UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

NURTURING AS SAFE EXPLORATION PROMOTES THE EVOLUTION OF

GENERALIZED SUPERVISED LEARNING

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

By

BRYAN HOKE
Norman, Oklahoma
2017

NURTURING AS SAFE EXPLORATION PROMOTES THE EVOLUTION OF
GENERALIZED SUPERVISED LEARNING

A THESIS APPROVED FOR THE
UNIVERSITY OF OKLAHOMA

BY

_____

Dr. Dean Hougen, Chair

_____

Dr. Amy McGovern

_____

Dr. Ingo Schlupp

Contents

Abstract

The ability to learn is often a desirable property of intelligent systems which can make them more adaptive. However, it is difficult to develop sophisticated learning algorithms that are effective. One approach to the development of learning algorithms is to evolve them using evolutionary algorithms. The evolution of learning is interesting as a practical matter because harnessing it may allow us to develop better artificial intelligence; it is interesting also from a more theoretical perspective of understanding how the sophisticated learning seen in nature—including that of humans—could have arisen. A potential obstacle to the evolution of learning when alternative behavioral strategies (e.g., instincts) can evolve is that learning individuals tend to exhibit ineffective behavior before effective behavior is learned. *Nurturing*, defined as one individual investing in the development of another individual with which it has an ongoing relationship, is often seen in nature in species that exhibit sophisticated learning behavior. It is hypothesized that nurturing may be able to increase the competitiveness of learning in an evolutionary environment by ameliorating the consequences of incorrect initial behavior. The approach taken is to expand upon a foundational work in the evolution of learning to enable also the evolution of instincts and then examining the strategies evolved with and without a nurturing condition in which individuals are not penalized for mistakes made during a learning period. It is found that nurturing promotes the evolution of learning in these environments.

Chapter 1

Introduction

Chalmers [1] demonstrated that evolutionary processes can produce artificial systems that learn. The learning that emerged could be either specific to certain tasks or generalized for any task in the problem space. It was also possible that learning might not emerge if none of the behavior adjustment rules evolved performed better than adjusting behavior randomly.

In Chalmers's work, learning was the only viable phenotypic strategy that could emerge from evolution. However, if other strategies, such as fixed behavioral responses (responses that don't change during an individual's lifetime), were able to emerge alongside the strategy of learning, it might be found that learning is not always the strategy that emerges from evolution. Presumably, the evolutionary environment (or objective function) would have the effect of influencing which strategies emerge; after all, it is reasonable to think that no one strategy would always be optimal for every possible kind of environment.

A disadvantage of learning is that individuals tend to have low fitness early in their life before they have had the opportunity to learn correct behavior. This is in contrast to individuals that exhibit instinctive behavior and have high fitness at the start of life. Such effects could make it less likely for learning to evolve in favor of instinctual behavior.

However, if the penalties incurred during the early stages of learning could be mitigated then learning may be more likely to evolve. In the biological world this can be observed in the form of nurturing, often from a parent to a child. In this way,

nurturing may increase the competitive viability of learning by protecting the learner from otherwise costly errors made before proper behavior has been acquired [7].

This work examines whether learning is less likely to evolve when both instincts and learning have the potential to emerge, and how the introduction into the environment of a "nurturing" condition—where mistakes made during learning are not penalized—influences the evolution of learning as opposed to an "instinctual" strategy where behavior is fixed during an individual's lifetime.

To explore these questions, this work uses a combination of genetic algorithms and artificial neurons applied to a set of supervised learning tasks (the same tasks, in fact, that were used by Chalmers). The rest of this chapter introduces the following topics, in turn: genetic algorithms, artificial neurons, neuroevolution, the evolution of learning, and nurturing.

1.1   Genetic Algorithms

Genetic algorithms are a class of search algorithms inspired by the process of evolution by natural selection. Solutions to a problem are encoded in strings referred to as *chromosomes*. The distinct variables in the solution representation are encoded in substrings of the chromosome referred to as *genes*.

The procedure begins with a population of chromosomes which are typically randomly generated. The chromosomes are evaluated by an objective function—the problem to be solved—referred to as the *fitness function* to determine the fitness value of each one. The goal of the genetic algorithm is to find the chromosome which maximizes the fitness function and has the highest fitness; in this sense, a genetic algorithm can be used as a search algorithm that looks for the best solution for an objective function.

A process of selection is used to determine the chromosomes which will reproduce

into the next generation based on their fitness values, where individuals with higher fitness are more likely to be selected. In *roulette-wheel selection*, individuals are selected with a probability that is linearly proportional to their fitness. In *elitist selection*, the individual with the highest fitness is automatically selected.

The selection chromosomes then reproduce to create the next-generation population. One operation used in reproduction is *crossover*, where segments of two chromosomes are combined to create a new chromosome. The offspring chromosomes may also undergo a process of *mutation*, where each gene has its value changed with some probability, in order to explore the problem space.

