Learning and Evolution in a Continuous Environment

Renuka Nayak & Laura Fox

Abstract

In an experiment inspired by Nolfi, Elman and Parisi (1994), two populations of robots controlled by recurrent neural networks were evolved using a genetic algorithm that operated on starting network weights. Robot evolutionary fitness was based on food-finding ability in a continuous environment. The robot control networks of one of the two populations learned, via backpropagation, to predict a robot's next set of sensory inputs—the other did not. The two robot populations were compared in order to assess the effects of learning on evolution and evolution on learning, when evolution and learning were done on different tasks. Our results suggested that learning to predict in a continuous environment did not give robots an advantage in food finding over robust hard-coded strategies that remained unchanged by learning. Therefore, perhaps the model of interaction between learning and evolution put forth by Nolfi et al. may not be appropriate in continuous environments.

Introduction

Learning and Evolution

Many complex biological organisms, humans included, exhibit a wide range of intelligent and adaptive behaviors appropriate to the continuous and variable environments in which they live. Two distinct but interrelated natural processes can affect the development of adaptive behavioral tendencies within a species and within particular individuals of that species – *evolution* and *learning*. While both evolution and learning can affect eventual behavioral outcomes, they operate on very different levels. Evolution operates at the level of the *population,* on a time scale that spans many generations. That is, members of a population with genes that best facilitate successful reproduction are able to pass on those genes to the next generation, thus ensuring the propagation of good genetic material through the gene pool. The intelligent behavior that a member of an evolved population exhibits may be due in part to the inheritance of quality genes that encode for adaptive behavioral tendencies. Learning, on the other hand, operates at the level of the *individual,* within the time frame of an individual lifespan. Based on feedback from the environment, an individual organism may alter its behavioral tendencies during its lifetime to

make them better suit that environment. The intelligent behavior demonstrated by an individual within a population, then, may also be due in part to the process of learning what actions are most appropriate and useful in different circumstances.

Evolution is a powerful tool because an individual who is born with a set of adaptive behavioral tendencies that function well in its environment is better off than individuals who are not born with such an advantage. Baldwin (1896) suggested that learned behaviors that are beneficial to the individual spread throughout the population in successive generations. If a learned behavior continues to benefit individuals, then evolution will select for individuals that have genetically assimilated the behavior into their genotype. Individuals that have genetically assimilated the beneficial behavior into their genome are at an advantage over those who have to learn the behavior or those who don't have the behavior at all. However, because environments are not always stable across generations, a genetically hard-coded behavioral tendency that works well for one generation may not work well for the next generation, and hence the frequency of the gene or set of genes that codes for it will decrease in the gene pool. If assimilating hard-coded behaviors fails, then a learning mechanism can allow for adaptation to environmental circumstances that are not stable enough across generations and can promote the evolution of flexible agents that can adapt to their environment. An individual who is able to learn can adapt its behavior to the particular environment it has been born into, and can even adapt to changes in that environment within its lifetime.

Although the processes of learning and evolution operate on different levels and within different time scales, they can nevertheless interact and influence one another. For instance, the ability to learn that is shared by many complex organisms almost certainly *evolved* – that is, a genotype that makes learning from the environment possible must have facilitated the survival and reproduction of individuals from previous generations who carried it, leading to its spread throughout animal populations. The ability to learn confers evolutionary advantages, and the process of evolution can select genetic material that makes learning easier and faster for those who carry it. The interplay between learning and evolution is complex and interesting, but difficult to study with regard to biological organisms like humans, who evolve on a time scale of millions of years, and learn on a timescale of decades.

Because both learning and evolution contribute to intelligent behavior in biological organisms, researchers in artificial intelligence and robotics have tried to find ways of simulating these processes in the hopes of producing more intelligent behavior in artificial systems. Genetic algorithms can simulate evolution by allowing researchers to define a "genetic code" and to create a random "population" of programs, robots, etc. with different genotypes that perform in different ways. Researchers also create a "fitness measure" by which individuals who perform better on a certain task or measure are selected to "reproduce" and spread their "genes" to the next generation. Artificial neural networks can simulate learning through backpropagation or other learning algorithms, which are processes by which a network is altered based on the disparity between its actual output and its desired output in response to a particular input. Of course, genetic algorithms and neural networks do not exactly replicate their biological counterparts, but they do approximate the basic processes at an abstract level. Thus, through the use of genetic algorithms and neural networks, highly controlled experimental research on evolution and learning is possible.

Some research has been conducted using both evolution (via genetic algorithms) and learning (via neural networks). Many researchers have used neural networks which learn, and have evolved genes which specify initial internal *connection weights* for those networks. Other aspects of neural networks, such as their architectures and learning rules, can also be evolved, and research has been done that explores different such techniques (Yao, 1993). Although this body of research explores both evolution and learning, it does not fully explore the interrelationship between the two processes.

