1993

A hybrid neural network methodology for studying the development of external memory strategies in problem-solving.

Vivek Anumolu
University of Alabama at Birmingham

Order Number 9419283

A hybrid neural network methodology for studying the development of external memory strategies in problem-solving

Anumolu, Vivek, Ph.D.

University of Alabama at Birmingham, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A HYBRID NEURAL NETWORK METHODOLOGY FOR STUDYING THE DEVELOPMENT OF EXTERNAL MEMORY STRATEGIES IN PROBLEM – SOLVING

by

VIVEK ANUMOLU

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree
of Doctor of Philosophy in the Department of Computer
and Information Sciences in the Graduate School,
The University of Alabama at Birmingham

BIRMINGHAM, ALABAMA
1993

ABSTRACT OF DISSERTATION
GRADUATE SCHOOL, UNIVERSITY OF ALABAMA AT BIRMINGHAM

Degree PH.D. Major Subject COMPUTER SCIENCE

Name of Candidate VIVEK ANUMOLU

Title A Hybrid Neural Network Methodology for Studying the Development of External Memory Strategies in Problem–solving

Neural networks among other artificial intelligence methods offer advantages of learning to adapt to novel and nonstationary environments. The operation of robots in such environments is facilitated if they can learn new strategies appropriate to the situation. To address this situation, we turn to the field of cognitive development for guidelines on how humans construct and use strategies to solve problems. An object–target matching task devised by N. W. Bray and his associates to investigate the differences in the use of external memory strategies in children is seen as a representative case. This task involves matching a subset of objects with a subset of targets, in a specified temporal order and according to relations defining positions. In studying this situation, a hybrid neural network methodology is followed that involves the integration of neural components and mechanisms based on the instar, the outstar, a sequencer and shunting excitation into a single neural network to mediate desired behavior.

A hybrid neural network, known as the "sequence–associator," is constructed to solve the matching task and consists of a serial learning module and a series of associators. The influences of postsynaptic threshold on serial learning and of a nonspecific arousal on object–target matching are examined.

The sequence–associator is extended to incorporate novelty bias and accuracy factors, modeled after those postulated by R. S. Siegler and his associates; these factors are responsible for strategy evolution. The resultant model, known as the "novelty bias neural

network," demonstrates the phenomena of strategy diversity and advancement observed in the object–target matching task and other problems.

A variation of the novelty bias model known as the "components neural network" model discards the notion of novelty bias and introduces a notion of strategy components. The latter model is based on the assumption that accuracy feedback information is selectively encoded, first for objects, next for targets and ultimately for prepositions. The assumption is motivated in part by the work of R. J. Sternberg, which established that children spend less time than adults on encoding a given problem. The components model exhibits not only the phenomena of strategy diversity and advancement but also strategy discovery. We conclude that this model offers a plausible explanation in a neural network framework for strategy discovery, a very difficult problem which existing information processing theories in cognitive development as well as neural networks have yet to address in a fully satisfactory way.

Abstract Approved by: Committee Chairman Kevin D. Kelly

Program Director

Date _____ Dean of Graduate School

ACKNOWLEDGEMENTS

I express my gratitude to two outstanding individuals without whose support and encouragement this dissertation would not have materialized. Dr. Kevin D. Reilly, who has kindly served as my advisor throughout the duration of my graduate studies at the Department of Computer and Information Sciences and as Chairman of my dissertation committee, has always inspired me with his quest for knowledge in many domains. He has encouraged me to participate in research seminars and conferences and has thereby contributed to my professional development. Dr. Norman W. Bray has not only supported my participation in research conferences but has also helped focus my thinking on the development of the models found in this dissertation. I have enormously benefited from discussions with both individuals who invested much of their time with me in the development of the models.

I am grateful to Dr. Warren T. Jones, Chairman of the Department of Computer and Information Sciences, for supporting my studies and research on neural networks, for kindly serving on my dissertation committee and for encouraging me always. In addition, I express my gratitude to Drs. Alan P. Sprague and Franklin Amthor, for kindly serving on my dissertation committee. Finally, I would like to express my appreciation to my wife, Prasannata, who encouraged me always as I spent many, many days working after hours on the dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

CHAPTER 1

INTRODUCTION

Knowledge acquisition is a topic of great import to cognitive psychologists as well as to computer scientists. It includes the acquisition of problem–solving skills that allow a human as well as a machine to adapt to novel and nonstationary environments and perform well. Even though people acquire skills such as typing, computer programming, and arithmetic, the underlying cognitive processes are not well understood. The subject of acquisition of problem–solving skills therefore has been identified as one of the key issues in cognitive science (Norman, 1980). The various strategies people use while solving problems have a practical application in improved curriculum methods in education and training of children and adults. In the robotics domain, an autonomous agent should select a strategy that is most appropriate to the prevailing environmental conditions and adapt its strategy with change in the conditions.

In the 1980's, neural networks became a popular research topic. One reason for this popularity was that the representation and processing of knowledge in neural networks comes closer to that in brain than other schemes, e.g., expert systems. The research reported herein is concerned with how a neural network can capture problem–solving skills, specifically, strategies.

