Evolving Agents to Play Connect–Four Using Cultural Learning

Dara Curran, Colm O'Riordan

Dept. of Information Technology
National University of Ireland, Galway.

Abstract. This paper describes a cultural learning approach to the evolution of agents to play the game of connect–four. Each agent has a neural network responsible for perceiving the current board configuration and selecting an appropriate next move. Populations evolve through population learning, a process of Darwinian evolution, using genetic algorithms. Cultural learning is implemented by selecting highly fit agents as teachers to instruct the next generation. Teachers communicate with pupils through a hidden layer in each neural network (the verbal input/output layer) and pupils attempt to replicate utterances by back–propagation. Experiments are conducted comparing the performance of populations employing population learning alone and populations employing both population and cultural learning.

1 Introduction

Cultural learning allows populations to pass on knowledge to the next generation through non–genetic means through a process of communication or artifact creation. Populations employing such a mechanism should intuitively be inherently more robust to changing and hostile environments. From an artificial intelligence perspective, cultural learning is useful because it provides an interesting alternative to more traditional life–time learning simulations.

In order to simulate cultural learning, teachers are selected from the population and allowed to instruct the next generation. In this way, important information gained through the life-time of the previous generation is not lost and the fitness of the entire population can be improved. More importantly, no prior solution knowledge is required to produce solutions, making cultural learning an ideal candidate for sequential decision task solving and an alternative to reinforcement learning or test–case approaches.

In this paper, we chose the game of connect–four as the sequential decision task to be solved. While a simple game, connect–four requires the player to develop a clear strategy to be consistently successful. In order to ascertain the benefit of cultural learning, an initial set of experiments are performed with populations employing only Darwinian–based population learning through the use of genetic algorithms. A second experiment adds cultural learning to the population.

Some research has been undertaken in the evolution of connect–four players employing a library of existing games to train the neural networks by back–propagation [1] and reinforcement learning methods [2]. This paper presents an alternative approach to both.

The remainder of this paper is arranged as follows: Section 2 summarises related research and background material. Section 3 discusses the encoding technique employed to allow neural network agents to play. Section 4 presents the artificial life simulator employed to conduct the experiments. Section 5 illustrates the experiment results. Section 6 concludes and suggests future work.

2 Related Work

The following section outlines some background material including learning models and the game of connect–four.

2.1 Learning Models

A number of learning models can be identified from observation in nature. These can roughly be classified into population, life-time and cultural learning.

Population Learning Population learning refers to the process whereby a population of organisms evolves, or learns, by genetic means through a Darwinian process of iterated selection and reproduction of fit individuals. In this model, the learning process is strictly confined to each organism's genetic material: the organism itself does not contribute to its survival through any learning or adaptation process.

Life–time Learning By contrast, there exist species in nature that are capable of learning, or adapting to environmental changes and novel situations at an individual level. Such learning, known as life–time learning is often coupled with population–based learning, but further enhances the population's fitness through its adaptability and resistance to change. Another phenomenon related to life-time learning, first reported by Baldwin [3], occurs when certain behaviour first evolved through life-time learning becomes imprinted onto an individual's genetic material through the evolutionary processes of crossover and mutation. This individual is born with an innate knowledge of such behaviour and, unlike the rest of the population, does not require time to acquire it through life-time learning. As a result, the individual's fitness will generally be higher than that of the population and the genetic change should become more widespread as the individual is repeatedly selected for reproduction.

Research has shown that the addition of life-time learning to a population of agents is capable of achieving much higher levels of population fitness than population learning alone [4–7].

Cultural Learning Culture can be succinctly described as a process of information transfer within a population that occurs without the use of genetic material. Culture can take many forms such as language, signals or artifactual materials. Such information exchange occurs during the lifetime of individuals in a population and can greatly enhance the behaviour of such species. Because these exchanges occur during an individual's lifetime, cultural learning can be considered a subset of lifetime learning.

Experiments conducted by Hutchins and Hazlehurst [8] simulate cultural evolution through the use of a hidden layer within an individual neural network in the population. The hidden layer acts as a verbal input/output layer and performs the task of feature extraction used to distinguish different physical inputs. It is responsible for both the perception and production of signals for the agent.

