Simulations combining evolution and learning

Michael Littman
Department of Computer Science
Brown University
Providence, RI
and
Bellcore
Morristown, NJ

November 8, 1995

1 Introduction

Nature has devised innumerable ways to endow individual organisms with adaptive behavior. Many of these processes can be classified as being part of *evolution*, changes in the genome that alter inborn behavior; or *learning*, changes in an individual's behavior as a result of interactions with its environment. Researchers in the fields of genetic algorithms and artificial neural networks use caricatures of evolution and learning to solve difficult optimization problems.

The behavior of most real organisms is a consequence of learning and evolution in concert, so it is natural to assume that there is computational advantage in their combination. However, the formidable complexity of the two processes is multiplied when we consider their interactions.

This paper addresses the issue of how computational versions of learning and evolution have been made to interact in simulated systems. It examines various benefits of such combinations and details how supervised learning, reinforcement learning, and unsupervised learning can be adapted to fit into an evolutionary framework.

2 Evolution and Learning

We'll use a simple simulation framework for examining computational analogues of evolution and learning. At the top level is a genetic algorithm (GA) [9, 7] that maintains a population of individuals specified by their genomes (fixed-length bit strings). The genetic algorithm collects *fitness* values for individual genomes by evaluating them according to some prespecified function and then selectively replicates and recombines genomes on the basis of this fitness. This constitutes the "evolutionary" level.

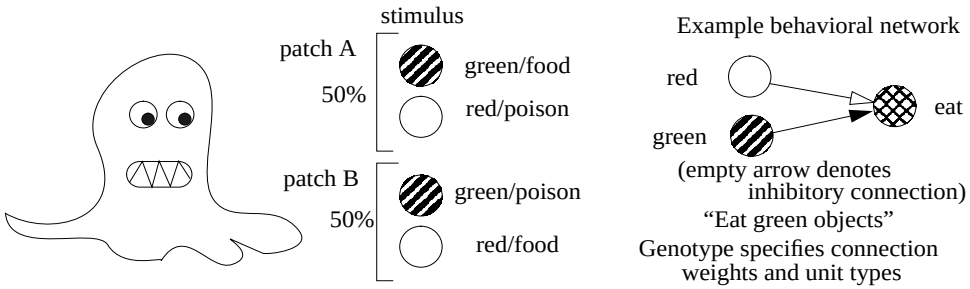

Figure 1: The generic associative eater environment.

"Learning" takes place during the fitness evaluation of the genomes. The bits of the gene string are used to generate a neural network, which then undergoes a series of interactions with a simulated environment. These interactions may result in changes to the weights of the individual's network according to a prespecified learning algorithm. The interactions constitute the *lifetime* of the individual and last some fixed number of steps. The network learns insofar as the behavioral mapping it specifies changes over time. The fitness of the genome is then determined by some measure of performance of the individual over its entire lifetime. Although an individual's network might change a great deal during its lifetime, these changes are *not* incorporated back into the genome but instead "die" with the individual.

As a running example, we consider a family of environments based on Todd and Miller's associative eater environment [17]. The scenario is depicted in Figure 1 and can be imagined as an underwater realm in which individual offspring are born into feeding patches, where they live out their entire lives. Feeding consists of deciding whether to consume substances that float past; these substances come in two forms, food and poison, and two colors, red and green. The association between color and substance is a function of the individual's feeding patch which cannot be known directly. On each step, each individual is presented with a substance (food or poison, red or green) and must decide, on the basis of sensory cues and experience alone, whether to consume the substance or to let it pass. Fitness is computed as the amount of food consumed minus the amount of poison consumed over a fixed lifetime of ℓ steps. Since half of the substances are food, and half are poison, fitness scores vary between $-\ell/2$ and $+\ell/2$.

As described here, no feedback is given to individuals that could be used to disambiguate between food and poison. In later sections, within-lifetime feedback about the consequences of eating food and poison will be the main way that the problem is manipulated to illustrate different types of learning.