When evolution and learning are done concurrently on the same task, we cannot easily attribute changes in behavior to one process or the other. That is, if individuals who perform well on a given task are selected to reproduce and propagate their genes, evolution is said to be operating on that task. If individuals also alter their behavior based on feedback about performance on that same task during their lifetime, then learning is also operating on that task. When evolution and learning are both operating on the same task like this, their separate effects on each other and on task performance cannot be disentangled. Nolfi, Elman, and Parisi (1994) recognized this problem and decided to conduct an experiment in which evolution and learning operated on different (but plausibly related) tasks, in order to examine the relationship between learning and evolution.

Nolfi, Elman and Parisi (1994)

In an interesting study, Nolfi, Elman and Parisi (1994) set out to explore learning and evolution using genetic algorithms and neural networks to control simple simulated agents called "animats" operating in a two-dimensional grid-world environment (See Figure 1). Nolfi et al. used two distinct tasks for evolution and learning. Animats were selected to pass on their "genes" (starting neural network weights) based on their ability to find and consume "food" in the environment, and agents received feedback during their individual lifetimes that allowed them to learn (through backpropagation) to predict their next set of sensory inputs given their current sensory inputs. In this way, evolution operated on the food-finding task, and learning operated on the sensory-prediction task.

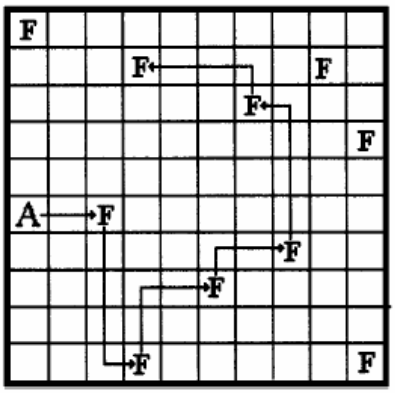

Figure 1. Nolfi's Grid World. An "animat" could move forward, turn right, turn left, or stay in one place. Its "sensors" indicated the angle and distance to the nearest piece of food in the grid world.

Nolfi, Elman and Parisi were able to look at the interplay between learning and evolution due to the fact that they were done on different tasks. They examined food-finding and predicting ability across generations and within individual animat lifetimes. They also compared a population of animats that simply evolved with a population that *both* evolved *and* learned during individual lifetimes. The three major findings from their study were: (1) Learning-and-evolving populations showed a better evolutionary increase across generations in average ability to find food than populations of individuals that evolved without learning, (2) Individuals from later generations of learning-and-evolving populations got better at food finding during their lifetimes as they learned to predict their sensory input, and (3) Individuals from later generations of learning-and-evolving populations seemed to

inherit a predisposition to learn to predict more quickly than individuals from earlier generations.

Kolen and Pollack (1990) showed that when the backpropagation learning algorithm is used to simulate learning in a neural network, the initial network weights have a very significant effect on the amount of time it takes the network to converge on a solution to the learning problem. This is important, because it means that evolution (via genetic algorithm) that is done on the starting weights of a neural network can, in theory, have a large effect on an evolved network's ability to learn quickly through backpropagation. This effect on starting network weights is one of the mechanisms through which evolution could have influenced learning performance in the experiments done by Nolfi et al.

The results of Nolfi, Elman and Parisi (1994) revealed a complex and mutually influential interaction between learning and evolution in contributing to animat behavior. Even though learning and evolution were done on different tasks, Nolfi et al. concluded that they influenced each other because the *performance surfaces* for prediction and food-finding were dynamically correlated. That is, evolution selected individuals located in a region of network weight space from which network weight changes due to prediction learning would be likely to increase evolutionary fitness (food-finding ability).

Learning and Evolution in a Continuous Environment

Although Nolfi, Elman and Parisi had very intriguing results, we thought that there was at least one crucial limitation in their method. Because theirs was one of the first studies of its kind, they (understandably) used very simple simulated agents and a very simple, discrete environment for their evolution and learning experiments. Their "animats" lived in a 2-dimensional grid world, were able to turn left, turn right, go forward, or do nothing, and had sensors that simply indicated the angle and distance to the nearest piece of "food" in the environment. There was none of the noise or continuity and little of the complexity of the real world in this grid world. Accordingly, we decided it would be interesting to see if we could replicate Nolfi, Elman and Parisi's pattern of results when learning and evolution are done using a continuous robot simulation instead of the discrete animat grid world.

We hypothesized that, if we could successfully create a population of simulated robots that evolved to find food and learned to predict sensory inputs, we would find that

our learning-evolution interaction results replicated those of Nolfi, Elman and Parisi. However, replicating their interaction results would not be possible if the food-finding and prediction-learning task surfaces were not dynamically correlated in our continuous environment, or if the prediction-learning tasks were too difficult for our neural networks to accomplish in that environment. A further possibility would be that the Nolfi et al. interaction model would not be appropriate if either evolution or learning were too strong a force on its own to be highly affected by the other process in our simulation.

Method

Purpose. To create a population of robots that learn to predict their next sensor values and evolve to find food, and to examine the interactions between learning and evolution.