This process is repeated until some stopping condition is met, typically after a certain number of generations.

Figure 1.1: A high-level view of a genetic algorithm.

1.2 Artificial Neurons

Artificial neurons are computational elements inspired by the biological neurons found in nature. An artificial neuron is comprised of a number of weighted input connections, an activation function which implements a response to input, and a number of output connections.

Figure 1.2: An artificial neuron.

Figure 1.2 depicts an artificial neuron with I inputs, one output, and an activation function f_{AN}. The neuron receives a vector of $I + 1$ input signals, $\mathbf{a} = (a_1, a_2, \ldots, a_I, a_{I+1})$, where a_{I+1} is known as a *bias unit* and always has a value of -1. The input vector is modulated by a weight vector, $\mathbf{w} = (w_1, w_2, \ldots, w_I, w_{I+1})$, where each weight w_i modulates the input signal a_i. The neuron computes the net input as the sum of the weighted input signals, giving

$$net = \sum_{i=1}^{I+1} a_i w_i.$$

The output signal, o, is then computed by applying the activation function to the weighted input, so that $o = f_{AN}(net)$.

A common choice for f_{AN} is the sigmoid function:

$$f_{AN}(net) = \frac{1}{1 + e^{-\lambda net}}$$

where the parameter λ influences the steepness of the function, but usually $\lambda = 1$.

Artificial neurons can be used to compute linearly separable functions without error. This means that, for such a function, there exists at least one threshold value such that the neuron can separate the space of I-dimensional input vectors which produce an above-threshold output from those which produce a below-threshold output by an I-dimensional hyperplane. This threshold is determined by the bias weight w_{I+1}, meaning that the neuron can be used to separate the input vectors for which $net > 0$ from the input vectors for which $net \leq 0$.

Learning is a technique by which the weights of an artificial neuron are updated to realize functions given by data. One type of learning is known as *supervised learning*, where the neuron is provided with a data set, known as the *training set*, consisting of training patterns of input vectors with associated target outputs. The weights of the neuron are then adjusted until the error between the actual outputs of the neuron and the target outputs in the patterns is minimized.

An example of a supervised learning rule is the so-called *delta rule*. The delta rule requires the definition of an error function, $\mathcal{E}$, to measure the neuron's error in approximating training targets. The sum of squared errors is usually used, given by

$$\mathcal{E} = \sum_{p=1}^{P_T}(t_p - o_p)^2$$

where t_p and o_p are the target and actual output for the pth pattern, and P_T is the total number of patterns in the training set.

Given a single training pattern, weights are updated using

$$w_i(t) = w_i(t-1) + \Delta w_i(t)$$

with

$$\Delta w_i(t) = \eta(-\frac{\partial \mathcal{E}}{\partial w_i})$$

where

$$\frac{\partial \mathcal{E}}{\partial w_i} = -2(t_p - o_p)\frac{\partial f_{AN}}{\partial net_p}a_{i,p}$$

and η is a constant known as the learning rate, net_p is the net input for pattern p, and $a_{i,p}$ is the ith input signal in pattern p.

The delta rule requires that its activation function f_{AN} is continuous and differentiable; this is so the gradient of the error can be followed toward lower values, giving the direction in which the weights should be updated. If we use the sigmoid function for f_{AN} then

$$\frac{\partial f_{AN}}{\partial net_p} = o_p(1 - o_p)$$

giving

$$\frac{\partial \mathcal{E}}{\partial w_i} = -2(t_p - o_p)o_p(1 - o_p)a_{i,p}$$

Artificial neurons are generally connected together into artificial neural networks to perform more sophisticated computation; however, this work is based closely on the work of Chalmers which considered single neurons in isolation [1].

1.3   Neuroevolution

Any feature of an artificial neuron, such as connection weights, activation functions, learning rules, and input features, can be subjected to an evolutionary process after being string-encoded [8]. However, this work will focus on the evolution of connection weights and learning rules.

The connection weights of a non-learning neuron can be evolved as the fixed response of that neuron to input. This is analogous to the evolution of instincts or what we think of as "hard-wired" behavior.

Alternately, the initial connection weights of a learning neuron can be evolved as the initial behavior which will be modified by the neuron's learning rule. This can be beneficial for evolving passably fit initial behavior that is then fine-tuned by learning but it can also be the case that the values of the initial weights are not adaptive and weights are evolved that are effectively random.

Learning rules can be evolved stand-alone by applying them to neurons initialized with random weights in the fitness function. In this situation the values of the initial weights do not matter because the idea is to evolve learning rules that will converge on a solution that is independent of the values of the starting weights.