A number of approaches were considered for the implementation of cultural learning including fixed lexicons [10, 11], indexed memory [12], cultural artifacts [13, 14] and signal–situation tables [15]. The approach chosen was the teacher/pupil scenario [16, 17, 11] where a number of highly fit agents are selected from the population to act as teachers for the next generation of agents. Pupils learn from teachers by observing the teacher's verbal output and attempting to mimic it using their own verbal apparatus. As a result of these interactions, a lexicon of symbols evolves to describe situations within the population's environment.

2.2 Connect Four

The game of connect–four is a two–player game played on a vertical board of 7x6 positions into which pieces are slotted in one of seven available slots. Each player is given a number of coloured pieces (one colour per player) and must attempt to create horizontal, vertical or diagonal piece–lines of length four. Players place one piece per turn into one of the seven slots. The piece then falls onto a free position in the chosen column, creating piles, or towers, of pieces. If a column is full, the player must select an available slot.

While the game appears simple, a certain amount of tactical knowledge is required to play proficiently. The most obvious approach is to scan the board for existing lines of three and either finish them to create four–in–a–line, or if the line is the opponent's, block it. However, as is the case in many games, the best approaches focus on forcing the opponent to contribute to the player's victory. The two most popular techniques are outlined below.

Open Lines The basic premise of the open lines strategy, illustrated in the left–hand board in Fig. 1, is to create a situation where a win is inevitable regardless of any opponent's move. A player must create a line of 3 pieces with space available on *both* sides of the line. Since the opponent can only move one piece at a time, this situation will always lead to player victory.

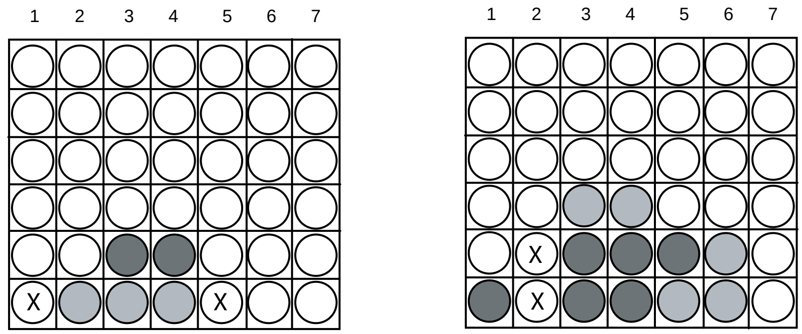

Fig. 1. *Connect–Four Strategies*

Forced Open Lines The forced open lines strategy follow the same basic premise as the open lines strategy, but actively forces the opponent into conceding victory. This is achieved by placing pieces in such a manner that the opponent must move into a column in order to prevent the player winning. Once the opponent's piece is in place, its position allows the player to complete a different winning line. This is illustrated in the right–hand board in Fig. 1.

3 Game Encoding

In order for a population of neural networks to play games of connect–four, a method must be developed to encode both the board's current position and decode the network's output into a valid move.

3.1 Board Encoding

As the game of connect–four consists of a simple matrix with only two types of piece (the player's and the opponent's), encoding the current board position is not difficult. Two bits are used for each position on the board, where bit patterns are chosen arbitrarily as: 00 for empty positions, 01 for player pieces and 11 for opponent pieces. Each board position is encoded sequentially, creating a bit string of 84.

3.2 Move Decoding

Move decoding takes the network's output and determine's the network's chosen move. During the preparation for these experiments, two decoding schemes were considered.

Multiple Move Selection For the multiple move selection approach, an agent's neural network was allowed seven output nodes, each representing a possible move. The network is shown the current board position and the strongest output representing a valid move is chosen as the network's output.

Multiple Board Selection The second approach changes the situation from a choice of moves to a choice of board positions. The current board position is taken and the agent's pieces are added iteratively into each slot. At each iteration, the network is shown the current board position, plus one of the seven possible moves. This time, the neural network has only one output node and the board position with the strongest output response is deemed to be the agent's preferred board position.

4 Simulator

The experiments outlined in this paper were performed using a previously developed artificial life simulator[18, 6, 7]. The simulator allows populations of neural networks to evolve using a genetic algorithm and each network can also be trained during each generation of an experiment to simulate life–time learning.

Each member of the population is in possession of both a phenotype (a neural network) and a genotype (a gene code). The gene code is used to determine the individual's neural network structure and weights at birth. If the individual is selected for reproduction, the gene code is combined with that of another individual using the process of crossover and mutation to produce a genotype incorporating features from both parents.

