Incremental Development of Complex Behaviors through Automatic Construction of Sensory-motor Hierarchies

(Slightly extended version of that appearing in Proceedings of the Eighth International Workshop on Machine Learning)

Mark Ring
Department of Computer Sciences
University of Texas at Austin, Austin, TX 78712
ring@cs.utexas.edu

Abstract

This paper addresses the issue of continual, incremental development of behaviors in reactive agents. The reactive agents are neural-network based and use reinforcement learning techniques.

A continually developing system is one that is constantly capable of extending its repertoire of behaviors. An agent increases its repertoire of behaviors in order to increase its performance in and understanding of its environment. Continual development requires an unlimited growth potential; that is, it requires a system that can constantly augment current behaviors with new behaviors, perhaps using the current ones as a foundation for those that come next. It also requires a process for organizing behaviors in meaningful ways and a method for assigning credit properly to sequences of behaviors, where each behavior may itself be an arbitrarily long sequence.

The solution proposed here is hierarchical and bottom up. I introduce a new kind of neuron (termed a "bion"), whose characteristics permit it to be automatically constructed into sensory-motor hierarchies as determined by experience. The bion is being developed to resolve the problems of incremental growth, temporal history limitation, network organization, and credit assignment among component behaviors.

1 INTRODUCTION

The real world is characterized by a seemingly unlimited degree of detail and regularity. Regularity occurs at multiple scales and to various extents. The simplest organisms may find regularity such as correlations between scent and food, or movement and danger, which allow the organism to survive and succeed. More developed creatures recognize subtler concepts—such as the patterns of movement involved in mate selection rituals—and carry out complex behaviors such as hunting and stalking prey. Humans are capable of grasping ever more complicated concepts as they mature, beginning with a small set of organized behaviors, constantly augmented by more and more complex behaviors. The world supports this continual learning process by providing a never ending multitude of complexities.

1.1 PROBLEMS

In order to construct an agent capable of continual development in a complex environment, there are several issues that must be addressed, including the following:

1. Discovering which behaviors are worth learning.
2. Recording events for indefinitely long periods of time (the history limitation problem).
3. Organizing behaviors in useful and accessible ways.
4. Assigning credit to those behaviors responsible for achieving rewards.

First, each of the agent's behaviors should be useful to the agent in some way, and it would be especially desirable if the behavior was known to be beneficial *before* being acquired. (A behavior could be defined here as a sequence of changes an agent brings about in itself or its environment). Second, the behaviors must be capable of spanning arbitrarily large time periods, since regularities in the environment may exist at all time scales. In order to learn a sequence of individual events, an agent will need to keep track of information over time scales larger than that of the individual events. Third, with a potentially enormous number of behaviors, there will need to be a way of organizing them so that they can be used and accessed easily, regardless of their possible complexities. Fourth, during reinforcement learning, those behaviors which are instrumental in securing a reward should receive the reward.

1.2 SOLUTION

The solution I propose to each of these is as follows:

1. Build behavior hierarchies from the bottom up.
2. Construct each behavior such that it will occur over some time period.
3. Distinguish between the selection of a behavior, its execution, and its accomplishment.
4. Assign credit to the highest level selections made.

The first aspect of the solution is the bottom up construction of sensory-motor behavior hierarchies. After low-level behaviors are learned, the agent will use them together in different ways, constituting higher-level behaviors. Moving south, for example, could be a low-level behavior; moving West could be another. These could then be combined into a single higher-level behavior: Move South, then Move West. This process can continue to encompass ever larger groups of behavior.

Second, the history limitation problem can be solved if each behavior occupies some temporal span. Then, a higher-level behavior will occupy a time span proportional to the sum of the duration of its components. This can continue indefinitely to allow behaviors of any duration.

Third, a distinction must be drawn between the selection of a behavior—reflecting a decision made by the agent—and the subsequent execution of its components. This allows the agent to operate at the simplest level possible at all times, always making simple high-level decisions, even though the complexity entailed by these decisions might be very great. When the behavior has been accomplished, this fact should be signaled in such a way that successive decisions can be informed as to the events that have recently occurred.