The initial connection weights and the learning rules that modify them can be evolved simultaneously. In this situation, instincts, learning, or a combination of both can be evolved depending on the environment. This work will examine the outcomes of the simultaneous evolution of connection weights and learning rules under different conditions, whereas Chalmers examined only the possibility of evolving learning rules.

1.4 The Evolution of Learning

The evolution of learning in artificial neural networks is a topic of great interest to the research community, and while many researches have investigated many different approaches to the evolution of learning, many open questions still remain [6].

Evolution and learning are both forms of adaptation. Evolution takes place across generations whereas learning takes places within an individual's lifetime. Evolution can produce adaptive behaviors such as learning or non-adaptive behaviors such as instincts.

If an environment is sufficiently diverse or dynamic, instinctual behavior is not reasonably adequate for good task performance; this is because such an environment will invariably present novel tasks to which the instinctual behavior is not well-adapted.

Evolved learning behavior that is scalable and adaptive to unknown or changing situations can be seen in countless species in nature [5].

Chalmers [1] evolved learning rules by encoding the parameters of a template formula as strings, demonstrating that the delta rule for supervised learning can be evolved in certain situations for artificial neurons in static environments.

Shah [5] similarly demonstrated the evolution of reinforcement learning in changing environments.

1.5 Nurturing

Nurturing is defined as "the contribution of time, energy, or other resources by one individual to the expected physical, mental, social, or other development of another individual with which it has an ongoing relationship" [7].

Nurturing is prevalent in the biological world and can be an important contributing factor to the evolution of learning [7] [2], an idea which has recently been echoed by others [6].

Nurturing itself can be either instinctive or learned. Leonce, Hoke, and Hougen [4] demonstrated the evolution of instinctive nurturing in robots.

Nurturing can cover the initial costs of a learner and the resulting benefits of learning can be paid forward to the next generation. In this way, nurturing can promote the evolution of learning.

Eskridge & Hougen [2] conducted experiments involving food patch estimation in uncertain environments, and their results demonstrated that nurturing as both social learning and safe exploration can promote the evolution of learning. Shah [5] demonstrated that nurturing as task simplification can promote the evolution of reinforcement learning in changing environments.

This work explores nurturing as safe exploration in that individuals do not incur

fitness penalties for mistakes made during the learning process.

1.6 Contents of the Thesis

The rest of this thesis is organized as follows: Hypotheses about how nurturing affects the evolution of learning and instincts are detailed in **Chapter 2**. Operational definitions for terms such as "nurturing" and "instincts" are introduced in **Chapter 3**. Procedures for the experiments examining the effect of nurturing on the evolution of learning and instincts are detailed in **Chapter 4**. The experimental results, which demonstrate that nurturing facilitates the evolution of learning, are outlined in **Chapter 5**. The conclusion that nurturing facilitates the evolution of learning is described in **Chapter 6**. Finally, ideas for exploring the interaction between nurturing and the Baldwin effect in future work are described in **Chapter 7**.

Chapter 2

Hypotheses

Three factors are considered with respect to their effects on the evolution of learning: the number of tasks, the presence or absence of nurturing, and the possibility of evolving instincts.

Chalmers demonstrated that the quality of evolved generalized learning mechanisms is proportional to the number of tasks in the evolutionary environment. When there are few tasks in the evolutionary environment, high-quality generalized learning is not expected to evolve.

Given environments like those used by Chalmers, where instincts cannot be evolved, the addition of nurturing alone is expected to have little impact on the evolution of generalized learning because learning is still the only viable strategy in that setup. Therefore, it is expected that the quality of evolved generalized learning mechanisms will remain proportional to the number of tasks and the quality of the learning evolved will be no greater with nurturing than without.

H$_1$: When only learning rules are evolved, the quality of generalized learning evolved will be proportional to the number of tasks in the evolutionary environment, both with and without nurturing.

H$_2$: When only learning rules are evolved, the quality of generalized learning evolved will be equal with nurturing and without.

The introduction of the ability to evolve initial network weights alongside learning rules allows for the evolution of "instincts" as an alternate strategy to learning. It is expected that higher-quality instincts will evolve for small numbers of evolutionary

tasks, but higher-quality learning will evolve as the number of evolutionary tasks increases. This will be because there will be a smaller number of instinct genes than learning genes to "tune" to a correct solution for small number of tasks, but because the number of genes required to represent instincts will increase as the number of tasks increases the number of instinct genes will become much larger than the number of learning genes as the number of tasks increases.

H_3: When both initial weights and learning rules are evolved, the quality of instincts evolved will be inversely proportional to the number of tasks in the evolutionary environment, both with and without nurturing.

H_4: When both initial weights and learning rules are evolved, the quality of generalized learning evolved will be proportional to the number of tasks in the evolutionary environment, both with and without nurturing.

