Cultural Evolution for Sequential Decision Tasks: Evolving Tic–Tac–Toe Players in Multi–Agent Systems

Dara Curran and Colm O'Riordan

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

Abstract. Sequential decision tasks represent a difficult class of problem where perfect solutions are often not available in advance. This paper presents a set of experiments involving populations of agents that evolve to play games of tic–tac–toe. The focus of the paper is to propose that cultural learning, i.e. the passing of information from one generation to the next by non–genetic means, is a better approach than population learning alone, i.e. the purely genetic evolution of agents. Population learning is implemented using genetic algorithms that evolve agents containing a neural network capable of playing games of tic–tac–toe. Cultural learning is introduced by allowing highly fit agents to teach the population, thus improving performance. We show via experimentation that agents employing cultural learning are better suited to solving a sequential decision task (in this case tic–tac–toe) than systems using population learning alone.

1 Introduction

Lifetime learning can take many forms - at its simplest it is a reaction to a particular stimulus and the adjustment of a world view that follows the reaction. Thus, very simple organisms are capable of learning to avoid harmful substances and are attracted to other beneficial ones. In computational terms, lifetime learning can be simulated using neural networks by employing an error reducing algorithm such as error back–propagation.

While this type of lifetime learning has been shown to be useful in the past, it relies on prior solution knowledge: in order to correctly train agents, the system must be aware of the solution to be attained. An alternative approach is the use of cultural learning, a subset of lifetime learning where agents are allowed to communicate information through a hidden verbal layer in each agent's neural network. Teacher agents are selected from the population and are assigned a number of pupils that follow the teacher as it performs its task.

Teachers and pupils exchange information each time a stimulus occurs and the pupils learn using back propagation to imitate the teacher's verbal output and behaviour. Since cultural learning does not require a priori solution knowledge, it is an ideal system for problems where perfect solutions are not available or are non–trivial to discover. Examples of such problems are sequential decision tasks, problems that can only be solved through repeated iterations, such as sorting problems and move–based games.

The focus of this paper is to examine the benefit of combining population and cultural learning over population learning alone for a simple sequential decision task problem: the game of tic–tac–toe. While the problem is not complex, it is of sufficient difficulty to illustrate the potential for cultural learning in domains of this kind.

The remainder of the paper is organised as follows. The next section describes some related work focusing on the types of learning that can be employed by multi–agent systems. Section 3 outlines the experimental setup, describing the artificial life simulator employed, the cultural learning framework and how these were adapted in order to learn the game of tic–tac–toe. Section 4 presents results of experiments where population learning was used to evolve tic–tac–toe players and where cultural learning was added. Section 5 concludes the paper and suggests avenues of future research.

2 Related Work

2.1 Evolving Game–Playing Agents

Many researchers have developed evolutionary techniques to generate game–playing agents for a variety of games [1–4]. However, little research to date has focused on the addition of cultural learning to such tasks. We feel that the game of tic–tac–toe, while a simple game, represents a good starting point for researching cultural learning in a game–playing domain.

2.2 Learning Models

A number of learning models can be identified from observation of nature. These can roughly be classified into two distinct groups: population and life-time learning. In this paper we consider another form of lifetime learning, 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.

Lifetime 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, know as life-time learning, still employs population learning to a degree, 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 [5], occurs when certain behaviours, first evolved through life-time learning, become 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 mutation 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 [6, 7]. Furthermore, population learning alone is not well suited to changing environment [8].

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.

Using genetic algorithms, the evolutionary approach inspired by Darwinian evolution, and the computing capacity of neural networks, artificial intelligence researchers have been able to achieve very interesting results.

Experiments conducted by Hutchins and Hazlehurst [9] simulate cultural evolution through the use of a hidden layer within an individual neural network in the population. This in effect, simulates the presence of a Language Acquisition Device (LAD), the physiological component of the brain necessary for language development, whose existence was first suggested by Chomsky [10]. 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 [11, 12], indexed memory [13], cultural artifacts [14, 15] and signal–situation tables [16]. The approach chosen was the increasingly popular teacher/pupil scenario [17, 18, 12] where a number of highly fit agents are selected from the population to act as teachers for the next generation of agents, labelled pupils. 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.

3 Experimental Setup

3.1 Simulator

The experiments outlined in this paper were performed using an artificial life simulator developed by Curran and O'Riordan [19, 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 [20, 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.

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.

3.2 Cultural Learning Framework

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. Their approach uses the last hidden layer of each agent's neural network as a verbal input/output layer (figure 1) and employs a fixed number of verbal input/output nodes. We have modified Hutchins and Hazlehurst's system to allow the number of verbal input/output nodes to evolve with the population, making the system more adaptable to potential changes in environment. In addition, this method does not make any assumptions as to the number of verbal nodes (and thus the complexity of the emerging lexicon) that is required to effectively communicate.

