1139

Learning and Evolution in Populations of Backprop Networks

B V Williams and D G Bounds

Department of Computer Science & Applied Mathematics
Aston University, Aston Triangle, Birmingham B4 7ET, England

e-mail: williabv@cs.aston.ac.uk

ABSTRACT

This paper reports an investigation of the relationship between learning and evolution in populations of backprop networks. The simulation environment, which has been used previously by Parisi, Nolfi and Cecconi (Parisi, Nolfi and Cecconi, 1991), consists of a two dimensional grid with a random sample of the vertices containing a reward for those animats that visit them. Each animat is a small backprop neural network with a limited lifetime. Initial weights are chosen using a genetic algorithm. Depending on the experiment, these weights may or may not be modified by backpropagation during the animats' lifetimes in response to their performance in the environment. We find no evidence that learning during animats' lifetimes has any beneficial effect on evolution for this particular task. We find that the task is learned more successfully by simple perceptrons than by multi-layer networks, which suggests that it is linearly separable. Finally, we offer an alternative, more simple, explanation for the phenomena observed by Parisi, Nolfi and Cecconi and originally explained by those authors in terms of behavioural self-selection of stimuli.

INTRODUCTION

Parisi, Nolfi and Cecconi recently reported a study of the relationship between learning, behaviour and evolution in a simple two dimensional world populated by animats. Animats consist of small feedforward neural networks which perform a simple food gathering task. Over time, new animats are generated from the more successful members of the animat population using a genetic algorithm, or GA (Holland, 1975; Goldberg, 1989). Parisi, Nolfi and Cecconi found that: "*Learning can accelerate the evolutionary process both (1) when learning tasks correlate with the fitness criterion, and (2) when random learning tasks are used. Furthermore, an ability to learn a task can emerge and be transmitted evolutionarily for both correlated and uncorrelated tasks.*"

It is not obvious why the above statement should hold for very simple artificial neural networks operating in simple problem domains such as the one described. Indeed, if true, it would have major consequences for the design of artificial neural networks for practical applications since it would imply that a network trained on one task should have useful performance even on other uncorrelated tasks. We therefore decided to carry out a more extensive study of this artificial world.

ANIMATS AND THEIR WORLDS

Following Parisi, Nolfi and Cecconi, each animat lives in a separate two dimensional environment containing randomly placed pieces of "food". Initially, both the food items and the animat are randomly located in cells within a 10x10 grid. During its lifetime an animat moves around its world, sometimes landing on a food cell and eating the food. Ultimately, those individuals that are most successful at finding food are more likely to reproduce and their offspring will become increasingly successful at this food gathering task: the fat cats will get fatter.

An animat consists of a feedforward neural network[1] that receives sensory input from the environment, in this case the angle and distance to the nearest cell that contains food, and generates an output action which results in the animat either moving forward by one cell, turning left or right through 90 degrees, or staying still. The neural network architecture, which does not change during an experiment, is shown in figure 1. It has four input units and two output units, fully connected to seven hidden units. Two of the four input units receive the angle and distance to the closest food cell, and the remaining two receive the output values from the network for the most recent action. The output units are thresholded to produce an action choice coded as two binary digits: 00 = halt; 01 = turn right; 10 = turn left; 11 = advance.

In any single experiment, all animats have the same network architecture and neuron functions; they differ only by having different sets of weight values. Initially, these weight values are chosen randomly, but a genetic algorithm is used subsequently to generate new animats.

Each animat's weight values are coded as a chromosome of floating point numbers. Initially, a population of 100 animats is created, each with a random set of connection weights. Each individual is then allowed to live for 20 lives (a life consisting of 50 actions from a random starting point) in 5 different environments (i.e. food placings). All animats are then assessed and the 20 individuals which have eaten the most food are selected as the basis for the next generation. Each of these individuals is reproduced 5 times, and each offspring is subjected to mutation by perturbing 5 weights, selected at random, by a random real value between ±1.0, to produce a population of 100 new animats. This process of evaluation, selection, reproduction

[1] Artificial neural networks are parallel, distributed machine learning architectures originally inspired by, and are loosely based on, biological neural systems. The reader unfamiliar with neural networks is referred to Rummelhart and McClelland (1986).

and mutation represents one generation, and the original experiments were carried out for 50 generations. In the current experiments, runs consisted of 70 generations since it was not clear that a plateau of performance had been reached with fewer generations.