It is expected that higher-quality generalized learning will evolve in the nurturing condition than in the non-nurturing condition—and higher-quality instincts will evolve in the non-nurturing condition than in the nurturing condition—because of the costs associated with the learning process in the non-nurturing condition; individuals tend to have low fitness early in their life before they have had the opportunity to learn correct behavior, but in the nurturing condition individuals do not incur fitness penalties for mistakes made early in life.

H_5: When both initial weights and learning rules are evolved, higher-quality instincts will evolve without nurturing than with.

H_6: When both initial weights and learning rules are evolved, higher-quality generalized learning will evolve with nurturing than without.

Chapter 3

Operational Definitions

An *individual* is a candidate solution to the genetic algorithm's fitness function whose characteristics are represented by a chromosome ("Computational Intelligence", pg. 129). In this work, individuals are manifest as artificial neurons.

A *task* is a set of input-output patterns, where for each pattern in the task an individual is evaluated on its ability to produce the pattern output when activated with the corresponding pattern inputs.

An *environment* is a set of tasks by which individuals are assessed.

An individual's *lifetime* is the period during which it is manifest as an artificial neuron and assessed by the fitness function. Specifically, an individual's lifetime consists of a supervised learning period followed by a final evaluation.

Instincts are to be represented by the genetically encoded initial weights of a neuron. The tasks are independent, so there will need to be a separate set of instinct weights for each task.

Learning is to be represented by the genetically encoded learning rule that is applied to a neuron throughout its lifetime.

For evolved neurons where both initial weights and learning rules are genetically encoded, instinct quality is measured by removing the learning rule and evaluating the fitness of the neuron in the environment in which it was evolved. For the same neurons, *learning ability* is measured by replacing the genetically encoded initial weights with random initial weights and evaluating the fitness of the neuron in the environment in which it was evolved; *generalized learning ability* is measured by evaluating the

fitness of this instinct-removed neuron in an environment different from the one in which it was evolved.

By default, an individual's lifetime fitness is the average of all fitness evaluations made during learning—when the individual is prone to making mistakes—and the final fitness evaluation that follows learning; this is the "non-nurturing" condition. In the "nurturing" condition, then, an individual's lifetime fitness is simply the result of the final fitness evaluation taken after the individual has had an opportunity to learn correct behavior.

Chapter 4

Procedure

4.1 Genetic Coding of Learning Mechanisms

The weight update rule for an artificial neuron needs to be based on the components of that neuron which are, as covered in Section 1.2, a_j the activation of the input unit j, o_i the activation of the output unit i, t_i the training signal on output unit i, and w_{ij} the current value of the connection strength from input j to output i.

Following Chalmers (1990), the genome encodes a function F such that $\Delta w_{ij} = F(a_j, o_i, t_i, w_{ij})$, where F is a linear function of its four parameters and their six pairwise products. Thus F is determined by specifying ten coefficients.

The genome encodes these ten coefficients, as well as an eleventh "scale" parameter. That is,

$$\Delta w_{ij} = k_0(k_1 w_{ij} + k_2 a_j + k_3 o_i + k_4 t_i + k_5 w_{ij} a_j + k_6 w_{ij} o_i + k_7 w_{ij} t_i + k_8 a_j o_i + k_9 a_j t_i + k_{10} o_i t_i),$$

where k_0 to k_{10} are the encoded coefficients.

The portion of the genome which encodes Δw_{ij} consists of 35 bits. The first five bits encode the scale parameter k_0 such that it can represent the values 0, $\pm 1/256$, $\pm 1/128$, ..., ± 32, ± 64, via exponential encoding. The first bit encodes the sign of k_0 (0: negative, 1: positive), and the next four bits encode the magnitude. If these four bits are interpreted as an integer j between 0 and 15, we have

$$|k_0| = \begin{cases} 0 & \text{if } j = 0 \\ 2^{j-9} & \text{if } j = 1, ..., 15. \end{cases}$$

The other 30 bits encode the other ten coefficients in groups of three. The first bit of each group expresses the sign, and the other two bits express a magnitude of 0, 1, 2, or 4 via a similar exponential encoding. If we interpret these two bits as an integer j between 0 and 3, then

$$|k_i| = \begin{cases} 0 & \text{if } j = 0 \\ 2^{j-1} & \text{if } j = 1, 2, 3. \end{cases}$$

4.2   Genetic Coding of Initial Weights

Initial weights may be evolved alongside learning rules. It would not be meaningful for each chromosome to encode a single set of weights to be used on all tasks, so instead each chromosome simultaneously encodes a distinct set of weights for each task in the evolutionary environment. The weights to be applied to each evolutionary task are encoded in distinct, consistent regions of each chromosome.