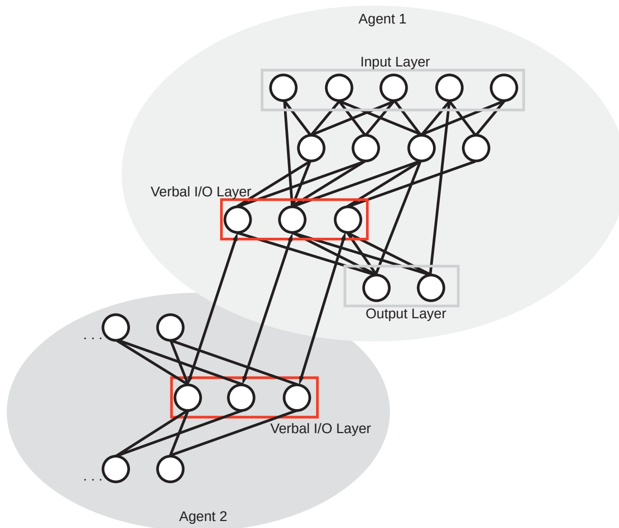

Fig. 1. Agent Communication Architecture

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 interacts with its environment. 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.

3.3 Tic Tac Toe

The sequential decision task chosen for this set of experiments is the game of tic-tac-toe. While this is a very simple game, we believe it serves to illustrate the benefit of cultural evolution for sequential decision tasks and can be used as a stepping stone to more difficult problems.

In order to evolve good players, it was decided that agents in the population would all compete against a perfect player rather than compete against each other. It was felt that populations of agents competing against each other would be likely to converge only to local maxima due to the lack of competitive pressure. To avoid over–fitting, the perfect player employs a modified minimax method to determine moves where the first move of the game is randomized so that agents play a variety of games rather than the same game at each iteration. Each agent plays four games in its lifetime: two where the agent moves first and the other two where the perfect player moves first.

In order to play tic-tac-toe, an agent's neural network structure must follow certain parameters. There are 18 input nodes, 2 for each board position where 01 is X, 10 is O and 11 is an empty square. Nine output nodes corresponding to each board position are used to indicate the agent's desired move where the node with the strongest response corresponding to a valid move is taken as the agent's choice. The simulator allows agents to evolve any number of hidden layers each with an unrestricted number of nodes, giving maximum flexibility to the evolutionary process. During the teaching process, a teacher agent plays alongside the pupil. At each move, both the pupil and teacher choose the next move and the pupil's verbal output is corrected with respect to the teacher's using error back-propagation.

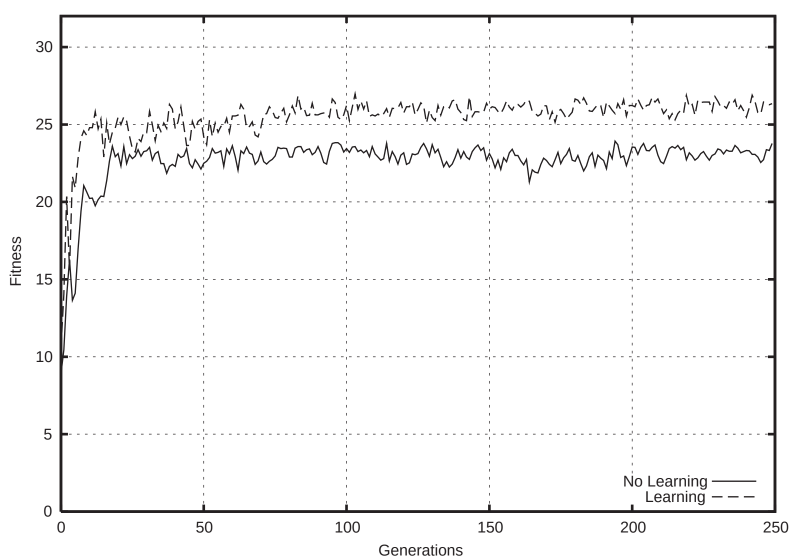

Fig. 2. Agent Fitness

Since the agents play against a perfect player, fitness is assigned according to how long each agent is capable of avoiding a loss situation. An agent's fitness is therefore correlated with the number of moves that each game lasts, rewarding agents capable of forcing the perfect player to as close to a draw as possible. The fitness function produces values in the range [0,32], where 32 is the maximum fitness (the situation where the agent draws all four games).

Populations of 100 agents were generated for these experiments and allowed to evolve for 250 generations. Crossover was set at 0.6 and mutation at 0.02. The teaching rate was set at five cycles and the value for n (the number of individuals selected to become teachers at each generation) was set to 10% of the population. In addition, a teaching mutation rate which modifies a teacher's output when training a pupil was incorporated and set at 0.02. The results presented are an average of 20 experiment runs.

4 Experimental Results

Two experiments were undertaken: one using only population learning to evolve players, and the other using population and cultural learning. Figure 2 shows the average fitness values for the two evolving populations. While both types of learning begin at similar levels of fitness, it is clear that agents employing cultural evolution are performing better as the experiment progresses.