When such a network is embedded in a robot, for example, the robot can acquire skills that would help it survive in a novel and nonstationary environment. Researchers were quick to see the application of neural networks to robotics in a problem–solving context, e.g., Reilly et al. (1987). Albus (1991) proposes an extensive system–theoretic approach of hierarchically decomposing a problem for constructing a robot. He mentions the application

of neural networks for problem–solving in robots but does not specify the details. Waxman and Bachelder (1992) have built a robot that classifies objects in its environment and exhibits conditioned behavior based on the object class, using neural networks. These examples illustrate the potential of neural network methodology for problem–solving by robots and the relevance of this topic to this area in computer science.

1.1 Review of Literature on Strategies in Problem–solving

The topic of strategies has been dealt with extensively in the discipline of cognitive development. One research thrust is to understand how children acquire and apply strategies in the context of problem–solving as they grow older. The precise definition of a strategy is subject to controversy (Bjorklund & Harnishfeger, 1990; Siegler & Jenkins, 1989). For the purposes of this dissertation, I define strategy as a goal–directed, nonobligatory procedure that is easy to execute and helps overcome the limitations of working memory.

Strategies may be internal or external. An example of an internal strategy is verbal rehearsal to memorize a list of digits. Examples of external strategies include writing a reminder note and adding two numbers by counting one's fingers. Recent research efforts in cognitive development have begun to investigate external strategies because they are more reliable, require less effort, and more accurate than internal strategies (Harris, 1980). Siegler has been investigating how strategies (including external strategies) evolve among children in addition, multiplication, time–telling and serial learning (McGilly & Seigler, 1989; Siegler & Jenkins, 1989; Siegler, 1991). Bray et al. (1993) have been investigating the types of external strategies that normal and mentally retarded children use in matching a set of objects with another set of objects in a specified order.

The subject of strategy development is closely tied to that of metacognition. Metacognition, simply put, refers to thinking about thinking. A person's beliefs about self and others, his knowledge about a given task and other environmental variables, and his knowledge about existing strategies all play a role in the further development of strategies

(Flavell, 1987). The beliefs and knowledge, which are jointly known as the "metacognitive knowledge," may be retrieved by that person either explicitly or implicitly. Brown (1987) discusses the influence of self– regulation and regulation from others such as teachers and peers on metacognition. After reviewing the literature on metacognition, she concludes that the concept of "executive control" is needed to explain the role of metacognition in problem–solving.

According to Sternberg's (1985) theory of intelligence, metacognition is responsible for the construction of strategies. The use of strategies involves the application to problem–solving of performance components such as encoding, inference, mapping, etc., and of knowledge acquisition components such as selective encoding, selective combination, and selective comparison. These components, in turn, guide the development of strategies. During the course of the development of this theory, Sternberg observes that adults typically spend more time on encoding to solve analogy problems than do children (Sternberg & Rifkin, 1979). He notes that even though children's practice of encoding only one or a few features of a problem reduces the initial memory load, ultimately it extends their solution time. We make use of this important observation in the construction of the components neural network model, to be presented in Chapter four.

Siegler views the development of strategies as an evolutionary process (Siegler, 1991; Siegler & Shipley, 1993). According to his theory of strategy development, there does not exist a one–to–one correspondence between age and the use of a specific strategy. Instead, at a given age, a person applies multiple strategies with different frequencies. Thus, as strategies evolve, the frequency of strategy use varies and new strategies are discovered. Based on evidence from many problem domains such as arithmetic, serial learning, time telling, and formation of past tenses, Siegler postulates that strategy evolution is guided by factors such as speed, accuracy, and novelty bias. If a given strategy results in faster or more correct execution of a given task, then a subject is likely to select that strategy again. Novelty

bias refers to the element of curiosity that propels a subject to select a strategy that has been recently discovered. Siegler derives this notion from Piaget's axiom that "if a child has a cognitive capability, he is going to put it to use."

In the current research effort, we are interested in the construction of neural network models that emulate the behavior of strategy evolution evidenced in the general problem–solving of humans. We accomplish this by specifically modeling the "object–target matching" task investigated by Bray et al. (1993), which forms our case study. We note that the subjects in this investigation use multiple strategies in a single experimental session on the same problem. Thus, the matching task meets the variability criterion in strategy usage, laid out by Siegler in his theory. Moreover, experimental measurements, readily available to us from Dr. Bray, include data on accuracy and strategy use. As the reader can see, the conditions are proper for testing the generality of Siegler's theory on strategy development by considering the specific problem of object–target matching. The predictions generated by neural network models that are constructed for solving this task can be validated by the available empirical data.

In addition to the appropriateness of Siegler's theory, we also find Sternberg's empirical studies on selective encoding relevant to the current research investigation. The relevance of Siegler's and Sternberg's theories to our neural network models will be further elaborated in Chapters three and four.

1.2 Review of Literature on Hybrid Methodology

Hybrid methodology is widely practiced in the field of neural networks. Hrycej (1992) provides an excellent review of various plausible hybrid paradigms. The two most widely used hybrid paradigms are based on learning and functionality, which we discuss here.