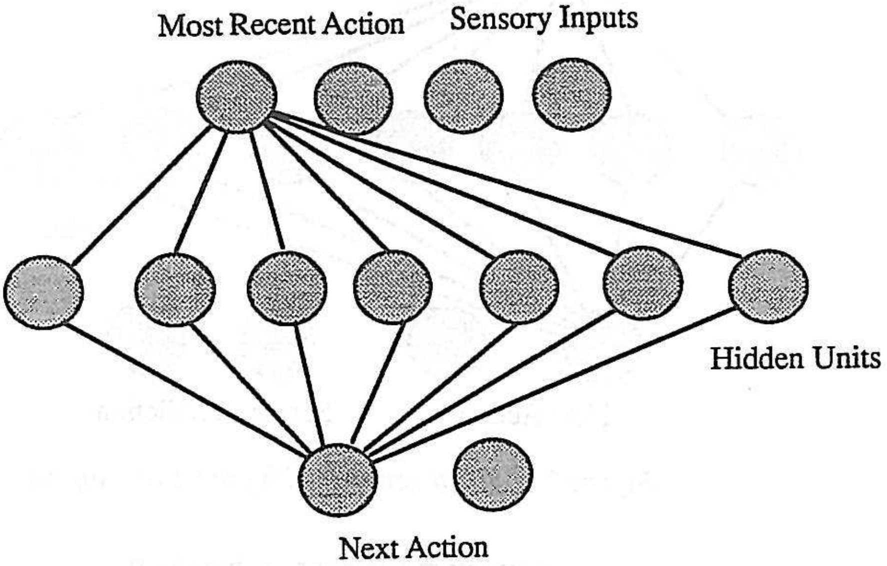


Figure 1. Animat Network

The animats of the first generation exhibit purely random behaviour; because their network weights are random they only consume food by chance. However, by selecting those individuals that have consumed the most food, and introducing slight mutations into the reproduction of selected individuals as described above, the population evolves and individuals in successive generations become more effective at finding food as useful sets of network weights are discovered.

In a second set of experiments, individuals are also allowed to learn during their lifetimes, by adapting their weights using the standard backpropagation method (Rumelhart, Hinton and Williams, 1986). This introduces an interesting possibility. The evolutionary adaptation is purely Darwinian since it is the initial weights at the beginning of an individual's life that are encoded on the chromosome. No modifications made to these weights during the lifetime of an individual are passed on to the next generation directly. However, learning may increase the chance of an individual being selected if it improves its ability to find food. In this second experiment what evolves is not necessarily just the ability to seek food efficiently; it may also be the ability to learn effectively during life.

In order to use backpropagation to train any network, an error signal on the output units at each time step is needed. However, for this task there is a temporal credit assignment problem since the payoff for finding food may occur many steps after any given move. To avoid this problem, Parisi, Nolfi and Cecconi provide an error signal from what they consider to be a related task: predicting the sensory consequences of the animat's most recent action. Two new output units are added to the network (see figure 2) and these units are used to predict what the sensory input will be on the succeeding move. The error signal is the difference between this prediction and the actual sensory input after the move is made. In this way it is possible to generate an error signal at each timestep which can be backpropagated through the network, modifying all weights except those between the hidden units and the two original (action) output units.

