Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia
Social and Behavioral Sciences

Procedia - Social and Behavioral Sciences 57 (2012) 419 – 425

International Conference on Asia Pacific Business Innovation and Technology Management

Evolutionary Neural Network for Othello Game

Gunawan[a*], Hendrawan Armanto[a†], Joan Santoso[a‡], Daniel Giovanni[a], Faris Kurniawan[a], Ricky Yudianto[a] and Steven[a]

[a]Computer Science Department, Sekolah Tinggi Teknik Surabaya, Ngagel Jaya Tengah 73-77 Surabaya 60284, Indonesia

Abstract

Game playing is a game method that require an AI (Artificial Intelligence), so that an AI can play against human in a game. Artificial intelligence involves two basic ideas[4]. First, it involves studying the thought processes of human beings. Second, it deals with representing those processes via machines (like computers, robots, etc.). AI is behavior of a machine, which, if performed by a human being, would be called intelligent. It makes machines smarter and more useful, and is less expensive than natural intelligence.

Othello is one example of game playing using AI. Even though it may appear as though Othello is a fairly simple game, there still are many important aspects of the game to consider. The most important of these are the evaluation function and searching algorithms. Why are these important? First of all, the game would be nothing without an evaluation function. And there are many interesting aspects of the evaluation which can greatly affect both efficiency as well as game play. Second, a good searching algorithm can fulfill the ideal properties of a good heuristic, providing a good answer in a reasonable amount of time.

Keywords: Game Playing; Neural Network; Reversi; Genetic; Evolutionary Neural Network

1. Introduction

Reversi (marketed by Pressman under the trade name Othello) is a strategic boardgames which involve play by two parties on an eight-by-eight square grid with pieces that have two distinct sides[5]. Pieces typically appear coin-like, with a light and a dark face. There are 64 identical pieces called 'disks' (often spelled 'discs'). The basic rule of othello, if there are player's discs between opponent's discs, then the discs that belong to the player become the opponent's discs.

Othello is a two player game that has a few mode such as player versus player, player versus computer, or even computer versus computer. The goal is to win by having more discs than your opponent. Though the original rules of othello were such that each player was limited to using no more than half of the disks (those in possession at the start), this rule has long been out of common practice; and, if using a physical board and pieces, the player whose turn it is simply retrieves a disk that is in possession of the opponent as needed.

* *Gunawan. Tel.:+62-31-5027920; fax:+62-31-5041509.*
E-mail address: hansmichaelgunawan@gmail.com
† *Hendrawan Armanto. Tel.:+62-31-5027920; fax:+62-31-5041509.*
E-mail address: hendrawan.armanto@acm.org/hendrawanarmanto@gmail.com
‡ *Joan Santoso. Tel.:+62-31-5027920; fax:+62-31-5041509.*
E-mail address: joan.santoso@acm.org/joan.santoso@gmail.com

doi:10.1016/j.sbspro.2012.09.1206

This means that there is now only one way a player will pass (always involuntarily) rather than place a disk, while formerly there were two. Each player's objective is generally to have as many disks one's own color at the end as possible and for one's opponent to have as few or, technically in consideration of the occasional game in which not all disks are placed, that the difference between the two should be as large as possible if the winner and as small as possible if the loser. However, simply winning is the basic goal, and maximizing the 'disk differential' is regarded as ancillary.

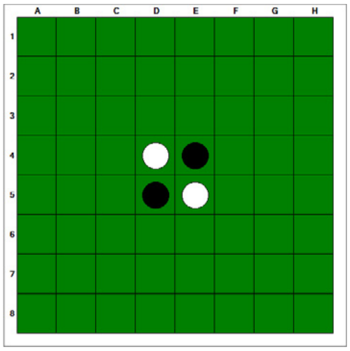

Fig. 1. Starting Position

The original rules of Reversi stipulated that the first two moves by each player, beginning on an empty board, were to place on one of the 4 central squares (Fig. 1) that remained unoccupied (so that there is no choice for the 2nd player's 2nd move) a disk of one's color, so that an essentially alternate starting position from the modern game could be forced by either player. Othello's rules, however, state that the game begins with four disks placed in a square in the middle of the grid, two facing light-up, two pieces with the dark side up, with same-colored disks on a diagonal with each other, and this is nearly universal practise in reversi play today. Convention has initial board position such that the disks with dark side up are lined up top right and bottom left, though this is only marginally meaningful to play (Where opening memorization is an issue, some players may benefit from consistency on this). The dark player makes the first move.