Initial weights are encoded using 3 bits each. The first bit is the sign of the weight, and the other two bits express a magnitude of 0, $\frac{1}{8}$, $\frac{1}{4}$, or $\frac{1}{2}$ via an exponential encoding. If we interpret the remaining two bits as an integer j between 0 and 3, then

$$|k_i| = \begin{cases} 0 & \text{if } j = 0 \\ 2^{j-4} & \text{if } j = 1, 2, 3. \end{cases}$$

For each task the number of weights encoded is equal to the number of inputs in that task plus one bias weight.

4.3 Evaluation of Fitness

There are two conditions of evaluation: the "nurturing" case and the "non-nurturing" case. During evaluation, each network is first trained for 10 epochs using its learning rule. Following this, the network is evaluated on the same tasks. In the "nurturing" case the individual's fitness is simply the result of the evaluation, whereas in the "non-nurturing" case the individual's fitness is the average of evaluation during all training epochs and the final evaluation.

Fitness evaluation for the nurturing case:

(1) Create a network with the appropriate number of input units for the task and a single output unit.

(2) Initialize the connection strengths of the network using the values encoded in the chromosome.

(3) For 10 epochs, cycle through the training exemplars for the task, where for each exemplar:

(3a) propagate input values through the system, yielding output values; then

(3b) adjust the weights of the system according to the formula specified by the weight update rule, on the basis of inputs, output, training signal, and current weights.

(4) At the end of this process, fitness on the task is measured by testing the network on all training exemplars, and dividing the total error by the number of exemplars, and subtracting from 1. This yields a fitness value between 0 and 1.

Fitness evaluation for the non-nurturing case:

(1) Create a network with the appropriate number of input units for the task and a single output unit.

(2) Initialize the connection strengths of the network using the values encoded in the chromosome.

(3) For 10 epochs, cycle through the training exemplars for the task, where for

each exemplar:

(3a) test the network on the exemplar and measure the error in the network output; then

(3b) propagate input values through the system, yielding output values, then adjust the weights of the system according to the formula specified by the weight update rule, on the basis of inputs, output, training signal, and current weights.

(4) At the end of this process, test the network on all training exemplars and divide the total error of this test and all tests from step 3a by the total number of tests that occurred, and subtracting from 1. This yields a fitness value between 0 and 1.

Fitness of the chromosome is obtained by evaluating its performance on each of the (typically 20) tasks, and taking the mean fitness over all tasks. In this way every chromosome is assigned a fitness between 0 and 1.

4.4 Parameters of the Genetic Algorithm

A fixed population size of 40 is used in all experiments. Following the fitness evaluation process described in Section 4.3, a new population of 40 individuals is created as follows:

(1) Elitist selection is applied, meaning that an exact copy of the individual with the highest fitness in the previous population is inserted into the new population.

(2) Roulette-wheel selection (where an individual's probability of being selected is linearly proportional to its fitness) is used to select 39 individuals from the previous population with replacement (so an individual can be selected more than once), then:

(2a) the first 32 selected individuals are reproduced by gene-wise two-point crossover in pairs to create 32 new individuals which are inserted into the new population;

(2b) the remaining 7 selected individuals are reproduced by cloning to create 7

new individuals which are inserted into the new population.

(3) Each individual in the new population is mutated such that each bit in each chromosome has a 1% chance of changing.

This cycle of fitness-evaluation and reproduction is repeated for 4000 generations.

Population size: 40 Crossover rate: 0.8 Mutation rate: 0.01 Elitism factor: 1 Number of generations: 4000

4.5   Post-Evolutionary Evaluation

Before each evolutionary run, 10 tasks are selected at random from the task pool and designated as "test tasks," and up to 20 tasks are selected at random from the remaining 20 tasks and designated as "evolutionary tasks." New evolutionary tasks and test tasks are selected for each repetition of an evolutionary run and between evolutionary runs nurturing condition and non-nurturing condition . After each evolutionary run, the chromosome with the highest fitness score in the history of the populations in that run is identified. From this chromosome, the learning rule bits are isolated and evaluated on the test tasks (using both the nurturing and non-nurturing conditions) and the bits encoding the initial weights are isolated and evaluated on the evolutionary tasks. Each evolutionary run is repeated 30 times.

Chapter 5

Results

This chapter presents the results of all experiments performed.

5.1 Results of Initial Evolutionary Runs

This section presents the results of experiments where learning rules can be evolved but initial weights cannot, performed under both the nurturing and non-nurturing conditions.

Figure 5.1: The maximum fitness for populations evolved using 20 tasks under the nurturing and non-nurturing conditions.

Figure 5.2: The average generalized learning capabilities for the best individuals evolved under the nurturing and non-nurturing conditions.

5.2 Intra-Learning and Post-Learning Fitness Test

This section and the following sections present the results of experiments where learning rules and initial weights can be evolved, performed under both the nurturing and non-nurturing conditions.

The individual with the highest fitness in each evolutionary run is re-evaluated on the evolutionary tasks from that run using both the nurturing and non-nurturing conditions (recall that it is evaluated using only one of these conditions during evolution).

