Cardiff University

Initial Project Report

CM3203 - One Semester Project (40 Credits)

Neural Networks and their Ability to Improve Artificial Intelligent Go Players

January 30th 2016

Author:
Iain Majer - C1335600

Supervisor:
Dr. Frank C Langbein
Moderator:
Dr. Martin Caminada

Project Description

Go is a two player turn-based Chinese board game of perfect information and no chance, this means there is no missing information between the two players. In this game players take it in turns to place one of their tokens on the board in an attempt to capture their opponent's tokens (e.g. surround a token on all sides) and control over 50% of the board by the end of the game.[1]

Go has been adopted by the realm of Artificial Intelligence as it is such a complex game. The number of possible games is 10^{761}, a lot greater than that of chess with only 10^{120}. With the search space much greater than most other games it forces people use different AI techniques than those generally used, and in more creative ways.

Recently Google used a similar approach and created AlphaGo, an AI which takes the Monte-Carlo Tree Search and used two Neural Networks to lessen the width and depth they were searching through. I plan to look into this deeper and while looking at similar approaches to this problem.[2]

In this project I hope to show how Neural Networks can be used to improve the abilities of more common AI approaches that are used in Computer Go Players. I plan to do this by having a Neural Network influence the decisions of a standard AI in some manner. I will then test the AI's abilities by having it play against the original AI to see which is better in a selection of aspects (speed, player ability, memory usage, etc).

Aims and Objectives

The aim is to achieve this by creating a Framework which can be used to play Go. The Framework will be written in Java so to reduce time developing since I can concentrate more on the structure and usability of the code rather than the niches of a new language. I will use a Server and Client structure, so that I am able to run any AI I develop for this Framework on a different machine so the AI can use as many resources as it needs.

I will also have to create a API for the client programs to use to communicate in a standard way with the server. There will be two main client programs that use this API a more common AI approach (i.e. one more adopted by the community which seems to have a recognisable ability, but isn't too resource intensive). The other is the same approach but with the addition of a Neural Network augmenting the decision or process of the base AI.

Testing of these AI will be accomplished by a series of tests that will attempt to contrast their difference in abilities (e.g. speed, memory usage, ability, etc), and also play against both AI myself to subjectively evaluate which of the AI I feel is a better Go player.

---

[1] "Go (game) - Wikipedia, the free encyclopedia." 2011. 30 Jan. 2016 <https://en.wikipedia.org/wiki/Go_(game)>

[2] "AlphaGo | Google DeepMind." 2016. 30 Jan. 2016 <http://deepmind.com/alpha-go.html>

In summary I have the following aims:

- Determine well used AI models in the community
- Decide on an applicable model to augment with a Neural Network
- Design a Framework capable of running Go games
- Extend the Framework to allow AI to interact with the it
- Implement the Framework in Java
- Split the Framework into a Server and Client model
- Form an API to communicate with the Framework
- Create a basic AI using the API
- Modify the base AI to create a second AI with the addition of a Neural Network
- Compare the performance of the implemented AIs
- Conclude the effects that the Neural Network had on the Go AI

My hopes for this project is to show that you can use a Neural Network alongside a more common AI to help improve its ability to solve the problem it has been set out to accomplish, in this case playing Go. These improvements could be in speed, since this would allow some AI to search through more possible moves; resources, reducing the amount of processor power needed for the AI to actually perform as it needs means the extra processing power that has been freed up can be used to accomplish the same problem in a smaller time; ability, this would be the best outcome in my view since it means I have successfully helped the base AI with a problem space it wasn't entirely capable of handling on its own.

Project Plan

Time management for this project is going to be a key for accomplishing everything I want to achieve in the time I have. Below is a week by week breakdown stating a description of what I want to achieve that week along with a task list I hope to complete through the week. The order of the task list is the order I wish to complete the tasks in, and the number at the beginning of the task is how many days I plan to use to finish that task.

<u>Week 1 - 2 (1st February - 14th February):</u>

Start researching and documenting key information which is relevant to my project. Find what techniques people in the community advice, along with what has already been looked into, and what possible problems I may face further on in the project.

Task List:

- [1] Learn to play Go
- [1] Play some games of Go to earn a decent understanding of the game's mechanics
- [2] Find what AI models suit the problem space of playing Go
- [2] Define their advantages and disadvantages
- [2] Determine which of these models would be best to be used as a base AI to improve
- [1] Write up findings in report

Aims to be achieved:

- Determine well used AI models in the community
- Decide on an applicable model to augment with a Neural Network

Week 3 - 4 (15th February - 28th February):

I hope to design and layout the structure of my Framework, likely with a UML Diagram, along with other documentation describing briefly what some key parts of the Framework need to accomplish.

Task List:

- [1] Determine the structure of the network messages between the server and client API
- [2] Create a UML Diagram for the server side
- [2] Create a UML Diagram for the client API side
- [3] Create a UML Diagram for the base AI
- [3] Create a UML Diagram for the augmented AI
- [1] Document and describe how parts of the system are going to work

Aims that will be achieved:

- Design a Framework capable of running Go games
- Extend the Framework to allow AI to interact with it

Week 5 - 7 (29th February - 20th March):

This section of the project will be where I actually implement my design from the last section and the one that is most likely to take the most time to complete.

Task List:

- [1] Set up a git repository and workspace environment for the project
- [1] Lay out the class structure from the designed UML in Java for the server
- [1] Lay out the class structure from the designed UML in Java for the client API
- [4] Implement the server functionality for storing a Go game, and checking rules and win conditions
- [2] Create the API functionality for sending the desired messages to the server
- [5] Implementing the base AI with the client API
- [5] Implementing the augmented AI
- [2] Debugging

Aims that will be achieved:

- Implement the Framework in Java
- Split the Framework into a Server and Client model
- Form an API to communicate with the Framework
- Create a basic AI using the API
- Modify the base AI to create a second AI with the addition of a Neural Network

Week 8 - 10 (21th March - 10rd April):
These weeks will be dedicated to data collection and testing the two AIs against each other.

Task List:

- [1] Design an objective speed comparison
- [1] Design objective resource usage analysis
- [1] Design a test to compare their player ability
- [3] Run speed test
- [3] Run resource test
- [3] Run ability test
- [2] Play against both AI personally for subjective comparison

Aims that will be achieved:

- Compare the performance of the implemented AIs

Week 10 - 12 (11th April - 1st May):
Finally in the project I will compare the results I gathered from the previous section and show what conclusions can be drawn from them.

Task List:

- [1] Show a comparison of the data retrieved in the tests
- [4] Write what conclusions can be made
- [2] Suggest the reason for these results
- [4] Compare against original hypothesis and assumptions

Aims that will be achieved:

- Conclude the effects that the Neural Network had on the Go AI

As you can see in my plan the cumulative time taken for all the tasks in a given week is less than the amount of days in those weeks (excluding implementation). This will allow for some tasks to stretch further than their allocated time, and also for some jobs to be done before they are stated. It also leaves room for extra work to be done alongside this project.