A Neurogenetic Framework for Problem Solution Applied to a Game Situation

Marvin O. Schneider

University of Karlsruhe (TH), Institute IAKS, Germany

João Luís G. Rosa*

*Pontifícia Universidade Católica de Campinas, CEATEC, Brazil

Abstract

This paper presents Neurogenetic Connect Four (NGC4), an evolutionary connectionist game solution, which, in a novel attempt, combines evolution and artificial neural networks in order to automatically learn a game of moderate difficulty. Besides the complete system design, underlying principles are introduced and expectations are discussed. It is firmly estimated that the algorithm might not only give way to a truly automatic solution of game situations, but also have the potential to offer a starting point for automated problem solution systems outside toy domain.

Keywords: Artificial Intelligence, Artificial Neural Networks, Evolutionary Computing, Connect Four

1. Introduction

Taking the human game play as a basis, many symbolic approaches fully master the most different game systems, i.e. computer players easily arrive at a level of a simple human player and beyond. These systems, however, in most cases take a reasonable amount of planning and development. It is necessary for humans to mentally master the respective situation beforehand and fill in the systematic knowledge and principles to a machine (for some examples see [Russell and Norvig, 2003] or [Nilsson, 1998]).

Connectionist approaches came up as a biologically more plausible form of approaching some problems and a way of finding predictions in yet unknown and untreated environments. However, error-driven learning has the obvious need for a reasonable training set and cannot easily be applied to completely unknown problems (see [Haykin, 2001]).

The current system, Neurogenetic Connect Four, shall combine the power of Recurrent Artificial Neural Networks with the automation of evolution in order to automatically find the way to the problem's solution, merely guided by the lines of success and failure.

2. The Game and Its Tactical Aspects

Connect Four is a board game, reasonably known throughout several countries of the world. Two players, with 21 pieces each, shall establish lines of four pieces on a board of 7 (x-axis) times 6 (y-axis) positions. These lines may be formed in vertical, horizontal and both diagonal directions (ascending and descending). A special feature of Connect Four is, however, the fact that the board itself is vertical and pieces are "piled up" instead of being freely positioned. This produces the fact that there are at maximum 7 possible positions, where each player might place his/her piece.

Though being a fairly simple game by its rules, two tactical aspects have to be considered in particular: two open lines and forced open lines (for further details, please refer to Schneider and Rosa, 2002).

3. Related Work

Game systems are among the most classical solutions developed by Artificial Intelligence methods and systems, which successfully play Connect Four a reasonably widespread. However, most approaches are symbolic rather than connectionist or evolutionary (see, e.g., [Allis, 1988] or [Huang, 2001]).

The present system shall be understood as a more developed version of the system Neural Connect 4, introduced in [Schneider and Rosa, 2002]. Neural Connect 4 is a connectionist approach to the game of Connect Four, using a simple Multilayer Perceptron as the main controller. This Perceptron receives the board as input and provides the best move in the current situation at the output (giving points for each possible row). Its learning takes place by the observation of a small number of high quality games, which introduce the tactical concepts (section 2). Several tests were made using different numbers of hidden neurons. As an overall result it may be said that the system, though simple and straight-forward, has proven that a connectionist approach is an apt way of treating the problem.

As a second interesting approach, an evolutionary system is presented by [Curran and O'Riordan, 2004]. An artificial life simulator is developed with a population of artificial neural networks. Each member makes use of a phenotype (neural network) and genotype (a specific gene code). Reproduction is done by a selection of individuals, which recreate, crossing their genotypes and mapping it to the phenotypes. Inside each generation the fittest elements are determined and taken as teachers for the next generation. This brings about the necessity of interaction between agents. Learning is done via Backpropagation. NGC4 offers a simpler and more transparent approach, which shall arrive at least at the same quality of results.

4. System Design

This section shall summarize the system design of Neurogenetic Connect Four. The system is currently being programmed according to specification and initial tests have been executed.

4.1 Population

NGC4 makes use of a population of between 10 and 100 individuals, each of which possess a standard Recurrent Artificial Neural Network as control instance. All individuals are players and participate in a random tournament during the evolution phase. For every game two individuals are randomly chosen. They play up to 10 games. The winner receives a positive point at every time, the loser a negative point. After a configurable limit of between 1 and 20 negative points, the respective individual dies and is substituted by an individual generated through a mix of the two fittest individuals during tournament until that instance.