Fourth, reinforcement should be credited to the *decisions* made, not to the details entailed by the decision. This "chunking" approach (Miller, 1956) to reinforcement learning allows the pairing of decisions instrumental to securing a reward with the reward itself *over arbitrarily large periods of time*. And it does this by utilizing the regularities the agent has already learned about the environment.

Take for example the task of learning to go down a street to a lamp-post and then to turn left. The agent first needs to move down the street, constantly sensing its environment until it arrives at the lamp-post. It then needs to study the object until it can be recognized as a lamp-post. Then it needs to turn left and proceed. Each of these three activities might require a large number of time-steps involving many sensations and many actions. If standard reinforcement techniques were used, such as described in Barto et al. (1983), discussed next, then successfully delivering the reward received at the end of the activity to the behaviors involved could require very many trials. This is because standard methods reinforce each time step of the process, and the reinforcement is applied more and more weakly farther from the reward.

On the other hand, if the agent had already learned and separately encapsulated the behaviors of moving down a street, recognizing a lamp-post, and turning left, then credit would need only be assigned to these three, high-level behaviors, no matter what their time scale.

2 TRADITIONAL DESIGNS

Feed-forward neural networks and recurrent neural networks are two traditional methods used to address learning problems in reactive agents. For example, the Adaptive Critic Element (ACE) (Barto et al., 1983), uses feed-forward networks. These networks map current environmental states directly into actions. The ACE is a useful credit assignment device, since it spreads reward back in time to the decisions that caused the reward to be received. Feed-forward networks, however, can only make decisions based on information from the current sensory data alone (or from a fixed past number of steps, if delay-lines are used). There is no way for arbitrarily long histories to be generated, nor for regularities in the environment to be captured.

Recurrent networks, on the other hand, allow information to be held for indefinite periods of time. Combined with the ACE they can be effective reinforcement learning machines (Schmidhuber, 1990a). However, they too suffer from (1) a difficulty in correlating behaviors with distant rewards, (2) network size limitations, and (3) an inability to modularize and incrementally augment behaviors.

Regarding (1), it has been shown that recurrent networks lose information very rapidly (Servan-Schreiber et al., 1988). Unless information is used for a decision soon after it becomes available, the network will not retain it. The ACE is therefore a useful addition to the recurrent network (cf. Schmidhuber), since it seems to transport the effect of a distant reward to the decision(s) that caused it. But the reward must pass backwards over a number of previous states linearly related to the number of *time steps* occurring between the initial decisions and the subsequent reward. It therefore suffers from a brittleness encountered by Wilson (Wilson, 1989) with long classifier chains using the bucket brigade algorithm of Holland (Holland, 1975).

Secondly, the recurrent networks are of fixed size and face eventual saturation, though it's not clear yet when and to what degree this happens. And thirdly, the networks do not show (or have not yet shown), any

ability to augment behaviors continuously in any kind of incremental, hierarchical fashion as proposed above. (Though this issue has been dealt with in other ways. See section 4.)

3 HIERARCHIES OF "BIONS"

A "bion" is a new kind of neuron under development to allow the hierarchical accumulation of behaviors. Each bion represents a single behavior; where a behavior may be a sensation, an action, or a higher-level sequence of lower level behaviors.

A "bion" is so termed due to its dual nature: All bions that do not represent primitive sensations or actions have exactly two other bions as components. Also, all bions have two complementary functions: (a) execution of a specific behavior, and (b) perception that this same behavior has occurred. It therefore is diagramed, as in figure 1, with two parts: the intention

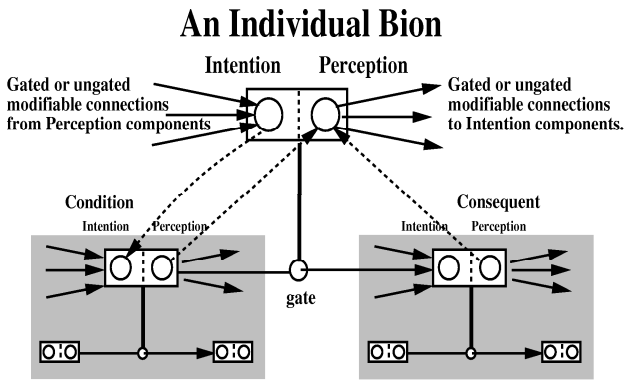

Figure 1: A Higher-level Bion Presiding Over The Transition From A Second Bion (Condition) To A Third (Consequent).

component, which causes the system to begin a certain behavior, and the perception component, which becomes activated when that same behavior has actually occurred.