The motivation for describing this model is simple—it is one of the simplest models that has been proposed that includes both evolution and learning components. Throughout the paper it is used to illustrate different approaches to studying how learning and evolution can interact.

2.1 Evolution finds optimal behavior

The behavior of each individual is controlled by a simple neural network encoded in the individual's genes. If it were the case that poison was red and food green in all feeding patches, it is easy to imagine that individuals would evolve with a simple network such as the one shown in the right half of Figure 1. The behavior of this network is to consume any green objects that float by (the filled-in arrow denotes a positive connection) and to ignore all red objects (the empty arrow denotes a negative connection: the eating unit will not activate).

Given the assumption that the substance to color mapping is constant, evolution can find networks that display optimal behavior. An experiment supporting this claim is provided in Todd and Miller's paper (their noise-free case).

2.2 Learning finds optimal behavior

In a slightly different scenario, individuals are born into one of two feeding patches uniformly at random. In patch A, food is green and poison red. In patch B it is the reverse. Individuals remain in the patch in which they were born, so for a single individual, the colors associated with food and poison are constant for its lifetime. A population of individuals with networks such as the one shown in the right half of Figure 1 that exhibit fixed behavior will not be optimal, since half of all individuals will be born into patch A and receive a fitness score of $+\ell/2$ and the other half will be born into patch B and receive $-\ell/2$ for an average fitness of 0.

Individuals that learn have the possibility of recognizing and exploiting structure in their surroundings and thus receiving fitness scores above 0 regardless of which feeding patch they are born into. However, the environment as described in Section 2 offers no opportunity for learning since it is impossible for an individual to distinguish food from poison. The environment must provide additional input for learning to be possible.

The machine learning community has identified three broad classes of learning: supervised, reinforcement, and unsupervised. They are distinguished by the amount and type of feedback information given to the learner. In supervised learning, the environment is assumed to provide the individual with direct and perfect information as to how it ought to behave in the situations it experiences (i.e., the right answer). The feedback signal in reinforcement learning is somewhat weaker in that the environment provides a single number, called a reinforcement signal, indicating the fitness consequence of performing the most recent action in the current situation. Unlike supervised learning, the individual must then experiment to find high fitness actions. In unsupervised learning, the feedback is weaker still and consists of latent structure in the input patterns themselves. Unsupervised learning algorithms can identify certain classes of structure which can then be used to make decisions.

Given any of these kinds of feedback, learning algorithms exist that can alter the weights of a neural network to find optimal behavior (e.g., backpropagation [14] for supervised learning, CRBP [2] for reinforcement learning, and Kohonen networks [10] for unsupervised learning) independent of the individual's feeding patch. These ref-

erences indicate that, given the correct network structure and feedback, learning is sufficient for finding optimal behavior.

2.3 Combination speeds adaptation

Although either process is capable of solving difficult behavioral optimization problems, combining evolution and learning can lead to faster adaptation than either alone.

As an extreme example, Hinton and Nowlan [8] showed that a problem that was extremely difficult for a simulated evolution system alone was solved fairly easily by a system that contained elements of both evolution and learning. The mechanism they examined is known as the Baldwin Effect [3, 15], the essence of which is that learning alters the search space in which evolution operates by smoothing spikes in the fitness landscape, thus speeding evolution.

The interaction between evolution and learning hinges on the structure of Hinton and Nowlan's simulation [8]. There, some aspects of the individual are genetically fixed while others are filled in by a random search process they call "learning." An individual lives for a fixed number of steps, and learning ceases whenever the optimal behavior is discovered. An individual's fitness is the amount of time during which it displays optimal behavior. In this framework, learning guides evolution by making near-perfect individuals receive high fitness scores since such individuals quickly adapt, via learning, to exhibit optimal behavior. Evolution accelerates learning by making individuals nearly perfect to start with.