4.2 General Evolution Principles

Instead of separating genotype and phenotype as given in [Curran and O'Riordan, 2004], the authors understand that for reasons of simplification and transparency evolution might work directly upon the neural networks, given that certain rules are followed in order to produce reasonable results. Thus, the problem was mainly treated from the point of effectiveness. One of the main principles, which had to be obeyed, was to produce only slight adjustments of the networks in order to prevent population, which might go dumber (winning by chance) instead of wiser. Generally speaking, evolution was to substitute learning from a connectionist point of view.

4.3 Neural Network Architecture

NGC4 uses a standard Recurrent Artificial Neural network with a sigmoid activation function. This network architecture was chosen, in order to better fit the situation, in which sequences are treated and not basic input-output patterns (see [Schneider and Rosa, 2005]). This way, NGC4 offers a considerable advantage over Neural Connect 4 (as in [Schneider and Rosa, 2002]) speaking of appropriate network architectures.

The input layer, consisting of 84 inputs, receives a codification of the network. As three values must be treated (free, black and white), it is necessary to have two bits for mapping, meaning that values are translated: free = 0 0, black = 0 1, white = 1 1. Depending on the number of hidden neurons, consequently a layer of up to 8400 synapses follows, i.e. inputs and hidden neurons are completely interconnected. The feedforward signal then passes through the hidden neuron layer, consisting of a configurable size of 2 to 100 neurons. This variable size was chosen in order to determine the best number of neurons in an experimental way, as also shown in [Schneider and Rosa, 2003] or [Schneider and Rosa, 2005]. The hidden layer is completely interconnected with the output layer with a number of up to 700 synapses. The outputs themselves provide decimal values for each of the possible moves. The one with the highest value shall be chosen. In case there are two or more equal maximum values, the algorithm shall choose randomly between them.

In the recurrent cycle of the network these outputs are fully reconnected to the hidden layer by up to 700 synapses. At the inputs of the hidden layer a mix happens between the recurrent and the feedforward signal. This mix can be configured in percentages in order to determine the best balance between the two signals.

4.4 Evolutional Learning

Although, quite remarkably the chosen network architecture has proven to work well with biologically more plausible learning algorithms such as GeneRec [O'Reilly, 1996], learning in NGC4 takes place merely by evolution, i.e., by the creation of new members of the population. This way, the network configuration of every individual is held constant during its short life time.

This brings about the obvious problem that the mix of individuals from a fix population has tight boundaries of possible resultant configurations. Thus, learning is done by several procreation mixes between two individuals, but also a variable amount of mutation in order to introduce new aspects into the population and avoid a population producing uniformity or an easy way to determine maximum circle of results.

The following algorithms of mix were defined:

- Small changes mix: This is the main reproduction algorithm as it produces the most reasonable results. One of the individuals is taken as initial configuration. The difference between this initial configuration and the other partner of reproduction is determined and a small percentage of that difference (less than 1%) is added to the initial configuration. The evident flaw is that the influence of one parent is very small compared to the other.
- Average mix: The average of the two parents is applied to all values. This produces heavy changes when the parents are very far apart from each other and smooth changes when parents are close to each other. Positive point is the influence of both. A disadvantage is its certain unpredictability and the fact that brand new results are not produced.
- Synchronous mix: Every two values are mixed, i.e. in two values one is determined completely by the father and one completely by the mother. This surely produces potentially surprising results and is moreover meant to shuffle results than to be the dominant reproduction algorithm.

- Formula mix: Applied in a very low percentage, this mix shall create desired random effects apart from classic mutation. In this mix, random arithmetical operations (addition, subtraction, multiplication and division) between the parent values will form the values of the child. To avoid obvious problems, limits are set and divisions by zero sorted out.
- Replication: Only replicates one of the parents without any mix of values. In a small percentage it also shall be applied in order to thoroughly shuffle results.