Apparatus. All experiments were done in a simulator. We used the Player/Stage package to implement our experiments. A simulated USC Pioneer (Figure 2) existed in a world (Figure 3) with 16 randomly placed pucks. Pucks represented "food" in our simulation. The world was circular with a radius of about 4 meters and there were no obstacles in the world aside from the pucks. The pucks had a friction of 0.7 so that when robots tried to consume them with their grippers, the pucks didn't bounce around excessively.

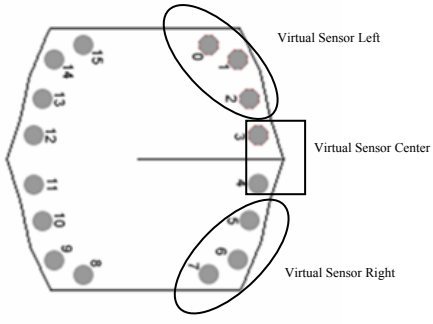

Figure 2. USC Pioneer. We used a simulated version of the Pioneer. Instead of using all 8 frontal sonars, we created three virtual sonars that represented a portion of the real sonars.

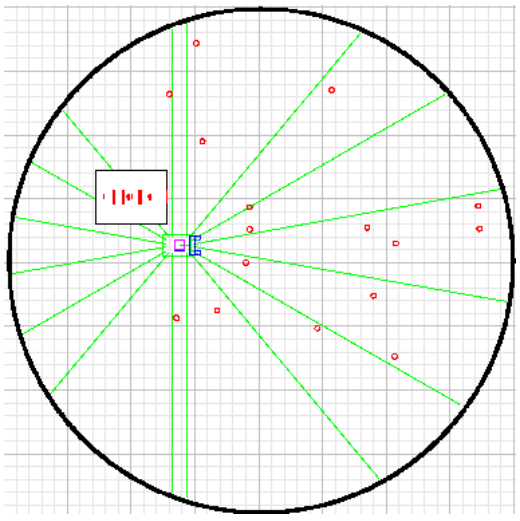

Figure 3. **Puck World**. Sixteen pucks were randomly placed in a circular environment. Our simulated Pioneer was also randomly placed in the environment. The dots represent the pucks and the lines emanating from the robot represent sonar readings.

Neural Network. A simple recurrent neural network served as the robot controller. The network consisted of 9 input nodes, a 4-node hidden layer, a context layer that held the contents of the hidden layer from the previous time step, and a 9-node output layer (Figure 4). There were 2 inputs for the motor sensors (translate, rotate) and 7 inputs for the sensors, including virtual sonars (real values), puck range (real values) and puck position (bit values) (Figure 5). Virtual sonars were used so as to minimize the size of the network (see Figure 2). Values from the input layer were delivered to the hidden layer and a direct copy of the activations of the hidden layer was delivered to the context layer. Since the goal of evolution was to find food, the weights to the motor output were frozen—only the genetic algorithm could modify these weights.

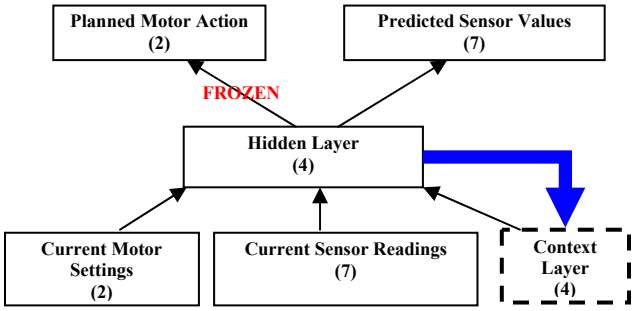

Figure 4. Schematic: Simple Recurrent Network. An individual robot could learn by changing the weights of this network. Weights between the hidden layer and the motor output could not be modified by the individual.

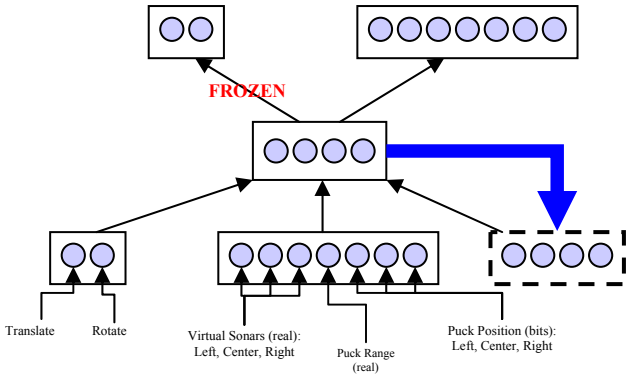

Figure 5. Details: Simple Recurrent Network. All values inputted into the network were real values ranging between 0 and 1, except for the puck position values, which were bit values representing the location of the closest puck.

Learning. The neural network was trained in real time to predict the next sensor values that the robot would get from its environment given the current motor values and the current sensor readings. Back propagation was used to modify the network weights to achieve this goal (weights between the hidden layer and the motor outputs were *not* modified using the back propagation algorithm). The learning rate was set to 0.1, the momentum was set to 0.5 and tolerance was set to 0.1. We used conx.py code written by Doug Blank and Lisa Meeden to set up and modify the neural network. Pyro was used to interface between the neural network and the simulated robot. In order to examine the effects of learning, we ran

two versions of our genetic algorithm – in one, individual robots learned to predict during their lifetimes, and in the other, individual robots did not learn during their lifetimes.