Fitness is measured "online"; that is, what matters is not just the individual's final behavior but also how quickly it begins behaving well; thus, faster learning means higher fitness. This means that any genetically specified aspects of the learning process have the potential of serving as a source of guidance between the two processes, e.g., the initial weights of the neural network, the number of hidden units, and settings for learning rate and momentum parameters [4].

2.4 Combination solves more general problems

The preceding section described how the combination of evolution and learning can make for quicker adaptation. However, the sort of straightforward combination used in these cases can only be applied when either evolution or learning alone is sufficient to solve the problem. This section describes other ways of combining evolution and learning that can make it possible to relax this requirement and thus solve problems that neither process can solve alone.

For evolution alone to find optimal behavior, the objective should not change radically from generation to generation. For learning alone to find optimal behavior, individuals need a feedback signal. Both of these assumptions are violated in Todd and Miller's original associative eater problem, as each individual is born into one of two feeding patches at random and no direct feedback is given to individuals during their lifetimes.

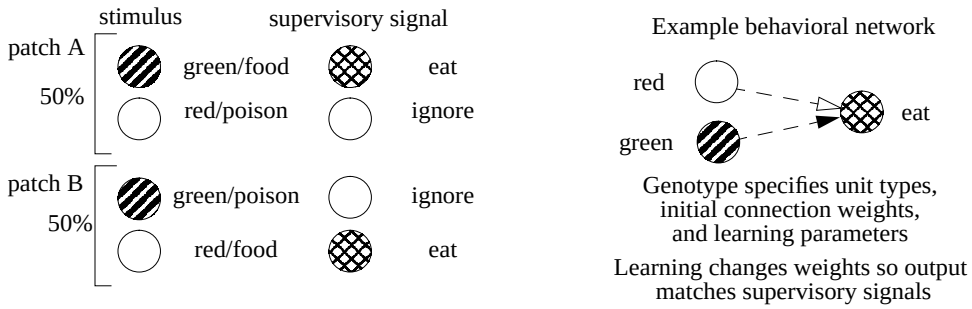

Figure 2: A supervised learning associative eater environment.

Todd and Miller show how to structure the simulation so that evolution can identify and take advantage of slowly changing, base-level properties of the environment while learning fills in the individual-specific details. This permits a hybrid system to solve a problem that neither evolution nor learning could solve alone.

3 Architectures for learning

One fundamental question must be answered before evolution and learning can be combined in a single system: from where does the feedback for learning come? In supervised learning and reinforcement learning, the feedback must be provided by the environment. In unsupervised learning, there is no direct feedback from the environment and instead the learner must exploit regularities in the *structure* of the environment to improve its behavior. Heuristic feedback methods, such as evolutionary reinforcement learning [1], and auto-teaching [12] are somewhat different as they involve learning via feedback signals, but the signals are tailored by evolution.

3.1 Supervised learning

Supervisory signals have been used in simulations that combine evolution and learning [4]. In supervised learning, the environment provides a training signal that informs the individual of the precise action that it *should* have performed in each circumstance. In the associative eater environment, a supervised learner must discover which attributes of the environment are needed to implement the correct mapping from color to action by providing the correct action output (eat/ignore) for each color input (green/red). In Figure 2, we show a network architecture for this type of simulation (dashed lines indicate connections that are changed by learning).

This kind of information-rich feedback ought to be sufficient for learning alone to find optimal behavior. As described in Section 2.3, however, evolution can speed the process along in principle by generating networks that can learn quickly.