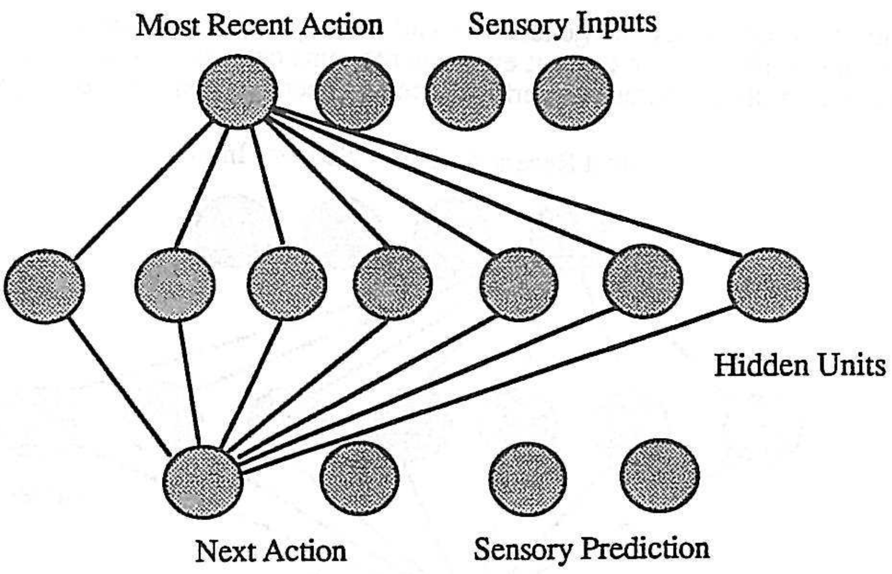


Figure 2. Animat network with predictive outputs

EXPERIMENTAL DETAILS

The experiments reported here follow those of Parisi, Nolfi and Cecconi wherever they gave explicit details. We have made the following choices where details were not given:

- The 'angle to the closest food cell' input is measured clockwise relative to the animat's current facing direction, and normalised so that food directly in front of the animat always has an angle of 0.0, food directly to the right an angle of 0.25, and so on.

- If the closest food cell is equidistant to one or more other food cells, then the one at the greatest angle is chosen to be detected by the animat.

- The distance input is normalised such that the diagonal distance across the 10x10 starting grid is approximately 1.0.

- The starting weights for the animats of generation 0 are random floating point numbers in the range ±1.0.

- In the experiments involving backprop, the original Rumelhart, Hinton and Williams algorithm was used with no momentum term, weight decay, or other convergence improvement strategy. A learning rate of 0.6 was chosen for all the runs reported here.

- At the start of each experimental run, 5 separate worlds with different, random food distributions were created. These worlds remained fixed for the duration of that experiment. In each generation, the animats were each allowed 20 lives of 50 moves in all 5 worlds. At the beginning of each life the world was restored to its initial state (i.e. any food eaten previously is replaced in the cell which had contained it) and the animat was placed in a randomly selected empty cell.

RESULTS

All experiments were repeated 16 times with different random weight starts and different worlds in order to obtain statistically meaningful results. All the figures in this section show average results over the 16 runs.

In addition to the evolutionary experiments already described, two benchmark runs were performed in which the animats' foraging strategy was hard-coded. In the first of these, the animat simply made a random action choice at each time step. Tested for 20 lives in each of 500 randomly generated worlds, this animat consumed an average of 0.63 food cells in each one. The second benchmark strategy was designed to represent an effective foraging behaviour, and consisted of the following rules. If the nearest food lies within a 40° arc in front of the animat, the chosen action is to advance. If the food lies outside this arc and to the animat's left, the animat turns to the left. Similarly, if the food is outside the arc and to the animat's right, the animat turns to the right. If the food is directly behind the animat, it turns right by default. An animat employing this strategy, tested for 20 lives in each of 500 worlds, consumed an average of 9.74 food cells in each one. Note that this is lower than the theoretical maximum of 10 foods consumed per life. One reason for this is that an animat's limited sensory information prevents it from planning any kind of optimal tour around the food cells - going directly towards each nearest food cell in turn (as per the well known greedy algorithm for the Travelling Salesman Problem) will result in a less than optimal tour.

Figure 3 shows the mean performance over the population, together with the performance of the most fit individual (peak), as a function of generation, where no learning takes place during the animats' lifetimes. The performance measure is the average number of food items eaten in a lifetime. The peak performance appears to be close to a plateau at around 8.8 food cells consumed. Note that the performance of networks generated after even a small number of generations is much better than the random walk benchmark.

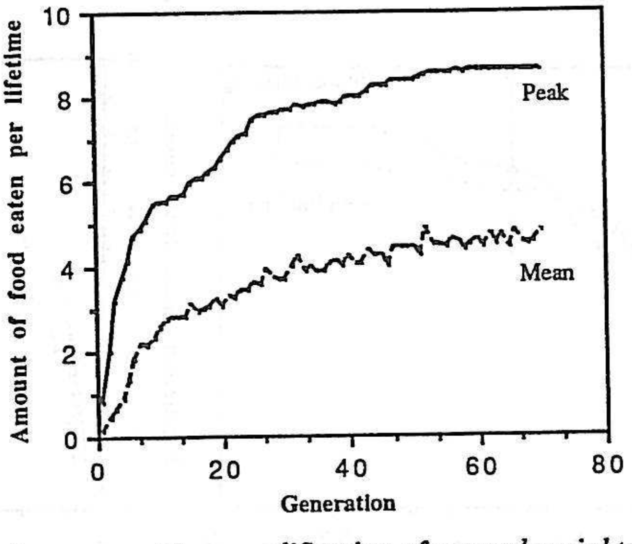


Figure 3. Animat performance with no modification of network weights during lifetime