Just as in natural evolution, artificial evolution cannot be truly fruitful without mutation, as this element – by introducing random features – is the true motor of evolution, because it produces changes coming from outside the universe of the population.

In NGC4 the following methods of mutation are implemented:

- Low percentage change: Adjusts values according to a very small percentage (less than 0.01%), adding the result to the current synapse value. This shall not be done with more than 5% of all values and is one of the most used forms of mutation in NGC4.
- Low absolute change: Adds randomly values between 0.0001 and -0.0001 to not more than 5% of the synapses. This algorithm is the second most used in NGC4.
- Random initialization change: Regardless of the initial value this algorithm puts the synapse value (not more than 1% of synapses) to an initial universe in the range of 0.01 and -0.01. This algorithm is used for not more than 5% of all cases due to its radical value change.

This framework offers the possibility of creating reasonable and new network results in NGC4 without the necessity to define genotypes. This may not configure a classical approach, but, as said beforehand, NGC4 aims to be a transparent and efficient, rather than a standard solution.

4.5 Natural Death

In order to avoid a stagnation of the population, NGC4 offers the possibility to configure a form of “natural death”, i.e. a number of maximum games that a player may survive, regardless of losing or winning the games. This number of games can be held at a reasonable number of around 100 games, in order not to bother normal and desired evolution issues. Future tests shall show the ideal number of iterations.

4.6 Symbolic Teachers

It is expected that the system creates its own way to good players by evolution. However, in order to have a mark of comparison and avoid unnecessary high numbers of evolution steps, NGC4 offers the possibility to hold a certain number of symbolic players in the population. These players shall not die (or get negative points) themselves if they lose, but make other members receive negative points in case they win. Thus they may direction evolution and evaluate evolution quality until a certain point.

The following algorithms were implemented according to [Schneider and Rosa, 2002]:

- “Naive 4”: Simplest algorithm, which plays randomly until it finds an open line of 3 pieces, which it closes in order to win. It also defends against open lines of 3 pieces placed by the opponent. Although being a truly “naive” algorithm, it actually cannot be won by simple maneuvers, i.e. the above mentioned tactical knowledge must be applied to win. Doing this, however, it is quite simple to win as the algorithm does not prevent any opponents’ constructions beforehand nor does itself think about constructing lines.
- “Constructor/Destroyer”: In short “C/D”, this algorithm works by constructing own lines of 4, 3 and 2 pieces and destroying lines of the opponent. It does not consider the tactical knowledge given above, but may be a reasonable challenge for a novice human player, especially speaking of diagonal lines, which may be difficult do decipher at first.
- “Constructor/Destroyer +”: “C/D +” has the same function of “C/D”, but also takes the tactical knowledge of “two open lines” and “forced open lines” (see above) into account. It is a good player and firm mark of comparison for any evolutionary algorithm. “C/D+” might well be considered anything needed to actually successfully treat the “problem” of playing the game of Connect Four, however, the focus of NGC4 is not playing the game, strictly speaking, but demonstrating that an evolutional algorithm paired with a simple Artificial Neural Network (ANN) can also be employed and might thus be used for more complex situations to automatically find the solution.

5. Biological Plausibility

Apart from the fact that NGC4 is a neurogenetic system, it may be seen also as a classic Artificial Life system with a limited universe of multiple agents. In the sense of [Langton, 1995] it represents “life-as-it-could-be”. Possibly, the agents of NGC4 can be compared to very simple forms of life, where agents are already born with apparently all necessary knowledge, merely reacting to the current situation. In a synthetic domain as a Connect Four tournament it may be inappropriate to speak of biological realism. However, if the result is compared to a colony or group of small animals, the result could be considered reasonably close, especially because issues like perception, reaction and natural selection are treated.

In a wider sense, the system fulfills even 5 of the 6 criteria of O’Reilly (1998) for biologically plausible neural networks, with special emphasis on bidirectional activation propagation (recurrent network) and unsupervised learning (through evolution), but also inhibitory learning is given indirectly by the evolutionary elimination process as well as biological realism (see above) and a distributed representation through the ANN.

For future versions of NGC4, an even closer relation to biological plausibility may be considered in order to provide a larger basis for more sophisticated systems. This could go along with the principles of Artificial Evolution (more specifically the evolution of Artificial Life systems), mentioned in [Schneider and Rosa, 2005].