The fitness test for the non-nurturing condition can be seen as testing how much

fitness an individual would collect on average during the learning process (and one final evaluation immediately afterward); as such, it will be referred to as the "intra-learning fitness test."

Similarly, the fitness test for the nurturing condition can be seen as testing how much fitness an individual would collect on average after learning as been completed; it will be referred to as the "post-learning fitness test."

Figure 5.3: Intra-learning/non-nurturing fitness test on evolutionary tasks for individuals evolved under the nurturing and non-nurturing conditions.

Figure 5.4: Post-learning/nurturing fitness test on evolutionary tasks for individuals evolved under the nurturing and non-nurturing conditions.

5.3 Pre-Learning Test

The portion of the chromosome that encodes the initial weights is isolated and evaluated on the evolutionary tasks by initializing a network with the encoded weights, with the intention of determining the contribution to the fitness of the whole chromosome of just the portion of the chromosome that encodes the initial weights. In other words, this is intended to test the instincts of the individual in the absence of learning.

The inverse-linear correlation between the pre-learning fitness and the number of evolutionary tasks is statistically significant for both the nurturing condition (Pear-

son correlation, $r = -0.6324$, $p = 2.4733 * 10^{-68}$) and the non-nurturing condition (Pearson correlation, $r = -0.8331$, $p = 6.4082 * 10^{-156}$).
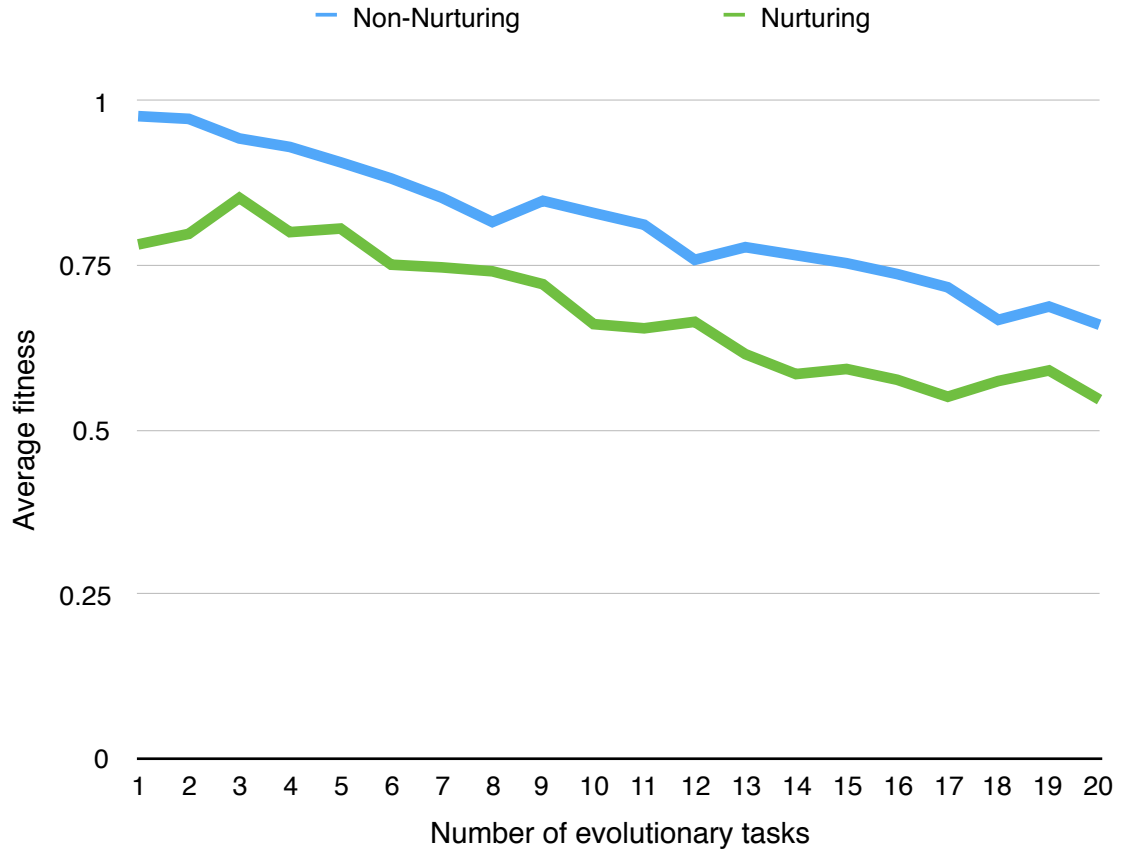

Figure 5.5: Pre-learning/instinct test on evolutionary tasks for individuals evolved under the nurturing and non-nurturing conditions.