1.2.1 Hybrid Paradigm Based on Learning

Three types of learning schemes usually are identified in neural network literature: unsupervised learning, supervised learning, and reinforcement learning. The hybrid

paradigm based on learning combines all or parts of these schemes to derive a neural–network–based solution to a given problem. A common architecture for two–phase learning is illustrated in Figure 1.1. In this particular hybrid model, an unsupervised learning module extracts the features from the input and a supervised learning module classifies the features into one of the known classes. Since the dimensionality of the input space is reduced in the feature extraction process, this hybrid model offers the advantage of making the work of supervised learning module easier. Counterpropagation networks, hierarchical feature map classifiers, and radial basis function networks all are based on this model (see Hertz, Krogh, & Palmer, 1991 for details).

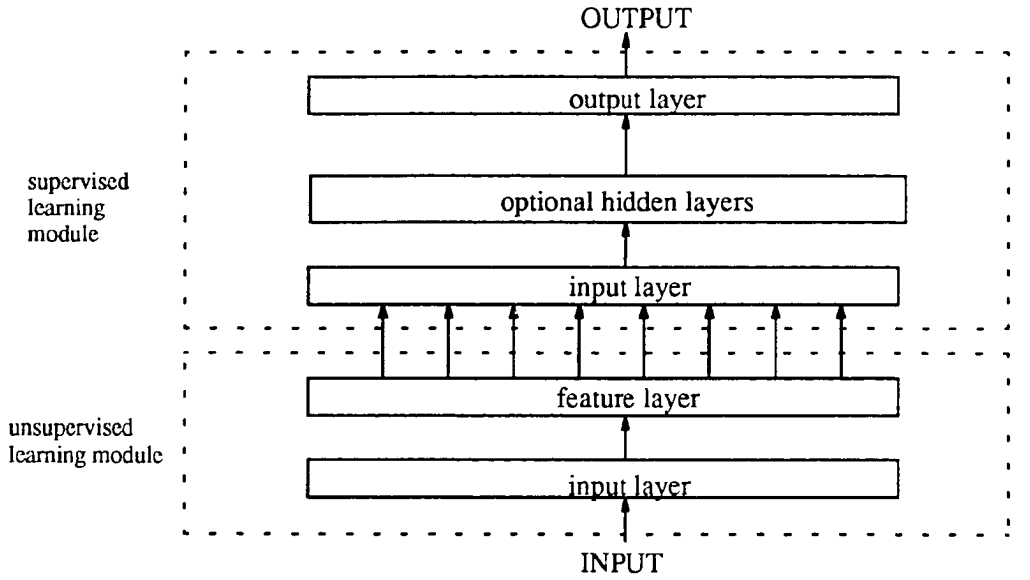


Figure 1.1. Architecture of a hybrid neural network model based on two–phase learning. There exists a one–to–one correspondance between the input layer of supervised learning module and the feature layer of unsupervised learning module (Hrycej, 1992, p.87).

1.2.2 Hybrid Paradigm Based on Functionality

Neuroanatomical studies indicate that different parts of human brain subserve different functions. Hybrid neural networks based on functionality are grounded on this simple principle. Hrycej (1992) notes that the functional hybrid paradigm is quite useful for most engineering and cognitive applications since these types of applications have inherent

structure. These applications are more easily solved by hybrid neural networks rather than by monolithic neural networks. In a hybrid solution, a task is decomposed into subtasks, and neural networks called "neural components" that match the subtasks are selected. This paradigm, thus, satisfies the principle of modularity, one of the key software engineering principles. Because neural components are neural networks themselves developed by various neural network researchers since the 1940's, the functional hybrid paradigm encourages the "reuse of code," another key software engineering principle.

The development of neural network models based on this paradigm is also appealing for the study of behavior because these models exhibit the property of "emergent behavior," which refers to behavior engendered by the network as a whole but not by individual components. Braitenburg (1984) presents various artificial neural network models that demonstrate such emergent behavior and identifies biological correlates of the hybrid models.

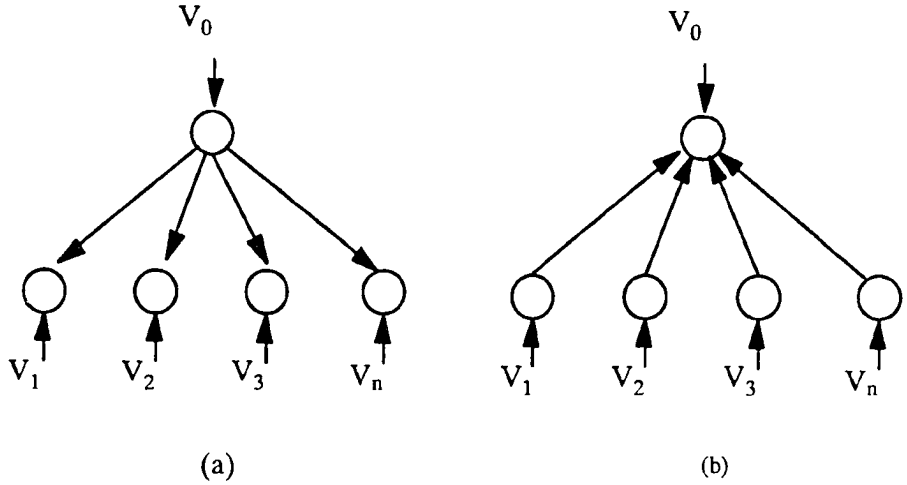
Grossberg (1974, 1978) pioneered the development of the functional hybrid neural network paradigm as applied to the study of behavior. He termed this concept the "method of minimal anatomies" because it involves the construction of a minimal network based on properties of biological neural networks. The network is analyzed rigorously using mathematical tools to account for observed behavioral patterns. This methodology has led Grossberg to discover several neural components including an instar, an outstar, a sequencer, and a gated dipole. We now briefly discuss some of the neural components developed by Grossberg and others. The interested reader is referred to Levine (1991) who provides an excellent review of various neural components that are available to a neural network builder. The work of Levine (1991) places special emphasis on cognitive and neurological modeling.

