Improving CNN Draughts Evaluators using Genetic Algorithms

Student Name: Thien P. Nguyen

Supervisor Name: Stefan Dantchev

Submitted as part of the degree of BSc Computer Science to the

Board of Examiners in the School of Engineering and Computing Sciences, Durham University

Abstract —

Background

Presently, competitive Draughts AI players are currently designed to play at a fixed ability. While it has produced very competitive and intelligent players, they require manual modifications in order to improve its performance. This is due to their dependency on pre-defined move databases, where optimal moves are pre-calculated, and recalled when necessary. By combining Neural Networks and Genetic Algorithms, this issue could possibly be solved by creating a player that can grow in ability over time, without the dependency on move-banks.

Aims

The purpose of this project is to explore approaches to tackle the game of English Draughts via the use of machine learning techniques. First, we study previous historical successes in the field, and look at the components that helped build their systems. Then, we look at contemporary methods of computer science that could be used to evolve the historical systems. The project will establish whether this approach provides an effective performance on the game.

Method

The initial population will consist of randomly generated AI players, which will play each other to determine the best player out of the population. The performance of championing AI players at every generation of the genetic algorithm are measured against previous champions. Appropiate algorithms are implemented to detect the overall development of the system's ability to play Checkers.

Proposed Solution.

The proposed solution starts with designing a neural network that evaluates the probability of a particular side winning, given a given state of a checkerboard. This is then used in a algorithm that evaluates future moves to predict the best move at a given position. This, alongside a set of weights for the neural network, creates a player that can evaluate potential moves. Finally, the player is then used on an existing Draughts framework that will provide the player with the ability to play Draughts.

Keywords — AI, Neural Networks, Genetic Algorithms, MiniMax, Alpha Beta Pruning, Draughts

I INTRODUCTION

The intention of this project is to explore the effectiveness of genetic algorithms to improve the evaluation of a neural network's probability to determine the performance of two players in a game of checkers. We attempt to use various crossover and mutation strategies to manipulate the weights of the network, and compare their performance relative to the overall performance of the system.

Draughts

English Draughts (or Checkers) is a popular 2-player boardgame played on an 8x8 chessboard. Players begin with 12 pieces each, and they are placed on light-coloured squares. Each player takes a turn to move a piece diagonally in one square. They also have the option to capture their opponments piece by moving two consecutive diagonal squares, where the opponments piece is placed immediately opposite the players piece. Pieces can be captured consecutively in a single turn if the moves afford the scenario. In the event that a piece reaches the opposite side of the board from where the piece started with, they are promoted to a 'King' piece. King pieces have the ability to traverse backwards in the same diagonal motion as pawns.

Genetic Algorithms

Genetic algorithms (GAs) are a group of search techniques used to find exact or approximate solutions to optimisation and search problems. It borrows techniques from Charles Darwin's evolutionism theory; individuals are created by the crossover of the genetic information of their parents. Genetic algorithms are a subset of evolutionary algorithms, where the larger group is also formed of similar strategies not including Evolutionary Programming [Fogel, 1993][McDonnel, 1993], and genetic programming [Koza, 1991].

Neural Networks

Neural Networks are non-linear statistical data-modelling tools, linking inputs and outputs adaptively in a learning process similar to how the human brain operates. Networks consist of units, described as neurons, joined by a set of rules and weights. The units are defined with characterisitics, and appear in layers. The first layer is defined as the input layer, and the last layer being the output. Layers between the two aforementioned are described as hidden layers. Data is analysed by processing them through the layers.

Learning takes place in the form of the manipulation of the weights connecting the units in the layers. This allows it to model complex relationships between inputs and output, and it can also find patterns in the data.

Motivation

Whilst the use of evolutionary algorithms and neural networks have been explored to create draughts players, my intention is to explore a subset of evolutionary algorithms to determine their viability. Can we produce a similarly performant draughts evaluator by using seperate classifiers for the different stages of the game? Can the use of genetic algorithms and neural networks (GANNs) make a competent Draughts player?

Deliverables

Minimum

- Implement a CNN
- Implement a Checkers Game Interface

- Implement a genetic algorithm with an evaluation function that consists of a round robin tournament against the population of CNN Evaluators.
- Implement a mini-max algorithm that chooses moves.

Intermediate

- A user-friendly interface to play against the AI
- A monte-carlo search of the move space.
- Analysis of Crossover methods (within Genetic Algorithms)
- Analysis of Mutation methods (within Genetic Algorithms)

Advanced

- Convolutional Neural Network Layer analysis
- The resulting AI can play to an ELO of at least 1200.

Related Work

II DESIGN

Requirements

A checkers gameboard is created
Agents are able to select a legal move
Agents are able to harness neural networks to assist in their move decision
Champions are able to create offspring.
The weights and biases of the Agent's neural network are saved to a form of storage.
Agents only choose valid, legal moves
Agents always return a valid, legal move
Agents make maximum use of the turn time where appropriate Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Choice of Programming Language

The project is to be written in Python 3.6 due to my familiarity and the support of popular scientific packages including NumPy and other machine learning tools. Python is also very portable with a very wide compatability, being pre-installed on all popular UNIX machines and university support. Object Oriented approaches are taken for the majority of the components of the system, ranging from the neural network library to the tournament system. Data structures are implemented using their own classes and methods where applicable. Players weights (for their neural networks) are stored in two forms, one of which is to be stored on an MongoDB NoSQL instance, and another local copy in JSON. This allows the individual agents to be played against humans.