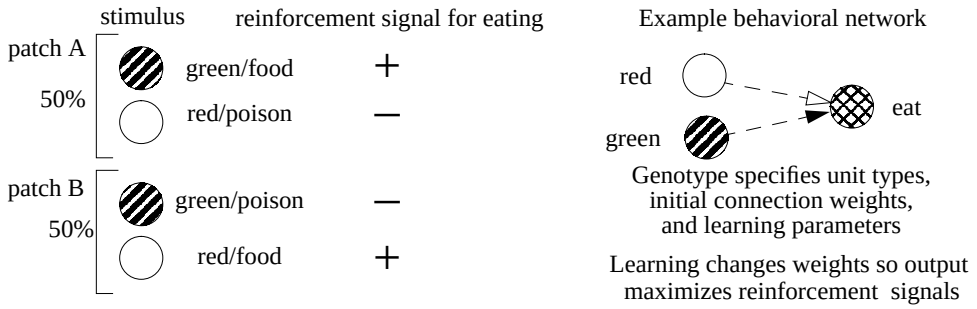

Figure 3: A reinforcement learning associative eater environment.

3.2 Reinforcement learning

Reinforcement signals provide less guidance than supervisory signals. Instead of giving the individual information as to the best action for each situation, in reinforcement learning only a single number is given which indicates how good the individual's last action was from the standpoint of achieving optimal fitness. By choosing actions that maximize the reinforcement signal, individuals can learn to behave optimally. In Figure 3, we summarize an associative eater environment in which a reinforcement signal is available.

For a behavioral repertoire consisting of two actions, reinforcement signals are practically identical to supervisory signals: being told that one of two actions is bad is the same as being told the correct action. When more actions are available, this type of feedback becomes weaker and weaker with respect to supervision. Nevertheless, in principle it is still sufficient for allowing any individual to learn optimal behavior given enough time (see, e.g., [2]).

3.3 Unsupervised learning

The learning methods described above use feedback signals that come directly from the definition of optimal behavior. That is, if the feedback signal says you did something wrong, you did something wrong. This sort of learning is implausible in natural systems since the only truly unambiguous signal is death. Other signals are simply correlated with good or bad behavior to some degree. For instance, because of regularities in the natural environment, the smell of food might be considered a hint that an individual's recent actions constitute successful behavior. Unsupervised learning methods allow the individual to learn and exploit this type of association.

In an unsupervised learning framework, individuals receive no direct guidance from the environment as to which actions are better than others. Instead, they must distinguish between good and bad behavior using associations between sensory cues.

Todd and Miller [17] described a simple scenario for the evolution of unsupervised learning, which they simulated to demonstrate its computational plausibility. Consider a version of the associative eater environment in which a second sensory input, smell, distinguishes food from poison consistently across food patches but with some probability of being misperceived. In particular, imagine food always smells sweet

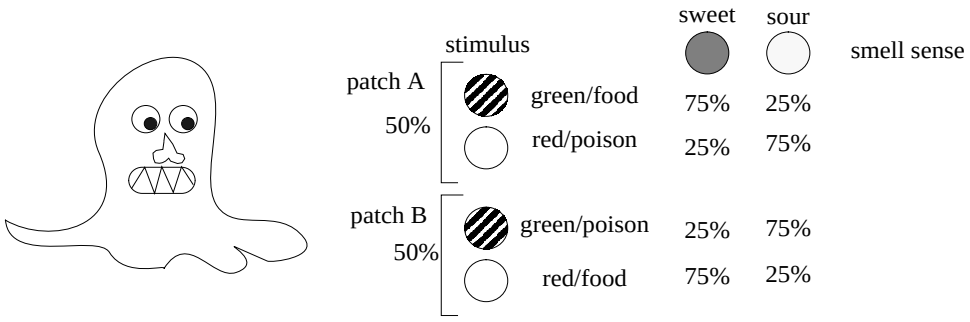

Figure 4: An unsupervised learning associative eater environment indicating correlations between color, smells, and object types.

and poison sour but on any given step, "turbulence in the water" might cause a smell to be improperly recognized with 25% probability. This information is summarized in Figure 4.