Evolution. A generation consisted of 20 individuals who were tested for their ability to consume pucks. Those that consumed the most pucks in their lifetime were more likely to reproduce and pass on their genes to the next generation of robots. A lifetime was defined as 3 minutes (approximately 1720 time steps) and genes were one-to-one representations of the initial network weights of the robots (our robots had a genome consisting of 101 genes). The mutation rate was set to 0.3 and we did not allow crossover to occur. After 50 generations of selection and evolution, we looked at the best individuals from the 1st, 10th, 20th, 30th, 40th and 50th generations.

Problems. Due to some sort of bug in the Player/Stage package, our genetic algorithm would terminate (randomly) after about 10-15 hours. Each time this happened, we restarted the genetic algorithm with saved weights from previous runs. Because of this issue, we were unable to run as many generations of the GA as we would have liked.

Results

Peak and Average Puck Consumption by Generation. Our learning and non-learning populations were both evolved for a total of 50 generations. Both average and peak fitness increased steadily through the generations for both the learning and non-learning populations. Across generations, the number of pucks eaten by the best robots in the non-learning population (N = 50; M = 10.10; SD = 3.17) was significantly higher than then number of pucks eaten by the best robots in the learning population (N = 50; M = 8.92; SD = 1.59) (t = 2.189, p < 0.05, df = 98) (See Figure 6). There was also a significant difference in the average number of pucks eaten by learning and non-learning populations in generations 1 through 40, with the learning population having a slightly higher average (t = 0.433, p < 0.001, df = 78). After generation 40, however, the non-learning population has a higher average number of pucks eaten (N = 10; M = 4.61; SD = 0.56) than the learning population (N = 10; M = 7.30; SD =

2.13). Data taken across *all* generations show that there is no significant difference between the average number of pucks eaten by learning vs. non-learning individuals.

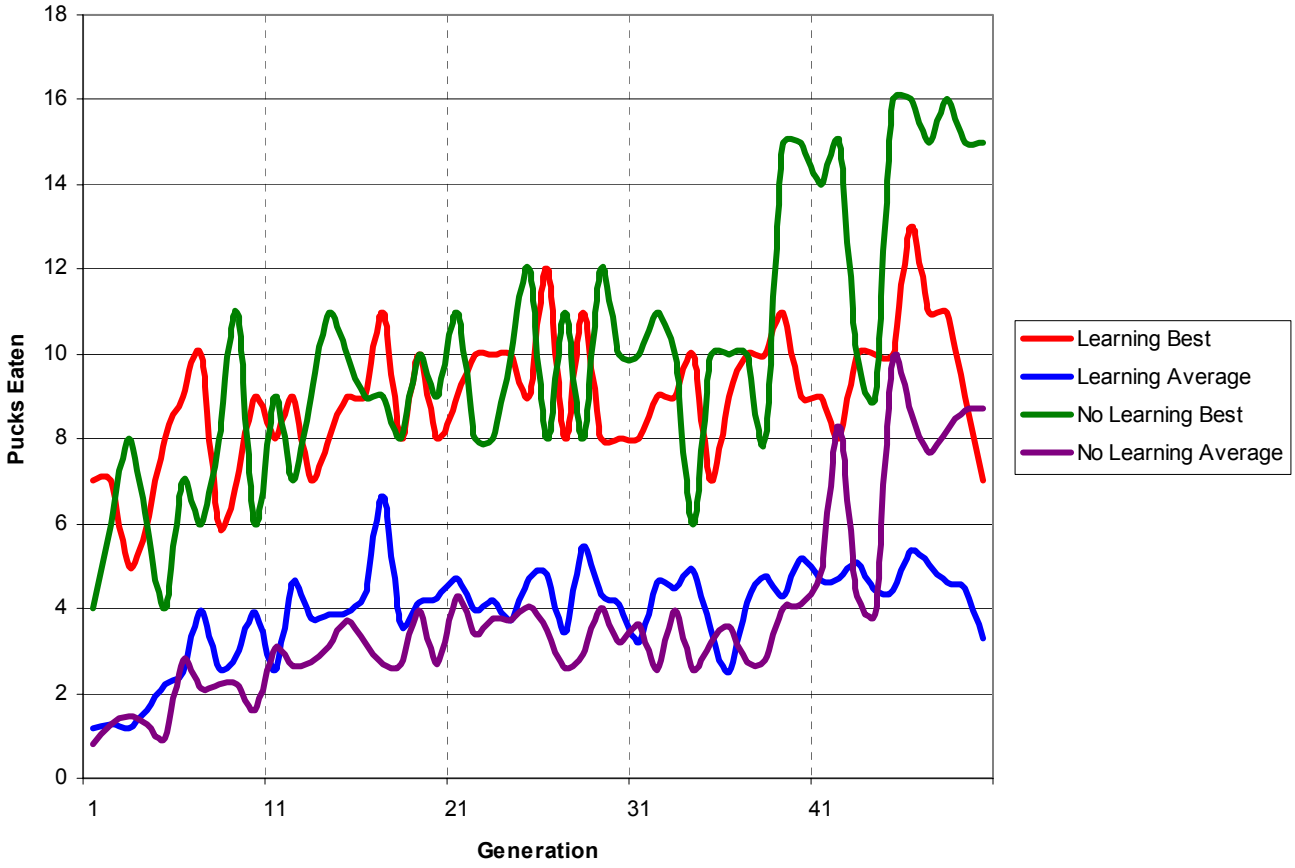

Figure 6. Puck-Eating Performance Across Generations. Graph displaying peak and average puck consumption for robots in generations 1-50 of the learning and non-learning populations. The non-learning population seemed to gain an advantage in the later generations.