Tools

The resulting program will be simulated on Durham's MIRA (128-core Intel) distributed system for 4-ply heavy loads, and debugging will occur on a lighter machine (4-core Intel i5 6200u). In order to keep simulations running on MIRA, MOSH is used to maintain a consistent UDP SSH connection to MIRA. The end champion is then hosted online on a Heroku instance as an API.

Architecture

Should include a diagram of the algorithm workflow

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Algorithms and Data Structures

.1 Neural Network

In order to evaluate the board, we use a feed-forward neural network in the form of the following node layers 91,40,10,1 where the input layer consists of 32 nodes, with the output node having 1. Our sigmoidal function of choice is the hyperbolic tangent. The input layer is the board, preprocessed to cover all possible subsquares of the checkerboard, rangig from a 3x3 kernel to a 8x8. And here we see figure 1.

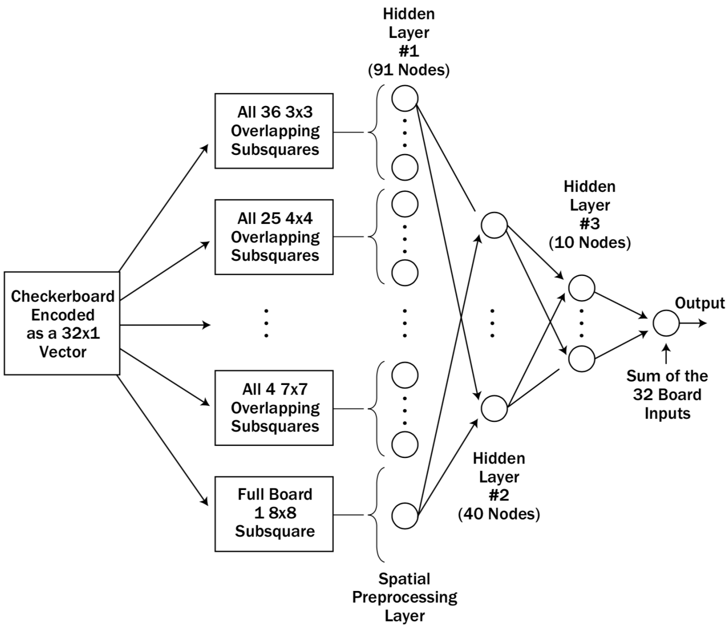

Figure 1: The chosen neural network model. Note that the checkboard is preprocessed.

.2 MiniMax Decision Making

The minimax algorithm revolves around the use of xxyy

.3 Tournament Method

The tournament algorithm follows the following pseudocode in Algorithm 1.

Algorithm 1 My algorithm

procedure MYPROCEDURE
    stringlen $\leftarrow$ length of *string*
    $i \leftarrow$ *patlen*
top:
    if $i >$ *stringlen* **then return** false
    $j \leftarrow$ *patlen*
loop:
    if *string*$(i) =$ *path*(j) **then**
        $j \leftarrow j - 1.$
        $i \leftarrow i - 1.$
        goto *loop*.
        close;
    $i \leftarrow i + \max(delta_1(string(i)), delta_2(j)).$
    goto *top*.

.4 Children Generation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

.5 Coefficent Mutation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

.6 Crossover Mechanism

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Testing and Evaluation

At the end of a given generation, we measure growth of performance using the champion of the generation. The score is calculated using the following formula, whatever it is.

Point Score for the champion games are measured by 1,0,-1 where a Win counts as 1 point and -1 for a loss. The weights are scaled differently to the regular tournament in order to accurately portray the difference between previous champions. At the end of the generation run, the end player will be used to compete against human players on ?? in order to determine an accurate ELO rating of the player.

Several approaches have been considered, ranging from the use of ELO ratings to Championship Knockout approaches, but

Presently we will use the medians of medians approach. The curren

A Figures and Tables

In general, figures and tables should not appear before they are cited. Place figure captions below the figures; place table titles above the tables. If your figure has two parts, for example, include the labels “(a)” and “(b)” as part of the artwork. Please verify that figures and tables you mention in the text actually exist. make sure that all tables and figures are numbered as shown in Table 1 and Figure 1.

Table 1: UNITS FOR MAGNETIC PROPERTIES

Symbol	Quantity	Conversion from Gaussian

B References

The list of cited references should appear at the end of the report, ordered alphabetically by the surnames of the first authors. The default style for references cited in the main text is the Harvard (author, date) format. When citing a section in a book, please give the relevant page numbers, as in (Budgen 2003, p293). When citing, where there are either one or two authors, use the names, but if there are more than two, give the first one and use “et al.” as in , except where this would be ambiguous, in which case use all author names.

You need to give all authors’ names in each reference. Do not use “et al.” unless there are more than five authors. Papers that have not been published should be cited as “unpublished” (Euther 2006). Papers that have been submitted or accepted for publication should be cited as “submitted for publication” as in (Futher 2006) . You can also cite using just the year when the author’s name appears in the text, as in “but according to Futher (2006), we ...”. Where an authors has more than one publication in a year, add ‘a’, ‘b’ etc. after the year.

References

Budgen, D. (2003), *Software Design*, 2nd edn, Addison Wesley.

Euther, K. (2006), Title of paper. unpublished.

Futher, R. (2006), Title of paper 2. submitted for publication.