Evaluation of Chess Position by Modular Neural Network Generated by Genetic Algorithm

Mathieu Autonès, Ariel Beck, Philippe Camacho, Nicolas Lassabe, Hervé Luga, and François Scharffe

Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier, 118 route de Narbonne 31062 Toulouse cedex, France

Abstract. In this article we present our chess engine Tempo. One of the major difficulties for this type of program lies in the function for evaluating game positions. This function is composed of a large number of parameters which have to be determined and then adjusted. We propose an alternative which consists in replacing this function by an artificial neuron network (ANN). Without topological knowledge of this complex network, we use the evolutionist methods for its inception, thus enabling us to obtain, among other things, a modular network. Finally, we present our results:

- reproduction of the XOR function which validates the method used
- generation of an evaluation function

1 Introduction

The game position evaluation function is a key part in a chess engine. It is composed of a long list of parameters [1], and using a genetic algorithm (GA) [8] to optimise them is relatively efficient, these parameters being obtained from extensive game experience. Another method, which consists of substituting an ANN for this list seems more interesting, because of the generalising capabilities of this model. In practice it is more difficult to implement. Network topology determination is the biggest problem. Evolutionist methods may help us, by evolving a network population which codes this function. We could code the matrix just as it is, in the chromosome [9], connections evolving through successive generations. This coding turns out to be unsatisfactory: the matrix size has to be prefixed and it is hard to predict it a priori. Network encoding then becomes the main problem of our study.

Boers and Kuiper's work [2] allows us, by using L-Systems, to generate modular neural networks whose size is independent of that of the chromosome, and crossover tolerant. We generate a population of L-System construction rules and then mark the resulting networks according to their capabilities to learn game position evaluations from real games. These positions are evaluated more and more deeply in the game tree, along with the increased complexity of the network.

M. Keijzer et al. (Eds.): EuroGP 2004, LNCS 3003, pp. 1–10, 2004.

2 Chess Engine

A chess engine contains three distinct parts: the management of the rules, investigation of the different variant pathways using a search algorithm, and the evaluation function.

2.1 Chess Rules

All chess engines need to know the rules to generate the legal moves or referee a game between two people. All legal moves are pre-calculated in the tables: the engine just needs to confirm that this move is one of them.

2.2 Search Algorithm

The search algorithm explores all the moves from a position and tries to find the best move. In our programme, we use the alphabeta algorithm with various heuristics [10], which are not mentioned here. The values of tree leaves are computed using an evaluation function, which is an ANN in this case.

2.3 Evaluation Function

The evaluation function is very important in all chess engines. It is very complex because it gives the final mark which it uses to select the move to play. The main operation of this function is to count the values of the pieces. After that, it is possible to refine the function, using a series of parameters to define:

- the king's safety
- maintenance of the bishop pair
- domination of the centre
- occupation of the open columns by the rooks.
- ...

We then calculate the sum of all the parameters to obtain the final mark. One of the main limitations of this technique is that we have to define the list of parameters and to set them up correctly, knowing that certain of these values will change during the game and are not self-compensating.

3 Neural Network

3.1 Presentation

Introduction. The human brain is certainly a most amazing organ. It is not a surprise if people try to pierce the secrets of its functioning, to recreate certain of its mechanisms artificially. Neural networks are directly inspired by this vision. The ANN are inspired directly by the structure of the human brain, which is schematically a field of neurons linked together. A Neuron is composed of three parts: dendrites, body and axon. The dendrites get the information (electrical impulses) from the other neurons. The body makes the sum of all this electrical information and if it goes beyond a certain level , the axon is activated, that is to say the neuron sends an electrical impulse to its successors.