For example, there may be a primitive action: "Move West." When the intention component of the bion responsible for moving west is activated, the agent attempts to move west. When the agent does successfully move west, the perception component of the bion turns on.

The intention component of a primitive sensory bion has no effect in the current implementation (though it does indicate the agent's intention or expectation to perceive the sensation). The perception component signals whether or not the condition monitored by the sensor has occurred in the world. For example, if a robot had a sensor for detecting a wall to its north side, it would have a bion whose perception component comes on whenever there is a wall to the robot's north.

The hierarchical nature of the bion is also shown in figure 1 by the higher-level bion's relationship with the two bions beneath it. The higher-level bion presides over the transition between the two lower-level bions, which might or might not be primitive. The lower bion on the left is called the "condition" and is the first bion of the transition sequence. The bion on the right is called the "consequent" and follows temporally after the completion of the first bion.

If we view the condition and consequent as representing behaviors, then the intention component of the higher-level bion, when activated, causes the condition's behavior to execute, and then, upon its completion causes the consequent's behavior to occur. When the sequence of these two behaviors has occurred, the perception component of the higher-level bion turns on. Thus, the presiding higher-level bion represents the combined behavior of the condition followed by the consequent. The intention component causes the behavior to occur, and the perception component registers that it has occurred.

3.1 BEHAVIOR OF A BION

The specific details of the bion's mechanics and functionality are quite complicated, and space allows only a general outline of its behavior.

When the intention component of a non-primitive bion is activated, it (1) turns on the intention component of its condition, (thus beginning the execution of its condition's behavior), and (2) opens the connection (i.e., closes the gate) from the perception component of its condition to the intention component of its consequent. When the perception component of the condition comes on, a signal is sent through the now open connection to the intention component of the consequent, thus initiating the behavior of the consequent.

The perception component of a higher-level bion simply signals that the bion's behavior has completed. It does this by becoming active when the perception component of its condition has come on and then is followed by the activation of the perception component of its consequent.

The intention component of a bion receives input from two sources: (a) from the intention component of any higher-level bion of which it is the condition, and (b) from the perception component of all bions connected to it through the gated or ungated modifiable connections. The strength of the input in (a) is fixed at the time the bion is created and depends only on the activation of the higher-level bion's intention component; but the strength of the input in (b) depends on the weights of the modifiable connections between the bions. These modifiable connections fully connect the network: there is one from the perception compo-

nent of each bion to the intention component of each bion. They are roughly equivalent to the asymmetric connections of standard neural networks except that here, each connection may be gated if a higher-level bion has been created to gate it (to be explained in section 3.3).

3.2 AN EXAMPLE

Figure 2 shows an example of the activity in a small bion hierarchy as it progresses through four successive time steps. The hierarchy represents the behavior: "Move west, then, if there is a wall to the north, move south." In the first time-step, t_0, the top-level bion is chosen for execution (i.e., its intention component is activated). The top-level bion then, in the same time-step, activates the intention component of its condition, MoveWest, and opens the connection from its condition to its consequent, <WallNorth MoveSouth>. At that point, the agent's MoveWest actuator attempts to move the agent west.