An outstar is capable of storing any arbitrary spatial pattern (Grossberg, 1974) (Figure 1.2. [a]). During training, a spatial pattern across nodes $V_1, V_2, .., V_n$ is associated with

activity at node V_0. After training, activity at node, V_0 triggers recall of the stored pattern across the series of nodes.

An instar is the complement of the outstar (Figure 1.2. [b]). During training, activity at node V_0 is associated with activity across nodes $V_1, V_2, .., V_n$. After training, activity that is a linear combination of activities across nodes $V_1, V_2, .., V_n$ is triggered at node V_0 (Grossberg, 1974).

In Grossberg's formulation, a sequencer is a fully connected neural network which learns to store a sequence of items in a specified order. Training of the network involves repeated presentation of items to its nodes (Grossberg & Pepe, 1971). This is analogous to a rehearsal strategy which is common among humans of all ages. Full details of the workings of a sequencer are presented in Chapter two.



(a) (b)

Figure 1.2. Architectures of neural components: (a) outstar, (b) instar.

The neural component known as "gated dipole" compares the current values of stimulus or reinforcement variables with recent past values of the same variables (Levine, 1991). Figure 1.3 shows its architecture. A nonspecific input, I, is fed to both channels in the network, y_1–to–x_1–to–x_3 and y_2–to–x_2–to–x_4, whereas an input J, such as electric shock given to an animal in a conditioning experiment, is fed to left channel only as shown in

Figure 1.3. The processes of transmitter depletion and feedforward competition in the network lead to patterns of activity as shown in Figure 1.4. The network exhibits a transient response after input J is removed which is analogous to relief observed in the animal after shock is turned off. Levine and Leven (1993) designed a network of gated dipoles to model consumer preferences for soft drinks. Their work takes into account not only sensory factors such as taste of a product but also motivational factors such as novelty and security. They conclude that sensory and motivational factors jointly explain the success or failure of a product.

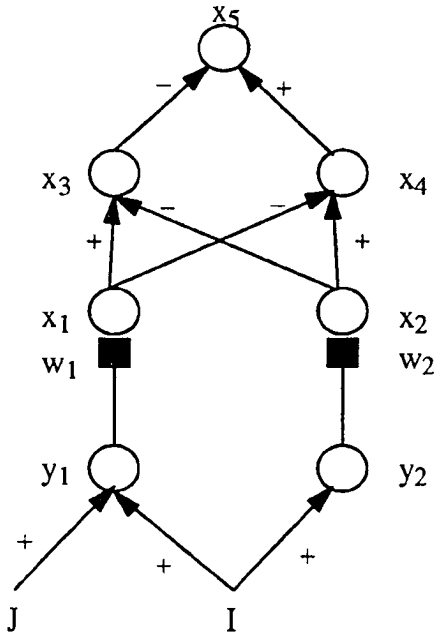


Figure 1.3. Schematic gated dipole network. J is a significant input while I is nonspecific arousal (Levine, 1991).

In his theory on neuronal group selection, Edelman (1978, 1987) postulated that neurons in the cerebral cortex organize into groups based on principles of confinement, selection, and competition. A neuronal group is a functional component that elicits varying levels of response to different stimuli. The organization of the cortex into neuronal groups

offers a plausible explanation for the existence of plasticity in auditory, somatosensory, and visual cortical areas of adult animals (Pearson, Finkel, & Edelman, 1987).