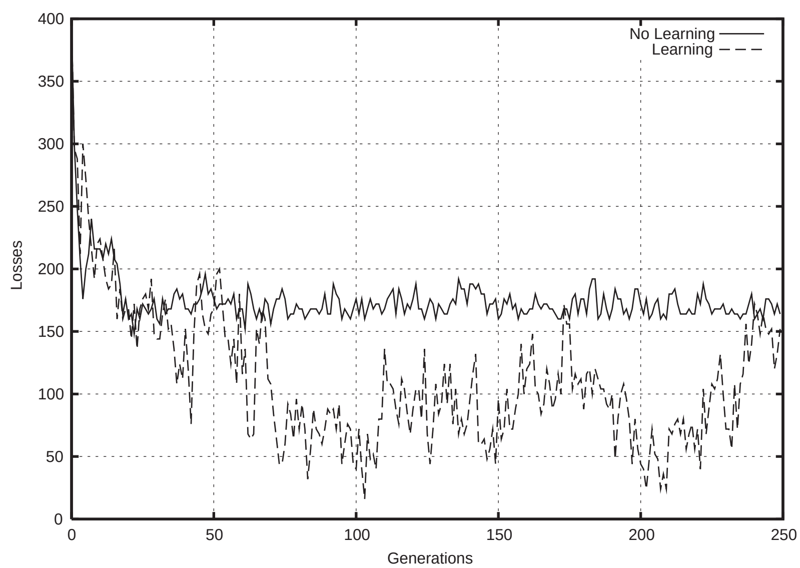

Fig. 3. Number of game losses

The number of game losses over time, illustrated in figure 3, show that while cultural evolution delivers less losses than pure population learning, it does not provide stability. While population learning losses seem to stabilise at above 150 losses, cultural learning losses seem to oscillate considerably, reaching a low of

24 losses and highs of 150. We posit that the inherent noisiness of the cultural learning approach causes agents to behave more erratically – the learning process may bring an output layer node from a dormant state to a sudden forefront position causing a dramatic change in the agent's playing pattern.

It is interesting to compare these results with those obtained by Angeline and Pollack [22] who used a competitive fitness function to evolve populations of neural network tic–tac–toe players. The population of evolving players was pitted against a number of 'expert' player strategies, including a perfect player. If we examine their results in terms of a draws/losses ratio, we find that their best evolved players (playing against a perfect player) obtain a ratio of 0.2405. By contrast, the cultural learning approach presented in this paper obtains highs of 0.94 and lows of 0.625.

5 Conclusions

The results of these experiments suggest that cultural learning is superior to population learning alone for simple sequential decision tasks. Future work will examine the effect of longer teaching cycles, varying teacher/population ratios and more complex tasks.

Acknowledgements

This research is funded by the Irish Research Council for Science, Engineering and Technology. The authors wish to thank the reviewers for their helpful feedback and comments.

References

1. D. Moriarty and R. Miikkulainen. Discovering complex othello strategies through evolutionary neural networks. *Connection Science*, 7(3–4):195–209, 1995.
2. L. Weaver and T. Bossomaier. Evolution of neural networks to play the game of dots-and-boxes. In *Artificial Life V: Poster Presentations*, pages 43–50, 1996.
3. K. Chellapilla and D. B. Fogel. Evolving neural networks to play checkers without relying on expert knowledge. In *IEEE Transactions on Neural Networks*, volume 10, pages 1382–1391, 1999.
4. Norman Richards, David Moriarty, Paul McQuesten, and Risto Miikkulainen. Evolving neural networks to play Go. In *Proceedings of the 7th International Conference on Genetic Algorithms*, East Lansing, MI, 1997.
5. J.M. Baldwin. A new factor in evolution. In *American Naturalist 30*, pages 441–451, 1896.
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. D. Curran and C. O'Riordan. Lifetime learning in multi-agent systems: Robustness in changing environments. In *Proceedings of the 14th Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2003)*, pages 46–50, Dublin, Ireland, 2003.
9. 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.
10. 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.
11. H. Yanco and L. Stein. An adaptive communication protocol for cooperating mobile robots, 1993.
12. C. Angelo and D. Parisi. The emergence of a language in an evolving population of neural networks. *Technical Report NSAL–96004, National Research Council, Rome*, 1996.
13. L. Spector and S. Luke. Culture enhances the evolvability of cognition. In *Cognitive Science (CogSci) 1996 Conference Proceedings*, 1996.
14. E. Hutchins and B. Hazlehurst. Learning in the cultural process. In *Artificial Life II, ed. C. Langton et al.* MIT Press, 1991.
15. 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.
16. B. MacLennan and G. Burghardt. Synthetic ethology and the evolution of cooperative communication. In *Adaptive Behavior 2(2)*, pages 161–188, 1993.
17. 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.
18. 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.
19. 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.
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.
22. P. J. Angeline and J.B. Pollack. Competitive environments evolve better solutions for complex tasks. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 264–270, San Francisco, CA, 1993. Morgan Kaufmann.