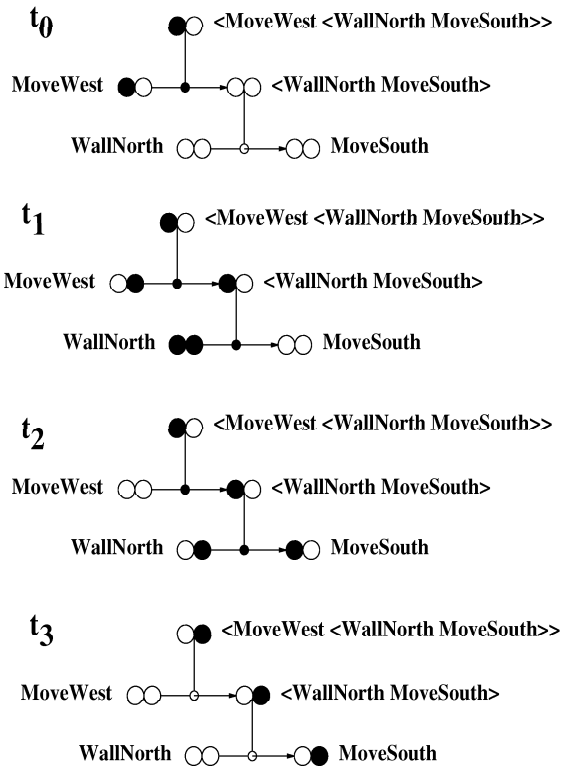

Figure 2: The Activity In A Small Bion Hierarchy As It Progresses Through Four Successive Time Steps.

Assuming the move is successful, at t_1 the perception component of the MoveWest bion is turned on to reflect that the move has occurred. Because the connection from MoveWest to <WallNorth MoveSouth> is open, the intention component of the latter is activated, which causes *its* condition to be activated, and the connection from its condition to its consequent to be opened. (But since the condition of <WallNorth MoveSouth> is a sense, WallNorth, the activation of its intention component has no effect, as was described in section 3.) However, in this example there is a wall to the north, so the perception component of the WallNorth sensory bion is activated.

At t_2, the intention component of MoveSouth is activated after it receives the signal from the perception component of the WallNorth bion. Assuming the agent can actually move south, at t_3 the perception component of MoveSouth turns on, and, since the higher-level bions <WallNorth MoveSouth> and <MoveWest <WallNorth MoveSouth>> have completed, their perception components are also turned on.

If an expected signal, such as one from the perception component of the WallNorth bion, failed to arrive at the expected time, the behavior sequence would come to a halt. (There is no explicit signal for failure.) The system would then randomly chose a bion (any bion) for execution, biased towards the bions with the greatest intention activation. Though their activations decay over time, the intention components of the higher-level bions may still be active, so the behavior could conceivably continue if the missing signal were to arrive late.

3.3 LEARNING

Modification of the weights can occur at every cycle, depending on the reward. Learning is accomplished with the delta rule (Widrow and Hoff, 1960) using reinforcement. The intention component of each bion is taken to be the bion's output, and the perception component is taken to be its target. In any given cycle, the intention component is compared against the perception component. The difference between the two is the delta value used to change the bion's incoming, modifiable weights. The amount by which each weight should be changed is then calculated, but it is *not applied* until a reward is received. While waiting for a reward, changes to the weights accumulate at each time step, but they also decay by some amount at each time step. The weight changes that occur long before the reward arrives are therefore less effective than those that occurred more recently. This entire weight change process is nearly identical to the eligibility trace of Barto et al. (1983) .

When the top-level bion in figure 2 is activated, the agent has made a single decision to choose it. After that, no more bions are chosen for execution until the entire behavior has completed. Once the behavior has completed, a new bion is chosen. When a sequence of two bion choices is made repeatedly, the system determines that their combined behavior is important and creates a new bion to represent that behavior. The new bion takes the first bion of the sequence as its

condition and the second as its consequent. The connection it gates is the modifiable connection from the perception component of the first bion to the intention component of the second. Its own modifiable connections are initially set to zero.

Since the perception components are used as targets, and since these values signal when a behavior has occurred, the agent will learn in each situation to use those and only those bions representing behaviors that actually occurred in those situations. So, though it is possible to create multiple bions that represent the same sequence (e.g. <MoveWest <MoveSouth MoveEast>>, and <<MoveWest MoveSouth> MoveEast>), in general, those constructs first built to represent a behavior will become those used to accomplish that behavior, thereby reducing the risk of creating redundant concepts.

Finally, since all mechanisms are the same for all bions, learning is identical at all levels of the hierarchy. Therefore, extending the agent's abilities is always the same, regardless of when in the agent's development the learning occurs.