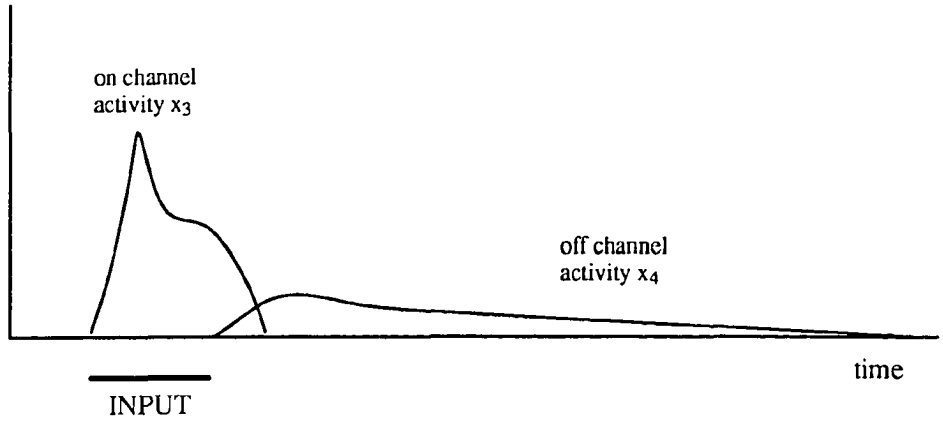


Figure 1.4. Typical time course of the channel outputs of a gated dipole (Levine, 1991).

The autoassociator type neural network models developed by such researchers as Anderson et al. (1977) and Hopfield (1982) may be deemed as neural components that store patterns by carving attractors in an energy landscape. These components may be termed "neural attractors" and can be used as part of a hybrid network.

Other neural network researchers have used the hybrid paradigm based on functionality, though often only implicitly. Edelman and Reeke (1982) in their Darwin II simulation combine two subnetworks for pattern classification: one neural network responds to local features of the stimulus, and the other to the global features. Levine and Prueitt (1989) study the effects of frontal lobe damage by combining neural components such as gated dipoles and outstars, and principles from the theory of adaptive resonance. Reilly and Villa (1990) propose a hybrid neural network scheme based on barrel structures in the somatotopic maps of rodents to be applied to computer and other communications systems.

In the context of the present research effort, which is concerned with the study of strategies for problem–solving, we first emphasize the hybrid paradigm based on functionality. As noted by Hrycej (1992), most cognitive and engineering applications have

a built–in structure that makes the use of this paradigm feasible. The important problem of object–target matching which is dealt with in this dissertation consists of a sequence of instructions, three sets of objects, targets, and prepositions used in the instructions, and a set of commonly used strategies. The interactions among all these entities give rise to the structure in the problem.

Besides the hybrid paradigm based on functionality, we apply the hybrid paradigm based on learning in the construction of neural networks discussed in Chapter three and Chapter four. These latter networks are based on unsupervised learning, which refers to self–organization and discovery of strategies by children, and on supervised learning, which refers to the external presentation of accuracy information to the network.

It is not known whether strategy development in the context of object–target matching and arithmetic has been studied in the framework of neural networks previously.

1.3 Overview of the Dissertation

In the current chapter, we have presented the significance of the current investigation and briefly reviewed literature from cognitive psychology and cognitive development as it pertains to problem–solving strategies. We also reviewed briefly the literature on hybrid neural network methodology.

In Chapter two, we introduce the object–target matching task which has been devised by Bray et al. (1993) for the purpose of investigating the differences in strategy use among children of various chronological and intelligence groups. We view this task as consisting of two subtasks: storage of a sequence of instructions and association of the stored instructions with a set of objects, targets, and prepositions. We present two neural network models designed to perform these two subtasks. The first neural network, a "sequence generator," stores and recalls a sequence of instructions. The second neural network, a "sequence–associator," is a hybrid of the sequence generator neural network and a series of associators that can associate the instructions with the sets of objects, targets, and

prepositions. We carry out computer simulations of these two models and present the results of the simulations using two interpretations: a probabilistic interpretation and a maximum activation interpretation. We also study the role of a postsynaptic threshold in serial learning and of a nonspecific arousal mechanism in serial learning as well as in object–target matching.

In Chapter three, we extend the sequence–associator neural network so that it can simulate the behavior of strategy selection and evolution in the context of the object–target matching task. The new model, known as the "novelty bias neural network," is based on the novelty bias and accuracy factors suggested by Siegler. In this model, we hypothesize that strategy selection as observed by Bray et al. (1993) in various chronological and intelligence groups are guided by these two mechanisms. We study the effects of strategy use on the recall accuracy of the various entities involved in the task.

We present a neural network model that manifests the behavior of strategy selection and evolution in Chapter four, similar to the novelty bias neural network model presented in Chapter three. The model in Chapter four, however, known as the "components neural network," eliminates novelty bias as a controlling factor in strategy evolution. Instead, it incorporates the idea of strategy components. The model is based on the hypothesis that human subjects, as suggested by the work of Sternberg and others, selectively encode accuracy information about the various components of strategies and that the ability to encode this information increases with experience with the matching task. First, the effects of strategy use by the model on recall accuracy are studied, and then the frequency of strategy selection by the model is compared to that of the subjects in the study of Bray et al (1993).

In Chapter five, we draw some conclusions based on our experience with neural network modeling in earlier chapters. Specifically, we discuss how the neural network models presented in Chapters three and four meet the five empirical phenomena which Siegler and Shipley (1993) associate with problem–solving strategies. We also address the question of

how the neural network models presented in this dissertation can be further enhanced and/or applied to related cognitive tasks, business applications and robotics.