Learning Error Drop-off. We also examined lifetime prediction learning error drop-off for the best individuals from generations 1, 10, 20, 30, 40, and 50 of both the learning and non-learning populations (individuals that had evolved in the non-learning population were given the ability to learn during these tests). As can be seen in Figures 7 and 8, learning error for individuals from both populations dropped off in the first 200 time steps and then seemed to reach a plateau. Interestingly, individuals from the evolved-non-learning population reached a lower error plateau than those from the evolved-learning population. There was no increase in the initial rate of error drop-off in later generations of the evolved-learning population, although there seemed to be a slight such trend in the evolved-non-learning population.

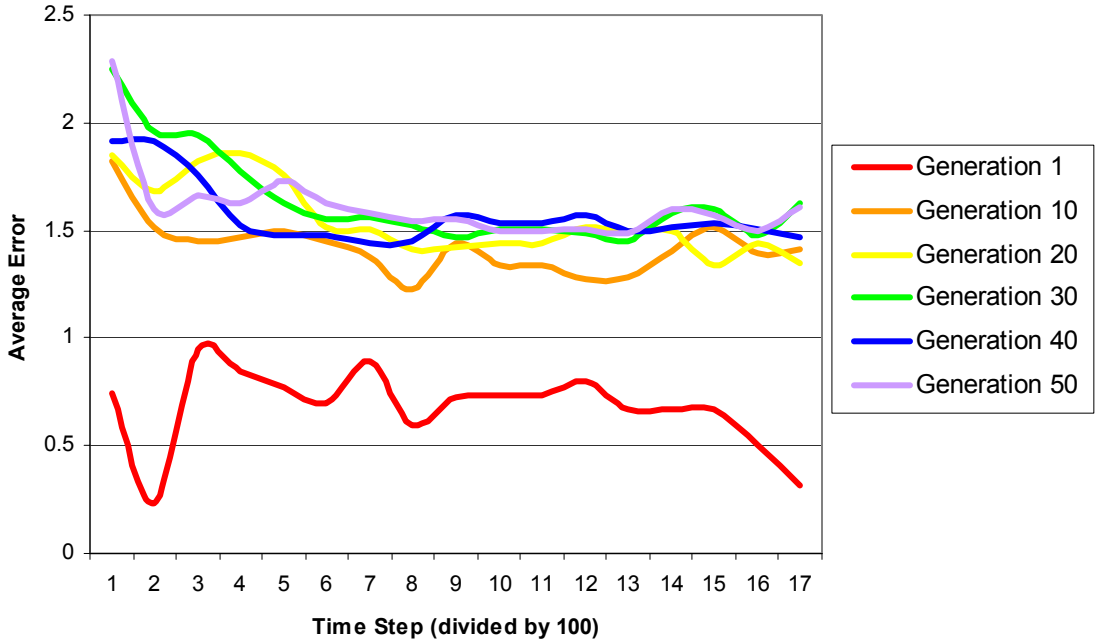

Figure 7. Learning Error Drop-Off – Learning Population. Graph displaying the drop-off of learning error during 1700 steps (3 minutes) of testing for the best individuals from the 1st, 10th, 20th, 30th, 40th and 50th generations of the evolved-learning population. Error was averaged across every 100 time steps.