The effect of allowing the animats to learn during their lifetime can be seen in figure 4, which shows the population mean and fittest individual performance as a function of generation.

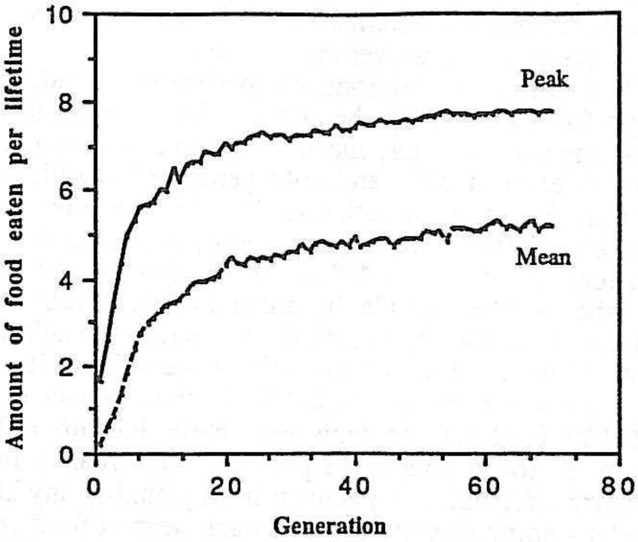
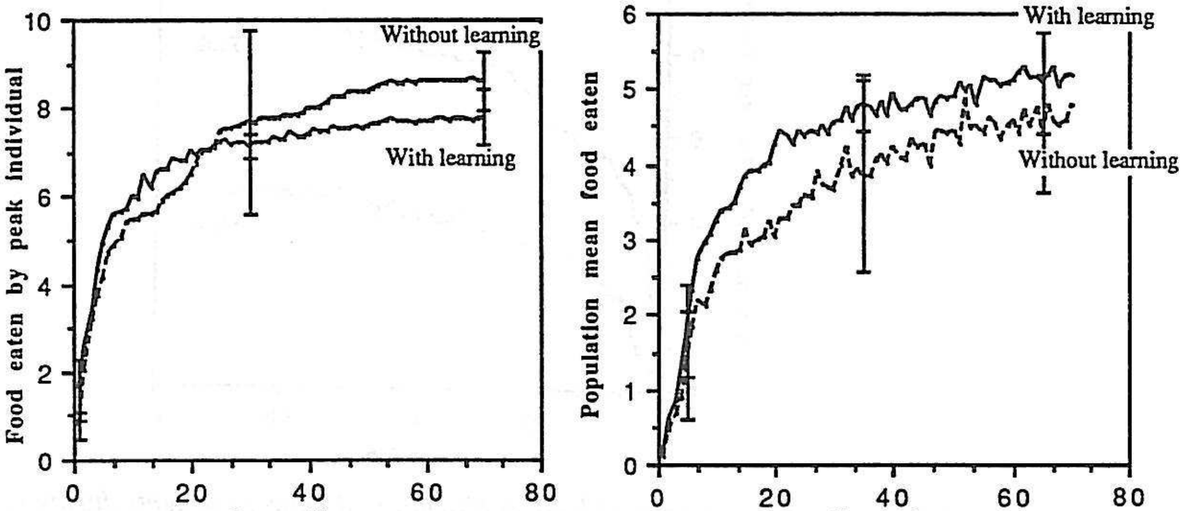


Figure 4. Animat performance with predictive learning during lifetime

Figure 5 compares the performance with and without learning for both the peak individual (fig. 5(a)) and the population mean (fig. 5(b)). Representative error bars (one standard deviation) are shown only at some generations in order to avoid cluttering the picture. In both cases there is no significant benefit resulting from learning during life; if anything, learning appears to be a handicap rather than a benefit. This is in stark contrast to the results obtained by Parisi, Nolfi and Cecconi, who observed a significant increase in performance when learning was included. We are unable to explain this difference, but in view of the large number of experiments which we have performed, we think it unlikely that further experiments would reverse our conclusions.