5.4 Learning Improvement

To determine the average amount of useful learning that evolves for a given number of tasks we can examine the difference between the post-learning fitness and the pre-learning (instinctive) fitness, as shown in Figure 5.6. This is because the pre-learning fitness is the individual's fitness before it learns, whereas the post-learning fitness is an individual's fitness after it learns.

Figure 5.6: The difference between evolutionary fitness and network fitness for individuals evolved under the nurturing and non-nurturing conditions.

5.5 Weight-Generalization Learning Test

The portion of the chromosome that encodes the learning rule is isolated and evaluated on the evolutionary tasks by initializing a network with random weights and the encoded learning rule, with the intention of determining the contribution of the portion of the chromosome that encodes the learning rule to the fitness of the whole chromosome.

Figure 5.7: Random-start, intra-learning/non-nurturing learning test on evolutionary tasks for individuals evolved under the nurturing and non-nurturing conditions.

Figure 5.8: Random-start, post-learning/nurturing learning test on evolutionary tasks for individuals evolved under the nurturing and non-nurturing conditions.

5.6   Task Generalization Learning Test

The portion of the chromosome which encodes the learning rule is isolated and evaluated on the test tasks by initializing a network with random weights and the encoded learning rule, with the intention of determining the capacity for generalized learning that was evolved. The differences between the results for the nurturing and non-nurturing conditions are statistically significant (randomized ANOVA, algorithm $p < 0.001$, interaction $p < 0.001$).
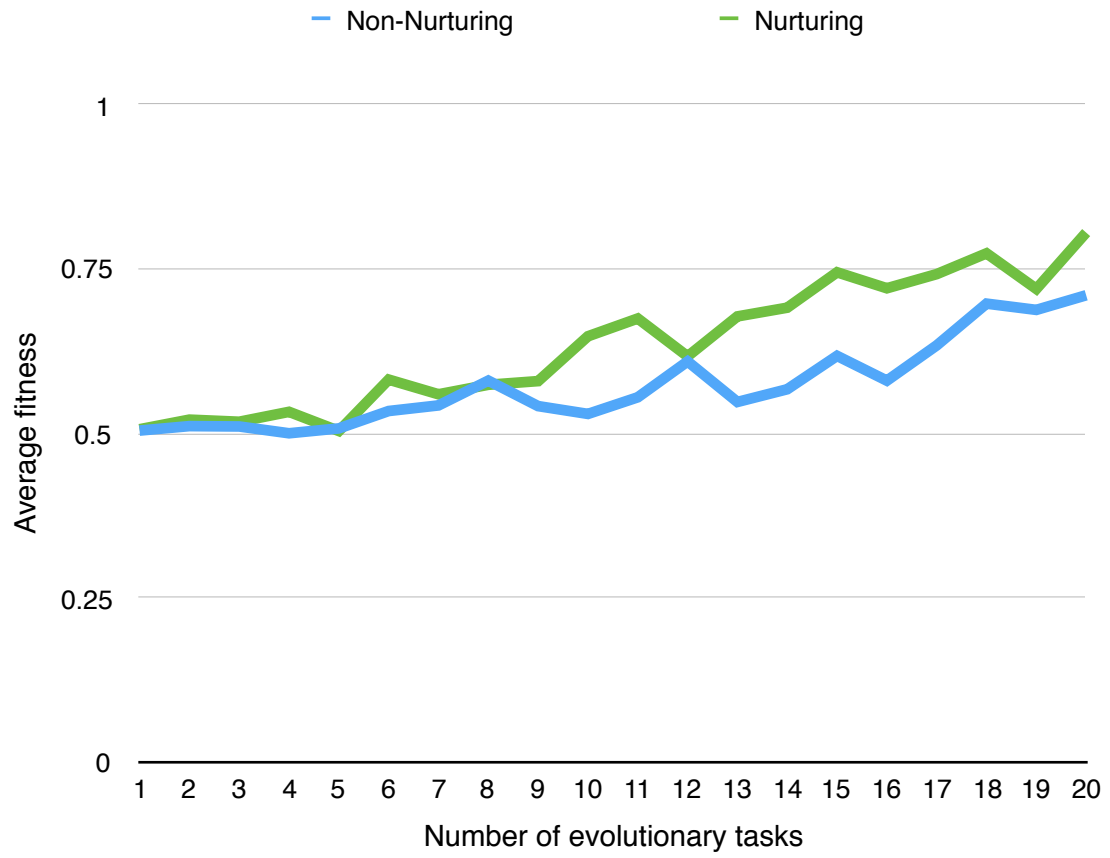
Figure 5.9: Non-nurturing generalization test on test tasks for individuals evolved under the nurturing and non-nurturing conditions.

Figure 5.10: Nurturing generalization test on test tasks for individuals evolved under the nurturing and non-nurturing conditions.

