Setup**(Week 7): Using the "**ConnectX**" game environment.
2. **Agent Design**(Week 8-9): Building an RL agent based on various models.
3. **Training**(Week 10-11): Training the agent through self-play and competition against a computer player. Experience replay and target networks will be utilized to stabilize training.
4. **Comparison and Analysis**(Week 12): Compare the agent based on its ability to outperform other agents during the game.

Division of Work

1. Junfeng: Test the environment and implement an agent with a random strategy to make sure the code works properly.
2. Peiran: Implement MCTS agent.
3. Yizhuo: Implement a Q-learning agent.
4. Hanmo: Implement an agent with a more advanced RL algorithm.

Risk Assessment

1. **Model Instability**: The RL models could face stability issues during training, particularly if the exploration-exploitation balance is not handled effectively.
2. **Time Constraints**: Given the limited time, we may not be able to train the agents with various models.
3. **Computational Resources**: Training the RL agent might require significant computational power.

To mitigate these risks, **milestones** are established:

- **Naive Agent**: By Week 9, ensure a working version of the naive agent.