CHAPTER 2

A HYBRID SEQUENCE – ASSOCIATOR NEURAL NETWORK MODEL

The study of Bray et al. (1993) provides one of the primary motivations behind the construction of all neural network models presented in this dissertation. In this chapter, we briefly review the experimental setup utilized in their study for investigating differences in the use of external memory strategies among mentally retarded and nonretarded children. We henceforth refer to the task involved in this study as the "object–target matching task." We follow the review with discussion of two neural network models: a sequence generator model and a hybrid sequence–associator model. The former model learns to generate a sequence of instructions. The latter model consists of a sequence generator as its neural component and additionally consists of an outstar–like associator that associates the generated instructions with cognitive mappings of entities. The outcomes of computer simulations of these models are presented based on two interpretations: a normalized activation scheme and maximum activation scheme. Lastly, we study the effects of a nonspecific arousal mechanism on the performance of the models.

2.1 Motivation

Humans tend to use a variety of strategies in improving their memory performance in daily life. Bray et al. (1993) investigated the differences between mentally retarded and nonretarded children in the use of external memory strategies. External memory strategies refer to the use of "external" memory aids such as putting objects in special places, writing reminder notes, and asking other people to help remember something.

An experimental setup designed to facilitate the testing of the use of external strategies and utilized in the study of Bray et al. (1993) is shown in Figure 2.1. A subject is asked to listen to a sequence of instructions and then carry them out. An instruction such as "put the apple on the couch" always has the format of "Put <object> <preposition> <target>." One of the 12 toy objects is to be matched with one of the 6 toy targets. Only two prepositions, "on" and "in front of," are used as part of the instructions. Throughout the current investigation, we only consider sequences of four instructions. An example of such a sequence is given below:

(1) "Put the apple on the couch."

(2) "Put the penny in front of the TV."

(3) "Put the rock in front of the table."

(4) "Put the stamp on the refrigerator."

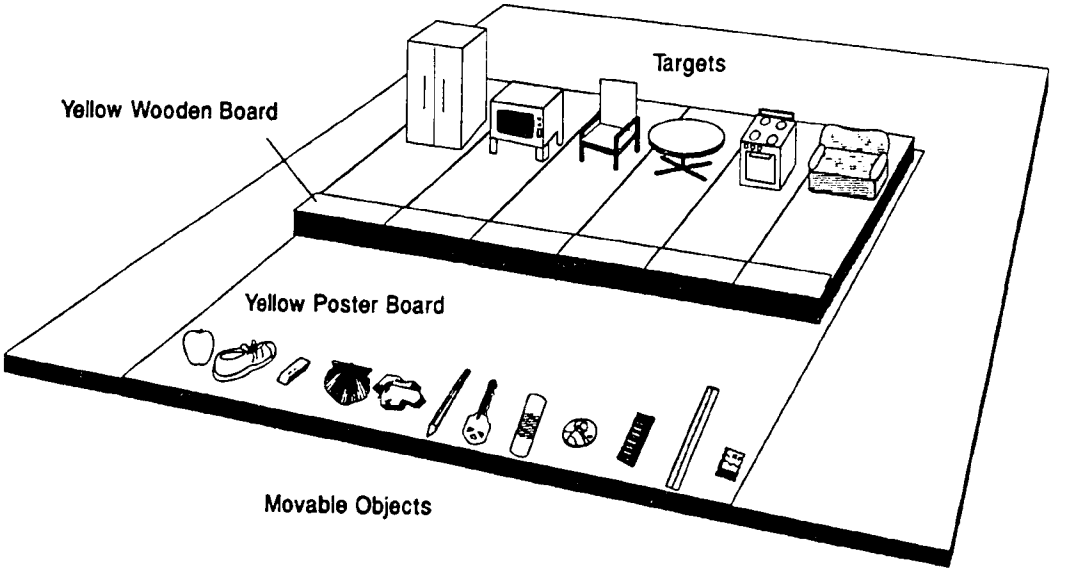


Figure 2.1 Experimental setup used in the study of external memory strategies (Bray et al., 1993).

While listening to the sequence, the children are allowed to use external strategies. A bell after the fourth instruction is a signal to begin carrying out the instructions in the order given. In variants of this experiment, order is waived, but we are concerned exclusively with the ordered case. Typical strategies that children use while listening are holding an object in one hand, pointing an object to a target, or moving an object close to a target.

Complex human behavior, in general, may be viewed as a product of multiple subsystems working together. We adopt this viewpoint in modeling this experimental task. The first neural network model that we present learns to store and recall a sequence of instructions and is henceforth referred to as the "sequence generator." Such learning is analogous to the development of seriality in children who acquire this concept over a period of many years as they actively participate in everyday life. This model is to be viewed as an independent network that can be embedded in or otherwise collaborate with another network in order to effect behavior.

After the sequence generator has learned the concept of seriality, it is coordinated with a second model for learning the associations between objects, prepositions, and targets in correct sequence. The second model may be viewed as a hybrid neural network that consists of a preconditioned sequence network and an associative network that is learned as a part of a situation the subject is expected to master. This model is henceforth referred to as the "sequence associator." The hybrid of a model that represents past learning and a model that quickly learns current situation–specific associations has great generality for modeling behavior of both humans and robots. Villa and Reilly (1992) review the role of hierarchical structures in building hybrid systems.