In order for this mechanism to function correctly, a mapping of a neural network structure to a gene code is required. This is achieved using a modified version of marker based encoding which allows networks to develop any number of nodes and interconnecting links, giving a large number of possible neural network architecture permutations.

Marker based encoding represents neural network elements (nodes and links) in a binary string. Each element is separated by a marker to allow the decoding mechanism to distinguish between the different types of element and therefore deduce interconnections [19–21].

In this implementation, a marker is given for every node in a network. Following the node marker, the node's details are stored in sequential order on the bit string. This includes the node's label and its threshold value. Immediately following the node's details, is another marker which indicates the start of one or more node–weight pairs. Each of these pairs indicates a back connection from the node to other nodes in the network along with the connection's weight value. Once the last connection has been encoded, the scheme places an end marker to indicate the end of the node's encoding (Fig. 3). This scheme allows any number of hidden layers and nodes, giving great flexibility for experimentation.

The networks undergo various stages throughout their lifetime. Firstly, the gene codes are decoded to create their neural network structure. Training is then performed using error back–propagation for a given number of iterations (training cycles). Each network is tested to determine its fitness and the population is ranked using linear based fitness ranking. Roulette wheel selection is employed to generate the intermediate population. Crossover and mutation operators are then applied to create the next generation.

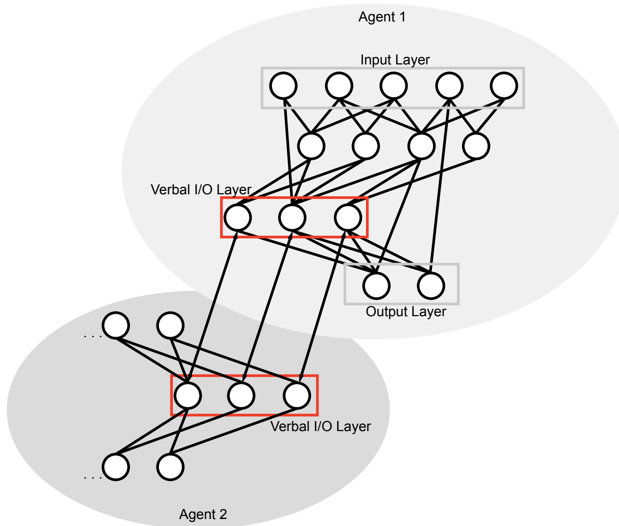

Fig. 2. *Agent Communication Architecture*

4.1 Simulating Cultural Evolution

In order to perform experiments related to cultural evolution, it was necessary to adapt the existing simulator architecture to allow agents to communicate with one another. This was implemented using an extended version of the approach adopted by Hutchins and Hazlehurst. The last hidden layer of each agent's neural network functions as a verbal input/output layer (Fig. 2).

	Start Marker	Node Label	Threshold	Link to Node	Link Weight	Link to Node	Link Weight	End Marker	
...	SM	5	0.8	4	0.83	3	-0.51	EM	...

Fig. 3. *Marker Based Encoding*

At the end of each generation, a percentage of the population's fittest networks are selected and are allowed to become teachers for the next generation. The teaching process takes place as follows: a teacher is stochastically assigned n pupils from the population where $n = \frac{N_{pop}}{N_{teachers}}$, where N_{pop} is the population size and $N_{teachers}$ is the number of teachers. Each pupil follows the teacher in its environment and observes the teacher's verbal output as it plays games of connect–four. The pupil then attempts to emulate its teacher's verbal output using back-propagation. Once the teaching process has been completed, the teacher networks die and new teachers are selected from the new generation.

5 Experiment Results

A population of 20 agents were allowed to evolve for 100 generations. At each generation, agents play in a tournament against all other players. In addition, each agent plays a minimax player with three levels of difficulty. In total, each agent plays 22 games of connect–four in its lifetime. Agents are assigned fitness according to each game's result: 3 points for a both a win and a draw and 0 points for a loss. This gives a fitness range of [0,66].

Crossover was set at 0.6 and mutation at 0.02. The cultural learning settings of teacher ratio and teaching cycles were set at 0.1 and 5 respectively. An additional parameter, cultural mutation, was also added. This generates noise in the range [-0.5,0.5] to the teacher's output when instructing a pupil with probability 0.05 and was found empirically to improve the performance of cultural learning.