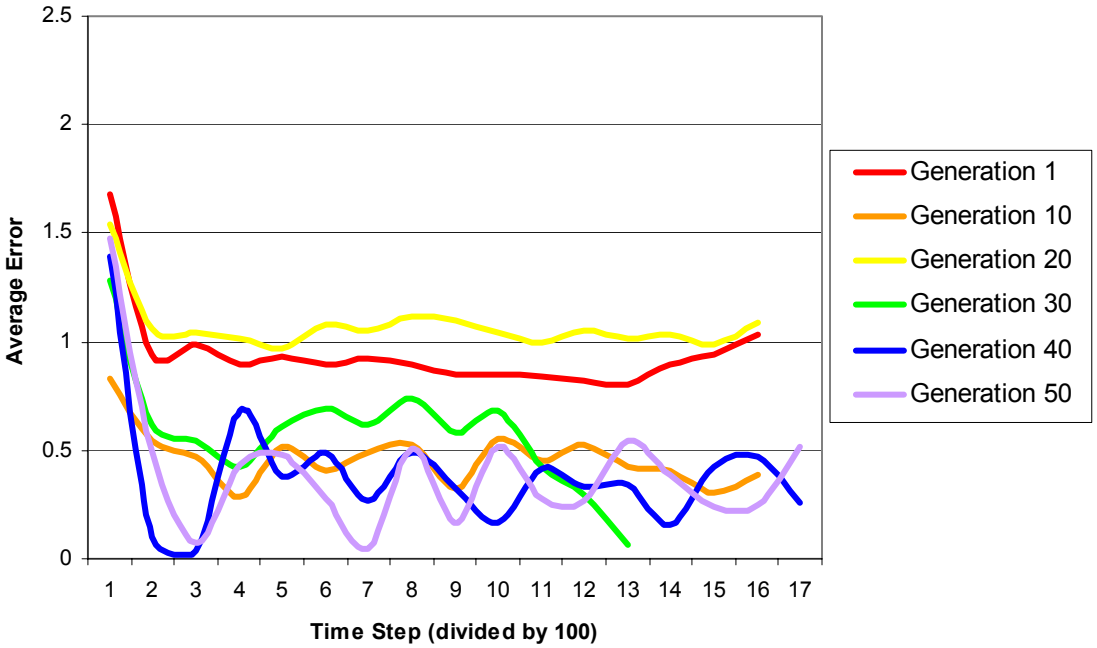

Figure 8. Learning Error Drop-Off – Non-Learning Population. Graph displaying the drop-off of learning error during 1700 steps (3 minutes) of testing for the best individuals from the 1st, 10th, 20th, 30th, 40th and 50th generations of the evolved-non-learning population. Error was averaged across every 100 time steps. The plunge in error shown in generation 30 was due to the fact that that robot became stuck against a wall (with no further change in sensor inputs), making prediction easy.

Puck-Eating During Individual Lifetimes. We examined puck eating performance within the individual lifetimes of the best individuals from generations 1, 10, 20, 30, 40, and 50 of the learning population in order to try to assess whether or not learning during a lifetime increased puck-eating performance (rate of puck-eating) for highly evolved individuals. It appears from the graph in Figure 9 that there may be a higher increase in the rate of puck-eating in later generations as learning progresses during a lifetime, at least during the initial learning-error drop-off period (the first 200 time steps or so), than in rate of earlier generations.

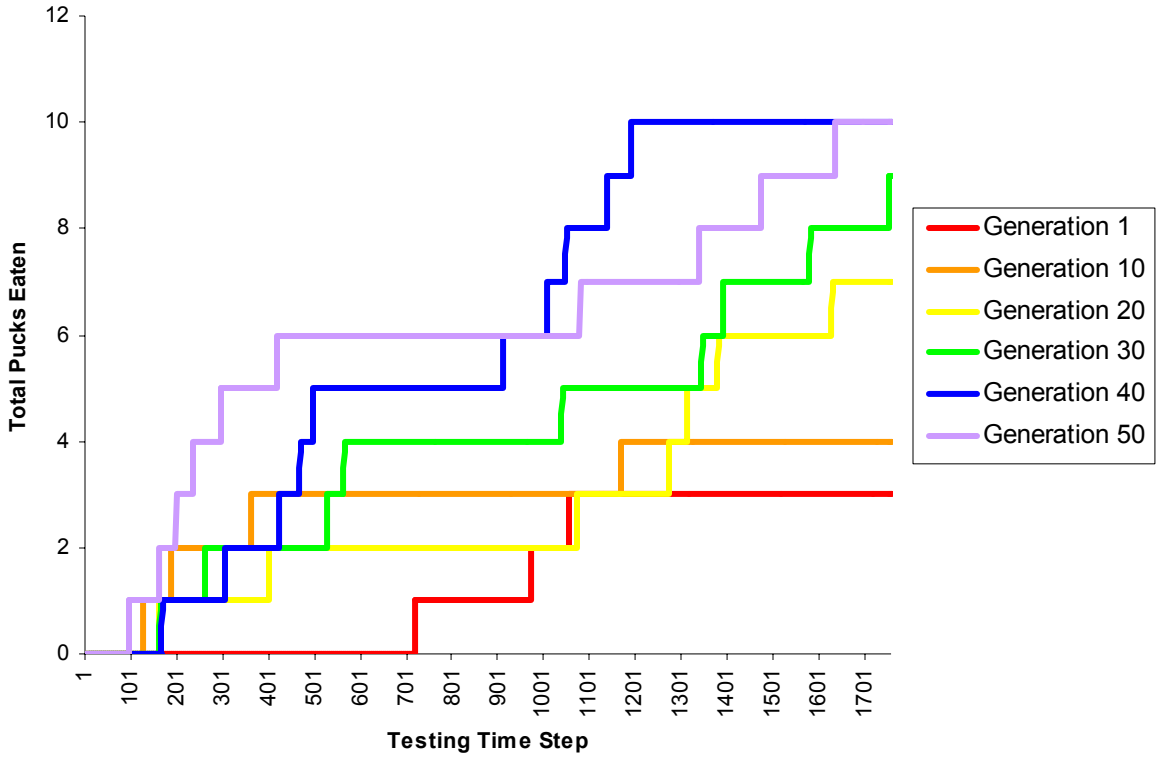

Figure 9. Puck-Eating Progress Within A Lifetime. Graph displaying the puck-eating progress made during 1700 steps (3 minutes) of testing by the best individuals from the 1st, 10th, 20th, 30th, 40th and 50th generations of the evolved-learning population.

