TD-Gomoku, Reinforcement Learning

Paul Garell (prg74) and Lauren Cali (lmc339)

Project Overview

The goal is to create a computer program that can play Gomoku competitively. Using an artificial neural network, we aim to iteratively train a model that can explore high level Gomoku strategies.

Gomoku is a two-player strategy game that traditionally uses a 15 by 15 Go board and white and black stones. Generally, the black stone plays first. Players alternate turns placing their pieces in a strategic manner at grid intersections. To win the game, a player must successfully get at least five of their pieces in a row unbroken vertically, horizontally, or diagonally. If the board fills and no player has achieved five consecutive pieces, the game ends in a draw.

The general approach will be modeled after the temporal difference Backgammon algorithm, TD-Gammon. In our implementation we will look at all possible legal moves and their possible responses. This requires looking two steps ahead, which on a 15 by 15 board will be bounded by 15^4 options. It follows that moves will be selected via specific weighted options. Each move will be ranked on its likelihood of bringing the algorithm closer to a win and the weights will be determined by an estimation of either white or black winning.

A crucial part of our approach is the algorithm is free of any prior knowledge or intervention. This means that it will learn via self-play, similar to the approach presented in Tesauro's 1992 paper.

Some interesting extensions to our project can involve adding intervention when nearing a complete board. This could mean if some percentage, say 90%, of the board becomes full, the model switches to a more analytical approach; this could be obtained via feeding specific end game cases. Additionally, we could look into changing the number of moves in advance. This could be beneficial since winning Gomoku requires getting five pieces in a row, so it begs the question of is the knowledge of four moves worth the computational tradeoff.

Approach

Our approach is to adapt Tesauro's 1992 approach to solving backgammon. While there is some additional overhead since there are more potential next moves, we believe it should still be possible. Our approach will implement a weight system based on the probability of winning the game. We plan to use an artificial neural network (ANN) and update weights based on proximity to current turn and maximum likelihood of a win.

Coherent Plan

We will use temporal difference learning. This involves using current states and past states to calculate the probability of future states. This means it is a dynamic program that will be updating weights iteratively throughout gameplay. Weights will be updated based on how close to the current turn they are; specifically we will use a hyperparameter λ between 0 and 1 to determine how many moves to historically update. Additionally, they can be weighted using a nonlinear function to further emphasize the importance of certain moves. The model will learn based on wins and losses and make predictions of whether white or black will win.

Timeline

***Stage 1*:** **Implement Gomoku**

Start: October 1, 2023 Target Completion: October 11, 2023

This stage involves system design. While actually building the game will not take two weeks, designing the high level architecture should be carefully thought through.

Responsibilities:

- Implement Gomoku (Lauren)
- Implement win detection (Paul)

Stage 2: **Research**

Start: October 1, 2023 Target Completion: November 1, 2023

This stage is primarily research oriented and should be started during Stage 1. Starting during stage 1 is beneficial as it may influence our system design choices.

Responsibilities:

- Historical temporal difference algorithms (Lauren)
- TD-Gammon, and Tesauro 1992 (Paul + Lauren)
- ANN's for game analysis (Paul)
- Game tree's? Maybe not directly useful, but definitely good to know (Paul + Lauren)

***Stage 3*:** **Model**

Start: October 25, 2023 Target Completion: November 22, 2023

This stage consists of building out the specifics of the model and determining hyperparameters. It will be somewhat time consuming because we will have to train small samples and determine the effectiveness of particular parameters. Additionally, we will investigate the effectiveness of different approaches that we find in our research stage.

Responsibilities:

- Build a weight function (Paul + Lauren)
- Build a preliminary ANN and test on small scale (Paul + Lauren)
- Test and experiment with hyperparameters to improve small scale performance (Paul + Lauren)
- Scale using cloud computing! (Paul)

***<u>Stage 4</u>*:** **Evaluation**

Start: November 15, 2023 Target Completion: November 29, 2023

This stage will predominantly revolve around testing and determining the effectiveness of our model and algorithm. This stage will also involve looking into how our Gomoku player competes against proven theory.

<u>Responsibilities</u>:

- Play against model to see how model fairs (Lauren has 100s of hours of logged play time)
- See how model fairs in online gameplay (Paul + Lauren)