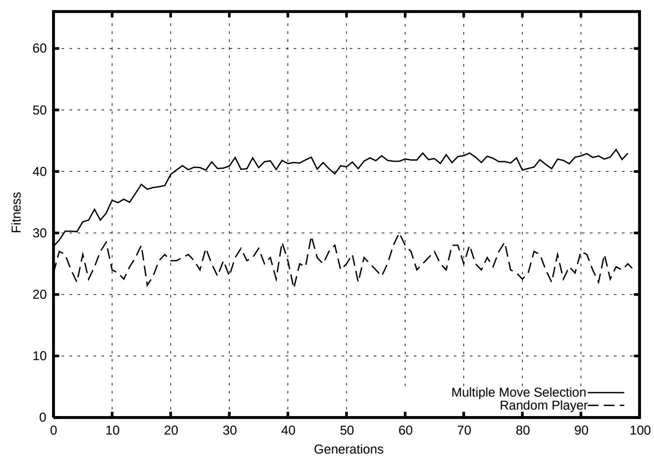

Fig. 4. *Multiple Move Selection*

The first two experiments employ population learning alone and are designed to determine the best move–decoding approach as well as the performance of population learning. The final experiment adds cultural learning to the population. For the purposes of comparison, the performance of a player employing a random strategy is illustrated along with the simulation results.

5.1 Multiple Move Selection

It is clear from the results illustrated in Fig. 4 that the agents are indeed evolving to play better games of connect–four but that this evolution quickly stabilises at fitness levels of around 42. The population appears to have stagnated and further improvement is unlikely. However, the population's performance far outstrips that of the random player.

5.2 Multiple Board Selection

The second decoding strategy was more successful. Fig. 5 shows both the multiple move and multiple board selection results to better illustrate the improvement. The population have attained a higher level of fitness than the previous method. In light of these results, we employed multiple board selection to conduct the final experiment involving the addition of cultural learning to the population.

5.3 Cultural Learning

This final experiment adds cultural learning capabilities to the population. Teachers play a full tournament while pupils observe, and at each move, the teacher corrects the pupil's perception through back–propagation. The results in Fig. 6 show that the addition of cultural learning provides the best performance and that the fitness levels show an upward trend at the end of the experiment, suggesting that the population is capable of further improvement.

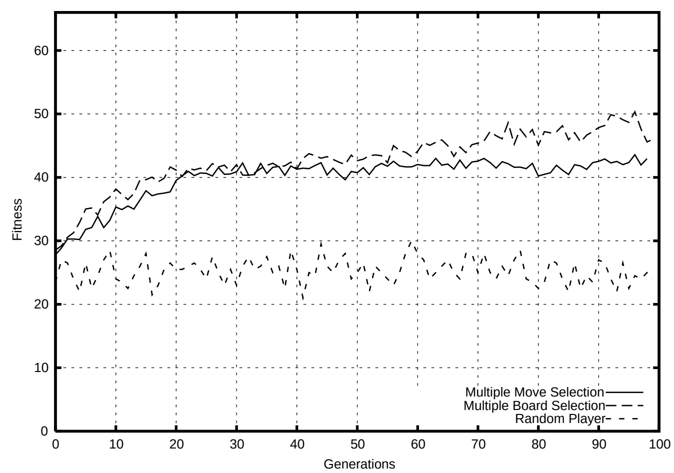

Fig. 5. *Multiple Board Selection*

6 Conclusion

These experiments show that cultural learning can be successfully applied in populations of neural networks to develop strategies for the game of connect–four. In addition, the preliminary experiments show that multiple board selection delivers better results than multiple move selection. Since neural networks are efficient classifiers it is intuitively preferable to present a selection of board configurations to rank. Furthermore, the type of output required from multiple move