2. Game Playing Othello

2.1. Alpha Beta Search Algorithm

The alphabeta algorithm is a method for speeding up the minimax[6] searching routine by pruning off cases that will not be used anyway. This method takes advantage of the knowledge that every other level in the tree will maximize and every other level will minimize. It works as follows: start off with alpha=-infinity and beta=infitity; traverse the tree until the depth limit is reached; assign a value for alpha or beta based upon what level preceeded the depthlimit level. Whenever max is called, it checks to see if the evaluation of the move it has been given is greater than or equal to beta. If it is, then max returns that value would not have been chosen by min anyway and neither would the subtree that max would have created, so it is a waste of time searching through them. The same logic occurs with min except that it checks if the move it has been given is less than or equal to alpha.

```
// board : current board boardition
// possibleMoves: search depth
// alpha: lower bound of expected value of the tree
// beta: upper bound of expected value of the tree
int AlphaBeta(board, possibleMoves, alpha, beta)
{
    if (possibleMoves=maxDepth || game is over) return Eval (board); //evaluate leaf boardition from current player’s standpoint
    result = - INFINITY; // preset return value
    possibleMoves = GeneratePossibleMoves(board); //generate successor  moves
    for i =1 to count(possibleMoves) do //look over all moves
    {
            execute(possibleMoves[i]); //execute current move
```

```
        value = -AlphaBeta(board, possibleMoves+1, -beta, -alpha); //call other player, and switch sign of returned value
        if (value > result) result = value; //compare returned value and result value, note new best result if necessary
        if (result > alpha) alpha = result; //adjust the search window
        Undo(possible_moves[i]); // retract current move
        if (alpha >= beta) return alpha; //cut off
    }
    return result;
}
```

Fig. 2. Pseudocode for Alpha-Beta Algorithm

Alphabeta can be called with a variable number of window size between alpha and beta. The smaller the window size, the larger the number of cutoffs there will be - anything that falls outside of the window is a cutoff. Null window alphabeta means that alpha = beta-1, the smallest window size possible - with the most cutoffs. It has been proven that a return value result of alphabeta with a window of will be one of three cases: beta < result < alpha implies result is equal to the minimax value desired; result <= beta implies that result "failed low" meaning result is an upper bound on the true minimax value desired; result >= alpha implies that result "failed high" meaning result is an lower bound on the true minimax value desired.

2.2. Neural Network

An Artificial Neural Network, usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. The structures contain 3 layer, input layer, hidden layer, and output layer. There are 64 nodes in input layer, represent the condition of board value. Connected to each other layer makes a fully connected graph structure. Using the basis of a rule of thumb[2], hidden layer consists approximately 2/3 the size of input layer, which make 42 nodes in hidden layer. The output layer consist only 1 node to represent the value after calculation.

A function that used to calculate Activation Function is Sigmoid Function, the most commonly used for calculate the Activation Function, where:

$$f(x) = \frac{1}{1+e^{-x}}.$$

In this research, Neural Network will be used to calculate the Static Board Evaluation. For each possible moves and leaf nodes in expanded node will be return a value from this Neural Network. The weight and threshold will continuously be updated when the computer learnt.

2.3. Genetic Algorithm

Genetic Algorithm developed by John Holland and the team in 1975 at University of Michigan. In that research, the team perform a trial to utilize the concept of evolution process in a software to find a solution. Basic concept that inspire Genetic Algorithm is evolution theory that Charles Darwin proposed[1]. In evolution theory, each species must adapt against their environment in order to survive. Each individu in a population must compete with other individu to fight against something vital. So that the best individu will survive, and the other will extinct. The concept that John Holland made is robustness, and also the balance between efficiency and triumph of a system to survive (survival of fittest) in every condition.

Inspired by genetic science, so the term used here will be using the term that used by genetic science. Individu in a population called string (strings) or genotype. In genetic algorithm, each individu only have on chromosome. Chromosomes consist of gene/character/decoder which composed linierly. Position which occupied by gene in a chromosome called loci. The value contained in the gene called alle. Data type of alle can be a binary, floating point, or an integer depending on the genetic representation used. While the alle combined can give a value to chromosome, this called phenotype. The genetic algorithm start with a population that produced at random. This population will be considered as the first solution being tested. Calculate the fitness value from each individu will obtain the best solution. The function used to calculate fitness value depend on the problem.