Puck Eating Performance With or Without Learning. The best individuals from the evolved-learning and evolved-non-learning each had puck-eating performance tested with or without prediction learning enabled. A total of 5, 1700-step (3 minute) trials were done for both individuals in both learning and non-learning conditions, and the average puck-eating totals across those 5 trials are shown in Figure 10. Both the evolved-learning and the evolved-

non-learning individuals performed much better when tested under the same conditions they were evolved under.

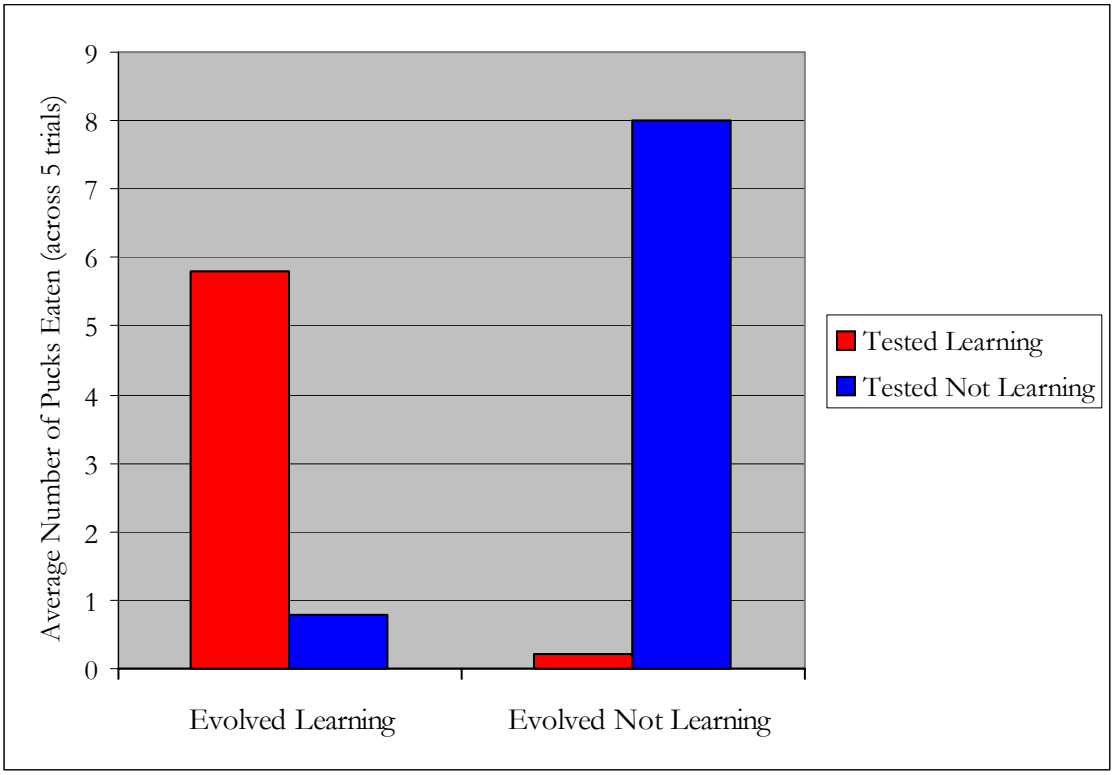

Figure 10. Puck-Eating Under Different Testing Conditions. Graph displaying the average number of pucks eaten across 5 trials by the best individuals from the 50th generation of the evolved-learning and evolved-not-learning populations when tested with or without learning enabled.

Puck-Eating Strategies. Almost all successful robots displayed strategies that involved spinning around the rink in irregular circles, a behavioral tendency that makes it less likely that a robot will get stuck against the rink wall, and that allows a robot to eventually cover a lot of the space within the rink, occasionally running into and consuming pucks. One individual in the learning population evolved early on to move around in a leftward circle and lunge at pucks that were in its range. Individuals in the later generations of the non-learning population tended to display more intelligent puck-seeking behavior (this is reflected in the large increase in average and peak performance for the non-learning population in generations 40-50, see Figure 6). A robot displaying one of these more intelligent strategies would spin until it was facing a puck, and then would begin on a curving path in one direction toward that

puck, periodically backing up and turning the other direction to keep the puck in its sight. For all robots examined, the tendency was always to turn to the left.