2.2 A Model to Learn the Serial Order of Instructions

Grossberg (1969), Grossberg and Pepe (1971), and Grossberg (1978) present neural network models for serial learning of a list of items. We find these models relevant in the simulation of the serial learning component of the Bray and coworkers (1993)' experiment.

However, these models need adjustment to exert refined control over both forward and backward associations (among the units of a list). In this chapter, emphasis falls on control of backward associations.

In the current modeling, an instruction unit activates object, preposition and target units as well as the next instruction unit. If backward associations are present, they sustain the activations of previous instruction units over longer durations and lead to continued recall of items of previous instructions. This is undesirable since frequent repeated recall of the items is not observed in humans. Only current instructions should be activated. For the purposes of modeling the psychological experiment at hand, backward associations need to be attenuated.

We introduce a postsynaptic threshold as a neurologically feasible mechanism to attenuate backward associations. Edelman (1987) discusses the role of such a threshold in modifying synaptic efficacy and in organizing cortical neurons into topographical maps, but the effects of such a mechanism have not been actively explored in the context of a working neural network model.

2.2.1 Training

Our first version of a sequence generator learns to store and recall a sequence of instructions. Its architecture is displayed in Figure 2.2a. All units are fully connected. The four instruction units correspond to the four instructions in the experiment.

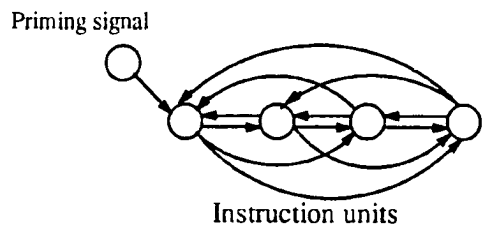


Figure 2.2a. A sequence generator neural network to learn a sequence of instructions. A priming signal stimulates the network to recall the learned sequence.

In our training procedure, we have initialized activation of instruction units to zero and initialized the excitatory connections among them to small random values. The first instruction unit is then presented with external input for one cycle and the activations and weights are updated using equations (2.1) and (2.2).

Activation update equation for instruction units :

$$a_i(t+1) = (1-\alpha)\, a_i(t) + \beta \sum_j [a_j(t)]^{\Gamma_1}\, w_{j\,i}(t) + I_i(t) \qquad (2.1)$$

where

$a_i(t)$ = *activation of unit* i *at time* t

α = *decay rate*

β = *net input factor*

Γ_1 = *presynaptic threshold*

$[x]^{\Gamma_1}$ = x *only if* $x \geq \Gamma_1$
= 0.0 *otherwise*

$w_{ji}(t)$ = *connection strength from unit* j *to unit* i *at time* t

$I_i(t)$ = *external input to unit* i *at time* t.

Weight update equation :

$$w_{i\,j}(t+1) = w_{i\,j}(t) + \delta_1\, [a_i(t+1)]^{\Gamma_2}\, [a_j(t+1)]^{\Gamma_3} \qquad (2.2)$$

where

$w_{ij}(t)$ = *connection strength from unit* i *to unit* j *at time* t

δ_1 = *learning rate*

$a_i(t)$, $a_j(t)$ = *presynaptic* and *postsynaptic activatons at time* t

Γ_2, Γ_3 = *presynaptic* and *postsynaptic thresholds.*

In the next cycle, the second instruction unit is presented with external input; the activations and weights are updated for all units until the last instruction unit. The network is then run without any external stimulus. This corresponds to a time interval elapsed between presentations of two sequences.

Then the whole procedure is repeated starting with the first instruction unit. The repetitions during the training phase correspond to the experience of a child with everyday tasks from infancy, which results in the learning of the concept of seriality. The number of repetitions in the computer simulation depends on chosen parameter values. Typical parameters that we have used in computer simulations are listed in Table 2.1. The weights in this model do not stabilize with increased training, so that, when the maximum weight

exceeds 1.0, we halt the training. With the parameters specified in Table 2.1, we have found nine repetitions adequate. Due to the absence of randomness in the processing of the model, the weights are found not to vary from one run to another. The effects of postsynaptic threshold on the development of weights is seen from Table 2.1.

At a lower value of this threshold (0.2), some backward associations are observed, whereas no such associations are observed at a higher value (0.4). Therefore, on all our subsequent simulations we have set the postsynaptic threshold to 0.4. We also note here that forward associations are established if the postsynaptic threshold is greater than the presynaptic threshold. Otherwise, forward associations as well as backward associations are established, leading to interference in the recall of the sequence.

Table 2.1. Weights at the end of training phase in the sequence generator network. The postsynaptic threshold Γ_3 is set to 0.2 (0.4).

Sending instruction unit	Receiving instruction unit			
	1	2	3	4
1	0.00 (0.00)	0.355 (0.354)	0.05 (0.05)	0.05 (0.05)
2	0.05 (0.05)	0.00 (0.00)	0.545 (0.465)	0.209 (0.05)
3	0.05 (0.05)	0.220 (0.05)	0.00 (0.00)	0.797 (0.536)
4	0.05 (0.05)	0.05 (0.05)	0.581 (0.05)	0.00 (0.00)

Parameters used in this simulation : α=0.7, β=0.7, Γ_1 = 0.1, Γ_2=0.1, δ_3= 0.4. The weights for Γ_3 = 0.4 are shown in parentheses.