2.3.1 The Flow of Genetic Algorithm

First is initialize the first population. The initial population will take effect to Genetic Algorithm's perfomance[1]. If initial population not vary enough, the probability to get the best individu is low. More vary of the initial population will produce many answer which can be a best solution. Second is selection. Selection process responsible to select each individu that will be generated as a new population. In the selection process, diversity of population take a vital role. There are several things in selection process that must noted:

- Sampling area
- Selection probability
- Selection mechanism

2.4. Evolutionary Neural Network

Many different ways can be used to create an Othello AI (Artificial Intelligence) player. One of the most powerful ways to create an AI is using Neural Network. With Neural Network, the computer can learn and update his move. Combined by the genetic, Evolutionary Neural Network will find the best move.

The evolution change each weights become a better weight in process. Always updating and learnt, that is the main view for this research. The best individu has the chromosomes that fit to each calculation. Each value that stored in chromosomes will be useful in Neural Network. The generating process will always pick the best individu with the best fitness value to generate. And so the offspring will have the chromosomes of the parent.

Neural Network's design uses each value of the chromosomes to calculate. With the full directed graph design, each value will affect the final result of the calculation. The output or the final result will be the return value which used to determine the value of each leaf node in the game tree. Using the chromosomes from the best individu, Neural Network will produce the value that become the best move which the computer must choose.

2.4.1.Genetic Algorithm on ENN

For this research, Genetic Algorithm will be used to find the best individu for playing Othello. Each individu for playing Othello consists of the chromosomes, which represent the weight that used to calculate in Neural Network. The structures of the chromosomes are a floating number. The genetic algorithm used Roulette Wheel operation to select and create a new individu. To create a new individu, elitism and mutation is used. Elitism will be clonning the last individu to be the new individu. On the other side, mutation will be create a new variation of individu, with the calculation:

$$X_{n+1} = X_n + (randomDouble * MutationProbabilty * (X_n - X_{n-1})$$

Mutation Probability is used to decide, whether this individu will be generated as a new individu or not. The new chromosomes will be using the previous chromosomes and the present chromosomes as a calculation.

Weight v_1	*Weight* v_n	*Weight* w_1	*Weight* w_n	*Threshold*	...

Fig. 3. Picture of Chromosome

2.4.2.Evolutionary Neural Network Initialization

The first step to do is preparing a population. A population will be generated 20 individu. At the beginning, the weight and thresholds from each individu will be randomized using Nguyen and Widrow Algorithm.

1. Define factor Nguyen Widrow,

$$\beta = 0.7(p)^{1/n}$$

   where, n = total node in input layer, p = total node in hidden layer;
2. Random each weight between -0.2 and 0.2[3];
3. Calculate vector length for weight ;

$$||Vi|| = \sqrt{V_{i1}^{\ 2} + V_{i2}^{\ 2} + V_{i3}^{\ 2} + \cdots + V_{in}^{\ 2}}$$

4. Update new weight;

$$V_{i\,new} = \frac{\beta \,.\, V_{i\,old}}{||Vi||}$$

5. Calculate threshold value ;

$$Threshold = random(-\beta, \beta)$$

2.4.3.Evolutionary Neural Network Finding Best Individu

Using a league system, each individu will be battled with each other. 2 individu will be playing a full round Othello. Fitness value for individu will be decided through this league, whenever an individu win a round, he get plus 3 poin for fitness value. Plus 1 poin if draw, and no poin if lose. The fitness value used to find the best individu.

-	A	B	C	D	E
A	-	W	W	D	W
B	W	-	L	L	W
C	W	L	-	D	W
D	W	D	W	-	D
E	L	W	D	W	-

From the example league above, the fitness value for each individu will be like this:

A : (3+3+1+3) = 10
B : (3+0+0+3) = 6
C : (3+0+1+3) = 7
D : (3+1+3+1) = 8
E : (0+3+1+3) = 7

Individu A will be choosen to be the best individu. Being the best individu means, the chromosomes will be used in benchmarking process (Section 2.7).

2.4.4.Evolutionary Neural Network Benchmarking

The population have to evolving into a better population. To see a population really evolving to a better population, benchmarking test is used[2]. The best individu will be playing 100 games against an opponent that using Negamax algorithm with some Static Board Evalution by using greedy concept, forfeit, mobility, stability, frontier, corner, and parity[7]; which will be summed and multiplied by the weight of each to get a good Static Board Evaluation.