(a) *(b)*

Figure 5. Comparison of (a) peak and (b) mean performance with and without predictive learning

Should we expect to see an increase in food-gathering performance as a result of incorporating predictive learning during the animats' lives? It is useful to consider the 'adaptive landscape' around a point in weight space (i.e. a possible animat). A point may be considered to have a good surrounding landscape if most of its adjacent points are higher (fitter) than it is, that is to say a small movement (mutation) is likely to lead to an improvement in fitness. Parisi, Nolfi and Cecconi suggest that, as the predictive task seems correlated to the food-gathering task, then learning to predict constitutes a local exploration in weight-space, biasing selection between points of similar fitness in favour of the one with the better surrounding landscape. This causes learning to guide the evolutionary process away from local optima. This has been called the Baldwin effect (Baldwin, 1896), and a similar process has been reported by Hinton and Nowlan (1987).

Let us consider the learning task itself. The network attempts to map the relationship between 'action' and 'resultant change in sensory input'. In a world with only one food cell, this would not seem a particularly difficult relationship to grasp - the change in angle and distance to the food is a simple and consistent function of movement. However, in an environment with many food cells, the mapping may become complex and inconsistent, as the sensory inputs only ever refer to the single closest food cell. A step which results in a new food cell becoming the closest will result in a totally unpredictable (from the animat's point of view) change in sensory input. During the early stages of evolution, when the animats' movements are largely random, such steps will occur frequently as the animat moves. This effect is reduced as behaviour evolves to allow efficient movement towards closest food cells, but the same situation will always occur every time a food cell is found and consumed. This inherent noise in the training data for the predictive task casts doubt over how much useful predictive learning the animat might be able to acquire during its life.

Given the simple nature of the original task, our results (eg. fig. 5(a)) suggest that this increased complexity may actually add an unnecessary layer of abstraction between the GA and the actual task being optimised, disrupting the search space and actually hindering the search, as the GA by itself (see figure 3) seems powerful enough to find near-optimal weight sets in a relatively small number of generations.

Parisi, Nolfi and Cecconi performed a further experiment in which the animats' networks were trained on an arbitrary task (the XOR problem) while undergoing evolution, as before, on the basis of their food gathering ability. They report that performance, even on this task, becomes related to fitness for animats which are trained on the task during their lives. They suggest that the learning is biasing evolution towards selecting points from regions of weight space in which the performance surfaces for the two tasks (XOR and food gathering) are similar, so that an ascent on one surface would imply an ascent on the other.

For such regions of the space to exist, if the two tasks really are uncorrelated, the number of free parameters in the network must be large enough for the network to encapsulate both learning tasks at the same time. A network has a finite capacity for storing information; by definition, if a network consists of a *minimum* set of weights adapted for one task, then further training of these weights for another, uncorrelated task, can only decrease performance in the original task. Learning the XOR problem during the animat's lifetime only guides the evolution of food gathering ability in as much as it forces the GA to find networks which are not only good networks for food gathering, but are also tolerant of those weight changes caused by training on XOR. This tolerance may only be possible if the network architecture contains excess free parameters.

This led us to investigate whether the animats' network architecture did indeed contain more parameters than are required for the food gathering task. We performed a set of experiments in which we removed the hidden layer altogether from the animats' networks. Figure 6 compares the performance of animats with no network hidden layer with those with the original architecture of seven hidden units. Figure 6(a) compares the performance of the most fit individuals, and figure 6(b) the performance of the population means, as a function of generation. In this experiment no learning takes place during the animats' lifetimes.