4 RELATED WORK

Bions most closely resemble the nodes described by Roitblat's "Cognitive Action Theory," (Roitblat, 1991). These nodes also have two functions, "potentiation," and "satisfaction," which are quite similar to the intention and perception components of the bion. The theory, however, does not have a method for incrementally adding new behaviors, though Roitblat does mention the need to add new nodes. An interesting characteristic of Roitblat's system is that his "potentiation" and "satisfaction" functions do not necessarily correspond to a single behavior. Rather, his nodes represent goals to be accomplished, by any of several possible means, and "satisfaction" represents the accomplishment of the goal.

Also highly related is the work of Wilson, (Wilson, 1989), who has studied classifier-based reactive systems. His work is notable in that it attempts to resolve the problem of reinforcement and credit assignment in long temporal sequences by using hierarchies of classifiers. How the hierarchies would develop is not clear.

Finally, Schmidhuber (Schmidhuber, 1990b) has studied the problem of task decomposition, explicitly decomposing tasks into subgoals. He has implemented what he calls "causality detectors" to discover useful sequences of behaviors, and then, given a start state and a goal state uses gradient descent methods to generate subgoals for purposes of both planning and reinforcement.

Unlike both Roitblat's and Schmidhuber's work, bion networks do not require goals. In fact, predefined goals seem somewhat antithetical to the task of continual growth and development in an environment. Rather, it should be the reinforcement which specifies the tendencies of the agent's behavior.

5 CONCLUSIONS

Experimental work has so far demonstrated the incremental creation and utilization by an agent of small though useful behaviors such as that of figure 2.

Referring now to the problems posed in section 1.1:

(1) The problem of incremental development of behavior is solved by building hierarchies from behaviors already in the agent's repertoire. New bions are added to encapsulate behaviors either used repeatedly by the agent or used less frequently but accompanied by reward.

(2) Because every higher-level bion represents a temporal behavior—a sequence of its two component behaviors—it lasts some number of time steps. As learning progresses, behaviors of any time span whatever can be formed by combining two already learned behaviors into a new behavior whose duration is the sum of the durations of its components.

(3) Each behavior, no matter how complex, can become represented by a *single* bion. Decisions made by the agent are therefore always simple and consist only in chosing a bion. Once a bion is chosen, its behavior, no matter how complex, occurs automatically with no more decisions required by the agent until the behavior has completed. After the behavior has completed, the perception component of the bion which represents it turns on, signaling in a straight-forward manner that the activity has occurred.

(4) Because decisions are always made at the highest level, reinforcement spans back in time, not just over the previous several time steps or primitive sense-action pairs, but over the last several *decisions*. Reinforcement has a weak effect on the automatic steps which must be taken once a decision is made, but it has a strong effect on the bion choices which actually resulted in the reward.

Acknowledgements

I would like to thank the following people whose comments and discussions have helped me with this work: Rick Froom, Eric Hartman, Jim Keeler, Kadir Liano, Risto Miikkulainen, David Pierce, Jürgen Schmidhuber, and my advisor Robert Simmons.

References

Barto, A., Sutton, R., and Anderson., C. (1983). Neuron-like elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:835–846.

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

Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97.

Roitblat, H. (1991). Cognitive action theory as a control architecture. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 444–450. MIT Press.

Schmidhuber, J. (1990a). Networks adjusting networks. Technical Report FKI-125-90 (revised), Technische Universität München, Institut fü Informatik.

Schmidhuber, J. (1990b). Towards compositional learning with dynamic neural networks. Technical Report FKI-129-90, Technische Universität München, Institut fü Informatik.

Servan-Schreiber, D., Cleermans, A., and McClelland, J. (1988). Encoding sequential structure in simple recurrent networks. Technical Report CMU-CS-88-183, Carnegie Mellon University, Computer Science Department.

Widrow, B. and Hoff, M. (1960). Adaptive switching circuits. In *IRE WESCON Convention Record*, pages 96–104. IRE. New York.

Wilson, S. (1989). Hierarchical credit allocation in a classifier system. In Elzas, M., Ören, T., and Zeigler, B., editors, *Modeling and Simulation Methodology*. Elsevier Science Publishers B.V.