Greedy concept is intended in the last game where players who have the most discs wins. Forfeit will be worth very advantageous if the opponent does not have movement, so the board position favorable for the player. The concept of mobility is also the key to reverse the game. If you can achieve a position where you can restrict the availability of moves to your opponent, then you are well on the way to victory. Stability is a stable discs that certainly will never be owned by the opponent. Players who have many stable discs have a great potential for a victory. Maintaining a small set of frontier discs will prove just as useful as it does in the regular game as it serves to increase your relative mobility. The player who put the disc in the corner, will have a large value of stability, because the discs can not be owned by the opponent. At mid-game, empty board positions will be divided into many groups that known as parity. Player who has the possibility of movement on the empty boards that have an odd parity, will give advantage to that players and made that players have the ending move in the group. [8]

At the time of benchmarking, program using 1 poin system and the test will result whether the individu is evolving or not. When the result of this battle is not good enough for an individu, generating a new population will be performed. Generating a new population uses Genetic Algorithm in section 2.4.1

3. Testing Result

The training for each match takes approximately 1-2 minute. The training in the league system will fight for 400 match, so total time taken to generate one population and obtain the best individu is approximately 400-600 minute plus benchmarking against another computer. The training has twenty individual in a population. The mutation probability uses 0.05. The benchmarking win rate must above 75% from the total matches.

The testing process matched the best individu from the last generation with 2 different computer players. The first computer player take from Apple's games, Tournament Reversi. The testing process' engage the best individu 20 times with the hardest difficulty. The statistic showed as below:

No	ENN			Tournament Reversi	
	Time / sec	Score	Winner	Score	Winner
1	163	36	√	28	-
2	180	36	√	28	-
3	150	36	√	28	-
4	154	36	√	28	-
5	170	36	√	28	-

Fig. 4. Picture of Statistic Game 1

Another testing process' engage the best individu with a computer player using Negamax as it algorithm method. The statistic showed as below:

No	ENN			Negamax	
	Time / sec	Score	Winner	Score	Winner
1	61	43	√	21	-
2	87	39	√	25	-
3	52	38	√	26	-
4	71	44	√	20	-
5	51	48	√	16	-
6	77	28	-	36	√
7	69	36	√	28	-
8	57	41	√	23	-
9	82	41	√	23	-
10	80	26	-	38	√
11	89	41	√	23	-
12	86	30	-	34	√
13	90	43	√	21	-

Fig. 5. Picture of Statistic Game 2

4. Conclusion

Our work is a research to prove that evolutionary neural network can produce a better quality static board evaluation rather than normal static board evaluation function. From our research, the testing prove that our AI is good enough to defeat the normal static board evaluation function that is using negamax algorithm. So based our research we get several conclusion that describe as follow:

- Speed up the calculation process evaluation board in the search tree.
- The weights are updating each time they learnt from genetic algorithm iteration process.
- Always create the best individu by having a generation full of winner's individu.

With our test statistic, we can get a good result against another AI. For the first testing, we get a 5 won in row against Tournament Reversi Program. From the second testing, we get 10 won from 13 games against a negamax alpha beta pruning algorithm.

References

[1] Kuswara Setiawan, Paradigma Sistem Cerdas 3rd Edition.

[2] Alexander J. Pinkney, Development of an Artificial Neural Network to Play Othello, 2009.

[3] Siang Y. Chong, Mei K. Tan, and Jonathon D. White, Observing the Evolution of Neural Networks Learning to Play the Game of Othello, 2005.

[4] K. Chitra, R. Alamelumangai, Artificial Intelligence, 2012.

[5] Wikipedia, encyclopedia, Reversi, http://en.wikipedia.org/wiki/Reversi, accessed on 14 May 2012.

[6] Joseph Melnyk, Advanced Artificial Intelligence, http://www.cs.kent.edu/~jmelnyk/othello, accessed on 15 May 2012.

[7] Steve MacGuire, Strategy Guide for Reversi & Reversed Reversi, http://www.samsoft.org.uk/reversi/strategy.htm, accessed on 12 May 2012.

[8] Tseng M.L. (Jan. 2011). Using a hybrid MCDM method to evaluate firm environmental knowledge management in uncertainty. *Applied Soft Computing* 11(1), 1340~1352.