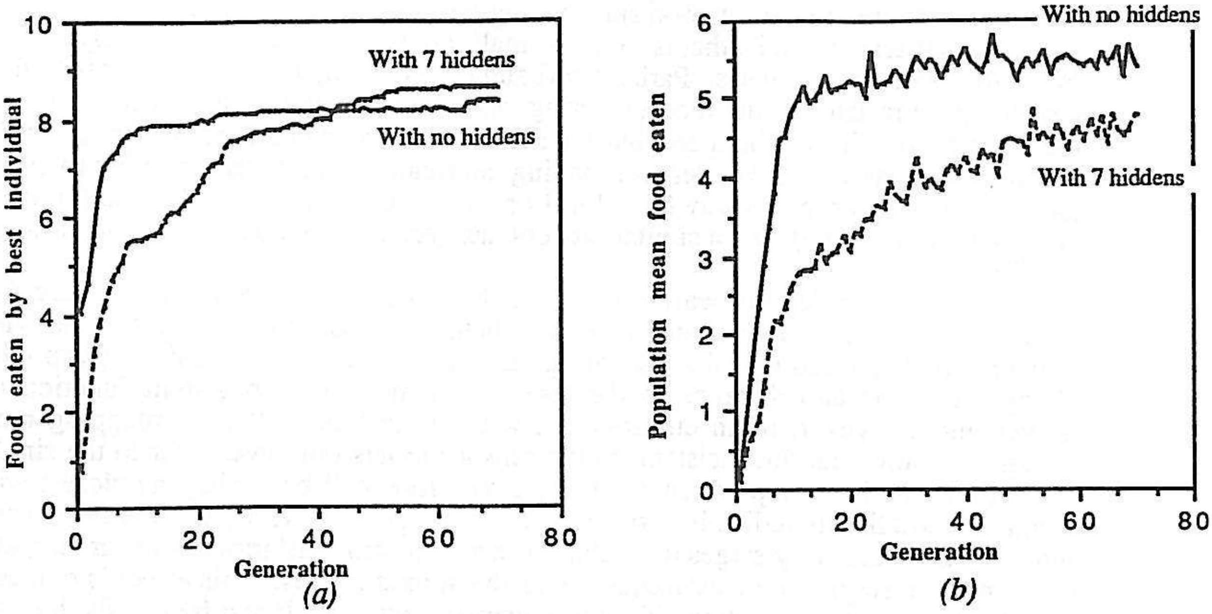


Figure 6. Comparison of (a) peak and (b) mean performance with and without a network hidden layer

It can be seen from figure 6 that the removal of the hidden layer makes no significant difference to the final ability to perform the food gathering task. In fact, the animats with the simpler architecture display a much greater ability in the early stages of the search, and evolve more rapidly, than those with a hidden layer. This can be attributed to the much smaller weight space that the GA has to search. We conclude that the original animat architecture, with seven hidden units, contains excess free parameters for this simple food gathering task. We further conclude that, as the simple perceptron (single layer) architecture can learn this task, the food gathering task may be linearly separable (see Minsky & Papert, 1969).

Finally, we consider the phenomena that Parisi, Nolfi and Cecconi observed and originally explained in terms of behavioural self-selection of stimuli. In their example, a given high performing animat (i.e. one from a late generation) was found to be much more likely, for example, to encounter food at relatively small angles (i.e. to its right) than at large angles (to its left), as it moved around its world. They also observed that the same animat was more likely to make an appropriate action choice, i.e. one which moved or oriented it closer to the food, if the food was detected at a small angle than if it were detected at a large angle. They concluded that, as the animat is able to influence its sensory input by its movements, its food gathering ability had evolved in two parts. On the one hand, it had learned to respond effectively to a reduced set of input stimuli, i.e. when the food is on the right, and on the other, it contrived to move in such a way as to encounter situations from this 'known' set of stimuli more often than other situations (such as the food being on the left). As further evidence of this process at work, it was observed that animats which were replaced at a random position after each move were much more likely to make inappropriate action choices, as they were being deprived of the benefit of self-selecting their input stimuli through their movements. This self-selection process was considered evidence of the influence of behaviour on the course of evolution.