How can an individual maximize its fitness in this environment? Eating anything that smells sweet is a reasonable strategy and an individual with this fixed strategy can expect a fitness score of:

$$\frac{3}{4}\left(\frac{\ell}{2}\right) - \frac{1}{4}\left(\frac{\ell}{2}\right) = \frac{\ell}{4}.$$

Recall that ℓ represents the number of decisions an individual makes during its lifetime. Such an individual does not use its color sensors at all and makes every decision based on its smell sense.

However, if an individual can keep statistics on which color tends to be identified with which smell (in patch A, green substances smell sweet three quarters of the time, for instance), then it can reliably identify a substance as food or poison by its color and earn a fitness score of $(\ell - k)/2$ where k is the number of trials before the individual has learned the association between color and substance. If k is small with respect to ℓ (less than $\ell/2$) this strategy is an improvement.

Todd and Miller [17] describe an elegant network architecture that is capable of supporting this form of learning. The genome specifies a set of connections from sensor units to an eating unit. The left half of Figure 5 illustrates a simple network that eats anything that smells sweet. The right half of Figure 5 illustrates a feedforward network that can learn to eat substances of the proper color. The dashed line is a "changeable" connection that is strengthened if the green unit is correlated with eating and weakened if it is anticorrelated (i.e., it is a Hebbian connection). Consider the network's behavior in patch A (the argument is symmetrical for patch B). Since the eating unit has a direct connection from the sweet smell unit, the activity in these units will be correlated 75% of the time initially. After some number of trials that depends on the initial network weights, the dashed connection is strengthened to the point that the eating unit will fire on the basis of the green unit alone. Thus, the network learns the association between color and behavior and then begins to behave optimally using only sensory cues and no direct feedback.

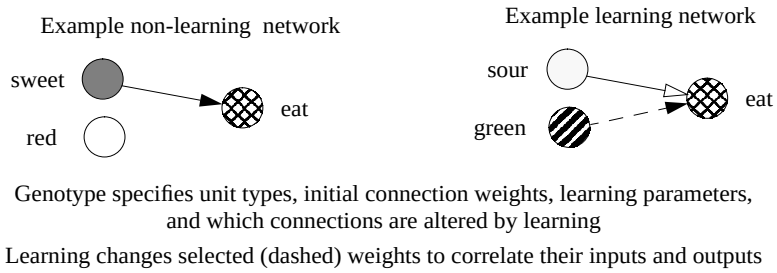

Figure 5: Networks for an unsupervised learning associative eater environment.

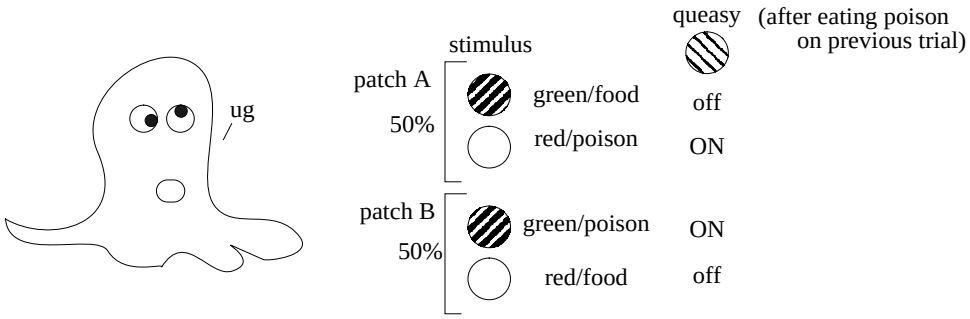

Figure 6: An evolutionary reinforcement learning associative eater environment.

Since the smell unit is positively correlated with whichever color predicts food, an unsupervised learning method can be used. However, it is the responsibility of evolution to recognize this relationship and to construct the proper network for exploiting it. Without the learning mechanism, individuals could not adapt to their food patch. Without evolution, the proper adaptation scheme could not be recognized. This is why a system that combines evolution and learning is able to achieve greater levels of fitness than one using either learning or evolution alone.