6. Discussion

The system is currently in its programming phase and a number of results should be available shortly. In accordance with initial tests, it is definitely expected that the system finds its own path to a good way of game play. It is especially interesting in this case that only the framework itself has been defined and the rest of the problem solution is truly automatic.

The evolutionary algorithm of NGC4 should at least arrive at the level of the best symbolic algorithm employed. In case this symbolic algorithm has several flaws, evolution should be able to detect these problems. To the authors' expectation this form of neurogenetic framework might be expanded to more difficult problems and might not only offer solutions by itself, but also offer the possibility to point out errors in existing approaches.

Especially interesting about the system's self-organization is the fact that it should bring about completely unexpected results and insights (like emergence according to [Johnson, 2002] or different evolved player characteristics as mentioned in [Bornhofen and Lattaud, 2005]), players with astonishing behavior and characteristics and thus arrive at points, where researchers might not really arrive for the specific domain because of an obvious limit of insight and ideas.

7. Conclusion

NGC4 is a neurogenetic approach to the solution of the game Connect Four, currently being programmed by the authors. This article presented a complete summary of the system design and discussed aspects of biological plausibility as well as the advantages of the neurogenetic framework.

The system is expected to find surprising solutions to the game Connect Four and it is suggested that a modification and expansion of the neurogenetic principle is able to revolutionize and automate solutions in many different areas of research.

References

ALLIS, V. (1988) "A Knowledge-based Approach of Connect-Four", Report IR-163, Faculty of Mathematics and Computer Science, Vrije Universiteit Amsterdam, The Netherlands

BORNHOFEN S. AND LATTAUD C. (2005) "Outlines of Artificial Life: A Brief History of Evolutionary Individual Based Models", In: Proceedings of the 7th International Conference on Artificial Evolution, Lille, France

CURRAN, D. AND O'RIORDAN, C. (2004) "Evolving Connect-Four Playing Neural Networks Using Cultural Learning", Technical Report NUIG-IT-081204, National University of Ireland, Galway, Ireland

HAYKIN, S. (2001) "Neural Networks: A Comprehensive Foundation", Prentice Hall,

HUANG, T. (2001) "Strategy Game Programming Projects", In: *Journal of Computing Sciences in Colleges*, Vol. 16, No. 4, pp. 205-213

JOHNSON, S. (2002) "Emergence: The Connected Lives of Ants, Brains, Cities and Softwares", Simon & Schuster

LANGTON, C. G. (1995) "Artificial Life: An Overview", MIT Press, Cambridge, Massachusetts, USA

NILSSON, N. J. (1998) "Artificial Intelligence: A New Synthesis", Elsevier Science & Technology Books

NORVIG, P. AND RUSSELL, S. (2003) "Artificial Intelligence: A Modern Approach", Prentice Hall

O'REILLY, R. C. (1996) "Biologically Plausible Error-Driven Learning using Local Activation Differences: The Generalized Recirculation Algorithm, In: *Neural Computation*, pp. 895-938

O'REILLY, R. C. (1998) "Six Principles for Biologically-Based Computational Models for Cortical Cognition", In: *Trends in Cognitive Science*, vol. 2, pp. 455-462

SCHNEIDER, M. AND ROSA, J. L. G. (2002) "Neural Connect 4 – A Connectionist Approach to the Game", In: Proceedings of the VIIth Brazilian Symposium on Neural Networks (SBRN'2002), IEEE Computer Society Press, Washington, Brussels, Tokyo, pp. 236-241

SCHNEIDER, M. AND ROSA, J. L. G. (2003) "Neural Labyrinth Robot - Finding the Best Way in a Connectionist Fashion", IV Encontro Nacional de Inteligência Artificial (ENIA'03), Campinas, Brazil, 2003, pp. 1751-1760

SCHNEIDER, M. AND ROSA, J. L. G. (2005) "BioAnt – Biologically Plausible Computer Simulation of an Environment with Ants", In: Proceedings of the International Joint Conference on Neural Networks (IJCNN'2005), Montreal, Canada, pp. 1505-1510