We now offer an alternative explanation for these observations. We have observed, in these experiments, that animats from later generations tend to display a reduced behavioural repertoire, relying on turns made in one direction only. A given population might come to contain individuals which only exhibit the ability to turn left, for instance, and so would have to make three left turns in order to turn right. Obviously any optimum food gathering strategy would incorporate the ability to turn either left or right, depending on the situation, and any individual which happened to evolve such ability would perform better than its purely left-turning contemporaries, so its offspring would come to dominate the population. Initially, however, the same would be true for any individual which evolved the ability to make useful sequences of left[2] turns, among a population of individuals which didn't turn at all, or turned either way at random. As an initially randomised population evolves, then, the ability to successfully employ one kind of turn, such as left, may evolve by chance, and confer such a selective advantage that the trait would then rapidly spread through the population. Under the influence of further evolution, this ability would be refined and, the greater the refinement, the less likely it would be for right-turning ability to occur by mutation. In order to arrive at an ideal individual, with the ability to turn either way, the movement through weight space would most probably have to be greater than that possible in a single mutation. Every intermediate step between a good left-turning individual and a possible individual which could turn both ways would have to represent an improvement in fitness in order for the transition to occur, and this is very unlikely - an individual that is half way between being left-turning and 'ambidextrous' is unlikely to be better than one well adapted to being purely left-turning.

The observation that a high performing animat only ever turns in one direction seems therefore to be simply evidence of premature convergence of the population on a sub-optimal solution, a common problem with GAs (see for instance: Baker, 1985; Booker, 1987; Eshelman and Schaffer, 1991). This in itself is sufficient to account for all those observations which Parisi, Nolfi and Cecconi explained in terms of self-selection of stimuli. An individual which only ever turns left will always tend to decrease the angle of a food cell relative to itself. This would account for an observed statistical bias towards encountering small food angles during the course of the animat's life. If the food is on the animat's right (i.e. the angle is large), the animat's chosen action, to turn left, may be highly inappropriate. The same action, however, becomes more appropriate with each repeated left turn. This accounts for the observation that correct action choices are associated with more commonly encountered sets of stimuli (in this example, small angles), and also the observed drop in the frequency with which appropriate actions are chosen if the animat is randomly re-positioned after each move. The principle of Occam's Razor would tend to favour our explanation for these phenomena, that they are all evidence of premature genetic convergence, for its simplicity.

CONCLUSIONS

In this paper we have presented the results of a study of the interaction between learning and evolution, based on the experimental scenario described by Parisi, Nolfi and Cecconi (1991). While we do not dispute that the Baldwin effect can be observed in many evolutionary systems, we have found no evidence that learning has a beneficial effect on evolution within the context of these experiments.

We have shown that the food gathering task in these experiments is learned more successfully by simple perceptrons than by multi-layer networks, which suggests that it is a linearly separable problem. Finally, we have offered a more simple explanation, that of premature convergence due to loss of population diversity, for the phenomena which Parisi, Nolfi and Cecconi explain as behavioural self-selection of input stimuli.

ACKNOWLEDGEMENTS

This work has been supported by the Science and Engineering Research Council (SERC). Bryn Williams would like to thank British Telecom for their support via a Total Technology Studentship.

[2] 'left' is used here for the sake of discussion. No bias towards convergence on either direction was observed.

1149

REFERENCES

Baker, J. E. 1985. Adaptive selection methods for Genetic Algorithms. *Proceedings of an international conference on Genetic Algorithms*, Morgan Kaufmann

Baldwin, J.M. 1896. A new factor in evolution. *American Naturalist,* 30, 441-451

Booker, L. 1987. Improving Search in Genetic Algorithms. *Genetic Algorithms and Simulated Annealing* Ch 5. Pitman, London

Eshelman, L. J. and Schaffer, J. D. 1991. Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. *Proceedings of the fourth international conference on Genetic Algorithms*, Morgan Kaufmann

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning* Reading, Mass.: Addison-Wesley

Hinton, G.E. and Nowlan, S.J. 1987. How Learning Guides Evolution. *Complex Systems*, 1, 495-502.

Holland, J.J. 1975. *Adaptation in Natural and Artificial Systems.* Ann Arbor, Michigan: University of Michigan Press

Minsky, M. and Papert, S. 1969.*"Perceptrons: An Introduction to Computational Geometry"* MIT Press, Cambridge

Parisi, D., Nolfi, S. and Cecconi, F. 1991. Learning, Behaviour and Evolution. *Proceedings of ECAL-91 - First European Conference on Artificial Life*, Paris

Rumelhart, David E & McClelland, James L. (eds.) 1986. *Parallel Distributed Processing*. The MIT Press, Cambridge, Massachusetts

Rumelhart, D. E., Hinton, G.E. and Williams, R.J. 1986a. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland (eds.), *Parallel Distributed Processing*. Vol 1. The MIT Press, Cambridge, Mass.