3.4 Evolutionary reinforcement learning

The unsupervised learning method described in the previous section uses a cross-generationally consistent input, namely sweet or sour smell, to determine which other input, namely red or green color, is a reliable indicator of appropriate behavior. As such, the smell units are being interpreted as a source of heuristic, though noisy, feedback. It is up to evolution to select how the sensory units will be used to generate these feedback signals.

Evolutionary reinforcement learning [1] uses the same fundamental insight. To apply this model of learning, we need to assume that an individual has an extended interaction with its environment and that it can perceive the effects of its own actions. Further, assume there is a reliable way of mapping the resulting sensations to the *utility* of the previous action. An environment that satisfies these assumptions can be called a "constant utility" environment [11] and evolutionary reinforcement learning can be used to find optimal behavior.

To illustrate this idea, here is another example modeled after the associative eater

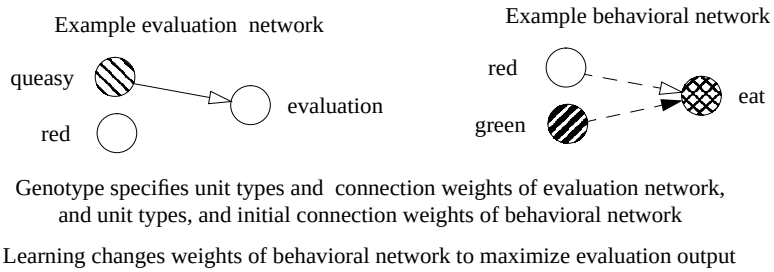

Figure 7: Evaluation and behavior networks for an evolutionary reinforcement learning associative eater environment.

(see Figure 6). An additional input, labeled "queasy," is activated each time the individual consumes poison. The mapping from sensations and actions to resulting sensations is *not* constant between individuals since eating something red produces queasiness in patch A but not patch B. However, the mapping from resulting sensations to utility or fitness *is* constant since queasiness at time t always means that the action at time $t-1$ was inappropriate. If the individual could map sensations to a feedback signal for the previous action, this could be used as a guide to learning.

A network architecture for this problem is illustrated in Figure 7. The behavior network (right) maps sensations to action, and the evaluation network (left) maps sensations to heuristic feedback signals. When the evaluation network gives a positive signal, a reinforcement learning algorithm [2] is used to modify the behavior network to make the *previous* action more likely given its previous inputs. Similarly, negative signals from the evaluation network make the previous action less likely. Littman and Ackley [11] studied a conceptually similar problem using this architecture and found that it worked quite effectively.

The primary influence of evolution here is in the specification of the fixed evaluation network, which generates the individual's heuristic feedback. Since individuals with good evaluation networks will tend to learn appropriate behaviors and earn high fitness, natural selection will create evaluation functions that do a good job of extracting the fitness consequences of actions. Evolution alone cannot solve this problem since individuals need to adapt to their own feeding patch. Learning alone is also inadequate because the connection between the "queasy" unit and fitness can only be detected across generations. Once again, evolution and learning can be made to work together to solve a problem that neither could solve alone.

4 Conclusion

From a computer science perspective, evolution and learning are two approaches to the problem of generating adaptive behavior. Nature gives us an indication that there is some advantage to combining the two and, indeed, simulation results have been consistent with this observation.

This paper provides two computational arguments that the combination of evolution and learning is superior to either alone. For problems where evolution and

learning are both capable of finding solutions independently, the combination can speed adaptation through the Baldwin Effect. For other problems, it is sometimes possible to combine evolution with unsupervised or heuristic learning methods to solve problems that neither learning nor evolution could be applied to alone.

Although the simulations reviewed in this paper suggest that there has been a great deal of work in the last few years combining evolution and learning in the form of genetic algorithms and neural networks, it seems we have barely scratched the surface. For example, Dennett [6] describes a powerful combination of evolution and learning in which an evolved "inner environment" is used to help generate individual behavior by *imagined* trial-and-error learning. Implementations of this idea have begun to appear in the last few years [16, 13] but not yet in an evolutionary setting.