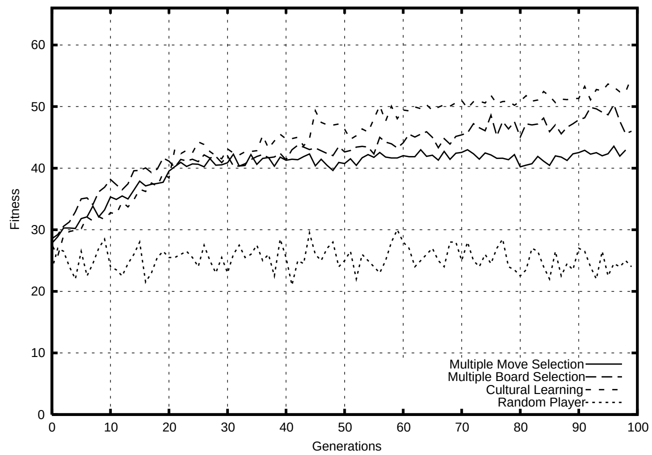

Fig. 6. *Cultural Learning*

selection is very disparate and difficult for the network to learn efficiently: an output of 000001 (a move in the seventh slot) is markedly different from 100000 (a move in the first slot).

Future work will concentrate on larger populations, larger simulations, varying the cultural learning parameters and more complex problems.

References

1. Marvin Oliver Schneider and Joo Lus Garcia Rosa. Neural connect four, a connectionist approach to the game, 2002.
2. Peer Sommerlund. Artificial neural networks applied to strategic games, 1996.
3. J.M. Baldwin. A new factor in evolution. In *American Naturalist 30*, pages 441–451, 1896.
4. F. B. Pereira and E. Costa. How learning improves the performance of evolutionary agents: A case study with an information retrieval system for a distributed environment. In *Proceedings of the International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Models (ISAS 2001)*, pages 19–23, 2001.
5. James Watson and Janet Wiles. The rise and fall of learning: A neural network model of the genetic assimilation of acquired traits. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, pages 600–605, 2002.
6. D. Curran and C. O'Riordan. On the design of an artificial life simulator. In V. Palade, R. J. Howlett, and L. C. Jain, editors, *Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2003)*, University of Oxford, United Kingdom, 2003.
7. D. Curran and C. O'Riordan. Artificial life simulation using marker based encoding. In *Proceedings of the 2003 International Conference on Artificial Intelligence (IC-AI 2003)*, volume II, pages 665–668, Las Vegas, Nevada, USA, 2003.

8. E. Hutchins and B. Hazlehurst. How to invent a lexicon: The development of shared symbols in interaction. In N. Gilbert and R. Conte, editors, *Artificial Societies: The Computer Simulation of Social Life*, pages 157–189. UCL Press: London, 1995.
9. N. Chomsky. On the nature of language. In *Origins and evolution of language and speech*, pages 46–57. Annals of the New York Academy of Science, New York. Vol 280, 1976.
10. H. Yanco and L. Stein. An adaptive communication protocol for cooperating mobile robots, 1993.
11. A. Cangelosi and D. Parisi. The emergence of a language in an evolving population of neural networks. *Technical Report NSAL–96004, National Research Council, Rome*, 1996.
12. L. Spector and S. Luke. Culture enhances the evolvability of cognition. In *Cognitive Science (CogSci) 1996 Conference Proceedings*, 1996.
13. E. Hutchins and B. Hazlehurst. Learning in the cultural process. In *Artificial Life II, ed. C. Langton et al.* MIT Press, 1991.
14. A. Cangelosi. Evolution of communication using combination of grounded symbols in populations of neural networks. In *Proceedings of IJCNN99 International Joint Conference on Neural Networks (vol. 6)*, pages 4365–4368, Washington, DC, 1999. IEEE Press.
15. B. MacLennan and G. Burghardt. Synthetic ethology and the evolution of cooperative communication. In *Adaptive Behavior 2(2)*, pages 161–188, 1993.
16. A. Billard and G. Hayes. Learning to communicate through imitation in autonomous robots. In *7th International Conference on Artificial Neural Networks*, pages 763–738, 1997.
17. D. Denaro and D. Parisi. Cultural evolution in a population of neural networks. In *M.Marinaro and R.Tagliaferri (eds), Neural Nets Wirn-96.New York: Springer*, pages 100–111, 1996.
18. D. Curran and C. O'Riordan. Learning in artificial life societies. In *Report number nuig-it-220202. Technical Report, Dept. of IT.* NUI, Galway, 2002.
19. K. O. Stanley and R. Miikkulainen. Efficient evolution of neural network topologies, 2002.
20. H. Kitano. Designing neural networks using genetic algorithm with graph generation system. In *Complex Systems, 4, 461-476*, 1990.
21. G. F. Miller, P. M. Todd, and S. U. Hedge. Designing neural networks using genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, pages 379–384, 1989.