Discussion and Conclusions

The overall results of our simulation did not seem to parallel those of Nolfi, Elman and Parisi (1994). Nolfi et al. found that learning populations showed a better evolutionary increase in food-finding ability than non-learning populations. Although the average puck-eating performance of the learning population in our simulation was higher than that of the non-learning population for generations 1-40, after generation 40 the non-learning population gained an advantage in both average and peak puck-eating performance. We speculate that perhaps the learning population had an advantage during earlier generations because good, hard-coded, non-learning strategies had not had time to evolve in the non-learning population yet. Between generations 40 and 50, sufficiently good, hard-coded strategies were able to finally emerge, giving the non-learning population an advantage. The learning population would have been unable to evolve robust, hard-coded puck-finding strategies due to the fact that network weights were constantly changing within a lifetime. However, the simple fact that network weights were constantly changing may have been what gave individuals the learning population an advantage over the unsuccessful hard-coded strategies in the earlier generations of the non-learning population—going around in circles of different radii is clearly better than going around in the same circle.

Considering that the environment was relatively stable, perhaps the Baldwin effect was taking place in the evolution of these two populations—those individuals that were able to inherit a genetically assimilated beneficial behavior out-performed individuals that did not already have this behavior hard-wired into their genome. Indeed, our experiments suggest that the best non-learning individuals out-perform the best learning individuals. It would be interesting to see whether continuing the GA past 50 generations would allow us to discern whether on average non-learning individuals outperform learning individuals in puck-finding ability. It is difficult to say why Nolfi et al. did not demonstrate the same effect in their experiments. One difference between our experiment and that of Nolfi et al. is that in our experiment learning to predict, seemingly, did not greatly help robots at the puck-finding

task. So perhaps the prediction learning and puck finding evolution surfaces in our experiment were in fact not sufficiently dynamically correlated in our continuous simulation to replicate the findings of Nolfi et al.

Nolfi et al. also found that individuals from later generations of evolved-learning populations seemed to inherit a predisposition to learn to predict, as demonstrated by the fact that those individuals showed faster learning error drop-offs than individuals from earlier generations. This was not the case for our learning population, although it seemed to be the case for our non-learning population. This is an odd result, because there is no reason why individuals in the non-learning population should have inherited any kind of predisposition to learn to predict when given the ability to learn. Although later generations of evolved-non-learning individuals have a higher rate of error drop-off initially, this does not necessarily mean that they somehow possess a predisposition to learn to predict. Perhaps the genetic algorithm is responsible for this pattern in learning error fall-off. If the non-learning robots are behaving like rapidly-learning prediction robots, then this would suggest that prediction may actually be helpful to robots that are trying to find pucks. By this account, a set of initial network weights that is successful at puck-finding might be a better set of initial network weights for learning to predict than a set of initial weights that is not as successful at puck finding. Therefore, the more successful non-learning robots would be better at prediction than the less successful learning robots. In other words, the learning robots have not yet perfectly grasped the prediction skill, even though that is their learning goal, whereas the GA has placed individuals of the non-learning populations in a weight space which allows them to be better at prediction than their learning counterparts.

Our results may have replicated Nolfi's third finding – individuals from later generations of learning populations seemed to get better at food-finding as they learned to predict during their lifetimes, at least in the first 200 time steps. These first 200 steps coincide with the rapid decline in learning error in learning robots. We suspect that a further decline in learning error did not occur because the network weights settled into a more stable state after the initial error drop-off.

We have also shown that for both populations, the best evolved individuals were vastly better at puck-eating when tested under the same conditions they were evolved under – i.e., the best member of the 50^{th} generation from the learning population performed very poorly when it wasn't allowed to learn, and the best member from the 50^{th} generation of the

non-learning population performed very poorly when it was given the ability to learn. Because learning individuals that were not allowed to learn perform badly, evolution alone cannot be responsible for the success of the learning robots. Conversely, because non-learning individuals under-perform when forced to learn, learning alone cannot improve puck-finding ability. These results further support the concept that evolution and learning interact in evolving, learning populations.

Improvements upon our experiment abound. First and foremost, we would have liked to have run our GA for longer than 50 generations to see whether the trends presented in this paper would continue in later generations. We also would have liked to have done more than a single replicate of the entire experiment. It is also possible that using different learning parameters and genetic algorithm parameters might have resulted in a different pattern of findings. Further experimentation might include making the environment more unstable by adding obstacles which change position every generation. This change might prevent the Baldwin effect from occurring and may lead to results that may be similar to Nolfi's results. Additionally, a non-circular world might be used to see whether agents would develop smart strategies other than turning in circles, although such an experiment might not look directly at the interaction between learning and evolution.

Overall, our findings suggest that Nolfi's model of the interaction between learning and evolution may not be appropriate for a simulation in a continuous environment. We suspect that there is something more complex happening with regards to the interaction between learning and evolution but we do not have the abundance of data that we need to exactly determine what the true interplay between evolution and learning might be.

References

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

Kolen, J.F. & Pollack, J.B. (1990). Backpropagation is sensitive to initial conditions. *Complex Systems,* **4**, 269-280.

Nolfi, S., Elman, J. L., & Parisi, D. (1994). Learning and evolution in neural networks. *Adaptive Behavior*, **3**, 5-28.

Nolfi, S. & Floreano, D. (1999). Learning and evolution. *Autonomous Robots*, **7**, 89-113.

Yao, X. (1993). A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, **4**, 203-222.