Animals successfully employ forms of learning such as imprinting and imitation to develop survival skills. Individuals in our simulated worlds might also benefit from this form of learning, if honed properly by evolution. An even more exciting prospect is that, with the large number of types of learning that are possible, perhaps we should take a hint from natural evolution and let evolution choose the learning architecture as well (see, e.g., [5]).

One thing is certain. Each simulation extends our understanding of how intelligent behavior can arise from the interaction of "unintelligent" processes and also reinforces the idea that we have only begun to explore the question of how evolution and learning can be combined.

References

[1] David H. Ackley and Michael L. Littman. Interactions between learning and evolution. In C. Langton, C. Taylor, J. D. Farmer, and S. Ramussen, editors, *Artificial Life II: Santa Fe Institute Studies in the Sciences of Complexity*, volume 10, pages 487–509. Addison-Wesley, Redwood City, CA, 1991.

[2] David H. Ackley and Michael S. Littman. Learning from natural selection in an artificial environment. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1. Lawrence Erlbaum Associates, Washington DC, January 1990.

[3] J. M. Baldwin. A new factor in evolution. *American Naturalist*, 30:441–451, 536–553, 1896.

[4] Rik Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In C. Langton, C. Taylor, J. D. Farmer, and S. Ramussen, editors, *Artificial Life II: Santa Fe Institute Studies in the Sciences of Complexity*, volume 10, pages 511–547, Redwood City, CA, 1991. Addison-Wesley.

[5] D. J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, editors,

Proceedings of the 1990 Connectionist Models Summer School, San Mateo, CA, 1990. Morgan Kaufmann.

[6] D. C. Dennett. Why the law of effect won't go away. In D. C. Dennett, editor, *Brainstorms*, chapter 5, pages 71–89. Bradford Books (MIT Press), Cambridge, MA, 1981.

[7] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, MA, 1989.

[8] G. E. Hinton and S. J. Nowlan. How learning can guide evolution. *Complex Systems*, (1):495–502, 1987.

[9] John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[10] T. Kohonen. Clustering, taxonomy, and topological maps of patterns. In M. Lang, editor, *Proceedings of the Sixth international conference on pattern recognition*, pages 114–125, Silver Spring, MD, 1982. IEEE Computer Society Press.

[11] Michael L. Littman and David H. Ackley. Adaptation in constant utility non-stationary environments. In Rik K. Belew and Lashon Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 136–142, San Mateo, CA, 1991. Morgan Kaufmann.

[12] S. Nolfi and D. Parisi. Auto-teaching: Networks that develop their own teaching input. In J. L. Deneubourg, H. Bersini, S. Goss, G. Nicolis, and R. Dagonnier, editors, *Proceedings of the Second European Conference on Artificial Life (Brussels)*, pages 845–862, 1993.

[13] R. Riolo. Lookahead planning and latent learning in a classifier system. In J.A. Meyer and S. W. Wilson, editors, *From animals to animats: Proceedings of the first international conference on simulation of adaptive behavior*, pages 316–326, Cambridge, MA, 1991. MIT Press.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error backpropagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the microstructures of cognition. Volume 1: Foundations*, chapter 8. The MIT Press, Cambridge, MA, 1986.

[15] J. Schull. Are species intelligent? *Behavioral and Brain Sciences*, 13(1):61–73, 1990.

[16] Richard S. Sutton. Planning by incremental dynamic programming. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 353–357. Morgan Kaufmann, 1991.

[17] P. M. Todd and G. F. Miller. Exploring adaptive agency II: Simulating the evolution of associative learning. In J.-A. Meyer and S. W. Wilson, editors, *From animals to animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 306–315, Cambridge, MA, 1991. MIT Press/Bradford Books.