Chapter 6

Discussion

6.1 Initial Evolutionary Runs

Figure 5.1 demonstrates the increase of maximum fitness with the progression of generations, as well as the increasing adaptation of the individuals to their environment.

Figure 5.2 demonstrates that the generalized learning capabilities of evolved individuals increases with the number of evolutionary tasks.

These results serve as a re-confirmation of Chalmers's original findings [1].

6.2 Intra-Learning and Post-Learning Fitness Test

Under both conditions it is clear that the objective function becomes more difficult to satisfy as the number of evolutionary tasks increases.

Individuals evolved using the non-nurturing condition perform better when re-evaluated using the intra-learning fitness test/non-nurturing condition than when using the post-learning test/nurturing condition, and individuals evolved using the nurturing condition perform better when re-evaluated using the post-learning test/nurturing condition than when using the intra-learning test/non-nurturing condition.

6.3 Pre-Learning Test

Networks evolved using the non-nurturing condition performed better on this evaluation than networks evolved using the nurturing condition, suggesting that instincts

(initial weights) make a larger contribution to fitness in the non-nurturing case than in the nurturing case.

6.4   Learning Improvement

As shown in Figure 5.6, the improvement in fitness due to learning increases as the number of evolutionary tasks increases. This suggests that learning becomes an increasingly important evolutionary strategy as the environment becomes more varied.

6.5   Weight-Generalization Learning Test

In both evaluation conditions, the learning rule evolved under the nurturing condition performed better than the learning rule evolved under the non-nurturing condition, suggesting that the learning rule makes a larger contribution to fitness under the nurturing condition than it does under the non-nurturing condition.

6.6   Task Generalization Learning Test

In both evaluation conditions, the learning rule evolved under the nurturing condition performed better than the learning rule evolved under the non-nurturing condition by a statistically significant margin, suggesting that better generalized learning is more likely to evolve under the nurturing condition.

Chapter 7

Conclusions

The impact of varying the nurturing condition alone was insignificant; fitnesses measured under the nurturing condition were generally higher than those measured under the non-nurturing condition, but this is to be expected because of the additional fitness penalties imposed by the non-nurturing condition.

When instincts and learning were allowed to evolve alongside each other, it was found that instinct-based solutions were more likely to evolve for small numbers of evolutionary tasks, but that learning became more likely to evolve as the number of evolutionary tasks increased.

By varying the nurturing condition when instincts were allowed to evolve alongside learning, it was found that higher-quality generalized learning was expected to evolve under the nurturing condition than under the non-nurturing condition.

Chapter 8

Future Work

8.1 The Baldwin Effect

Baldwin (1896) argued that learning accelerates evolution because suboptimal individuals increase their baseline fitness by acquiring more adaptive characteristics during life. Lifetime learning often involves a cost because the individual may be at risk at an early stage of its life or it may modify its behavior in ways that are not functional for its survival. Baldwin suggested that evolution tends to select individuals that are born with some of the useful features that would otherwise be learned. However, a consequence of this effect is that learned features may gradually become assimilated into the genotype and reduce the role of learning; this is not necessarily a good effect if it is desirable for the individuals to retain the ability to learn over evolutionary time. [3]

Nurturing can reduce the evolutionary costs of learning and thus may be able to inhibit the Baldwin effect. Inhibiting the Baldwin effect may result in evolved individuals that retain greater learning capabilities over evolutionarily significant periods of time.

Bibliography

[1] David J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In Morgan Kaufmann, editor, *Proceedings of the 1990 Connectionist Models Summer School*, pages 81–90. Morgan Kaufmann, 1990.

[2] Brent E. Eskridge and Dean F. Hougen. Nurturing promotes the evolution of learning in uncertain environments. In *IEEE International Conference on Development and Learning / Epigenetic Robotics*, 2012.

[3] Dario Floreano and Claudio Mattiussi. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. The MIT Press, 2008.

[4] Armand Leonce, Bryan Hoke, and Dean F. Hougen. Evolution of robot-to-robot nurturing and nurturability. In *IEEE International Conference on Development and Learning / Epigenetic Robotics*, 2012.

[5] Syed Naveed Hussain Shah. *Nurturing Promotes the Evolution of Learning in Changing Environments*. PhD thesis, University of Oklahoma, 2015.

[6] Andrea Soltoggio, Kenneth O. Stanley, and Sebastian Risi. Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks. arXiv preprint arXiv:1703.10371, 2017.

[7] Mark Woehrer, Dean F. Hougen, Ingo Schlupp, and Brent E. Eskridge. Robot-to-robot nurturing: Definition, importance, and evolution. In *IEEE International Conference on Development and Learning / Epigenetic Robotics*, 2012.

[8] Xin Yao. Evolving artificial neural networks. In *Processings of the IEEE*, volume 87, pages 1423–1447, 1999.