2.2.2 Recall

During the recall phase, the first instruction unit is stimulated with a priming signal, which corresponds to recall of the instructions at the sound of the bell in the laboratory experiment. The network is allowed to cycle without any weight modification or further external input. The changes in activations of all instruction units are plotted in Figure 2.3a.

The activations of instruction units rise and fall across cycles. The first instruction reaches its peak initially followed by the second instruction unit which is followed by the third unit. Lastly, the fourth instruction unit reaches its peak. The order of rise and fall of activations of instruction units indicates that the network is recalling the instructions in correct sequence. However, the peaks of activations are diminished as recall progresses toward the end of the sequence.

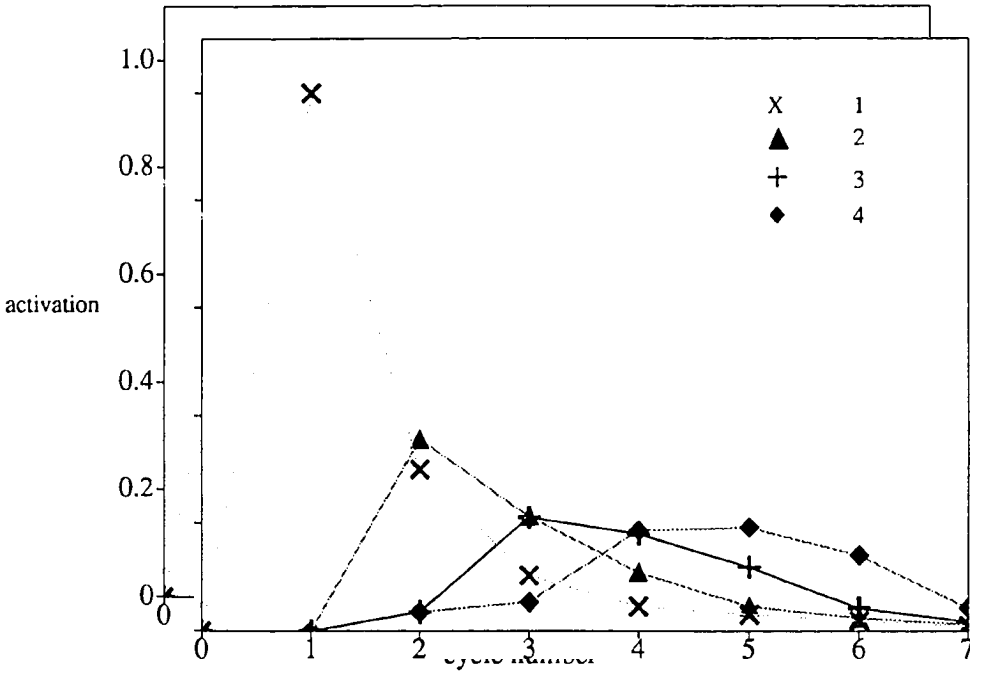


Figure 2.3a. Recall of instruction units in the sequence generator network. A priming signal stimulates the first instruction unit at the beginning of recall. A nonspecific arousal signal is absent in this case.

The recall performance of the network is assessed using the "normalized activation scheme," which is analogous to the "probabilistic scheme" of McClelland and Rumelhart (1988). These authors have successfully applied the latter scheme to cognitive modeling such as modeling effects of context and stimulus on word perception and speech perception. The normalized activation scheme consists of computing averages of activations over consecutive cycles and inferring recall accuracy from the running averages. Recall of an

entity is deemed accurate if its normalized activation is the highest in the cycle of its training. The formula for computing normalized activation of recall is stated by equation (2.3).

normalized activation of unit, i at time t :

$$n_i\ (t) \;=\; \frac{\overline{a}_i\ (t)}{\sum_k \overline{a}_k\ (t)} \tag{2.3}$$

where

$$\overline{a}_i\ (t) \;=\; \textit{average activation of unit } i \textit{ at time } t$$
$$=\; \Lambda\ a_i\ (t) + (1 - \Lambda\ )\ \overline{a}_i\ (t-1) \tag{2.4}$$

k ranges over units of the same class, for example, objects
Λ is a weighting factor; in current implementation, set at 0.5.

Figure 2.3b illustrates changes in normalized activations for all four instructions in the sequence generator model with time. Instruction one has the highest normalized activation of recall as represented by the white bar at cycle one in Figure 2.3b. At cycle two, instruction two has the highest normalized activation of recall, as represented by the black bar and so on. Thus the network has self–organized to store and recall a sequence of instructions in the correct order and has a recall accuracy of 100%.

Figure 2.3b also indicates that normalized activation for instruction unit one at cycle one is greater than that for instruction two at cycle two. Thus, the network displays a strong primacy effect in its activation levels. Furthermore, the normalized activation for instruction unit four at cycle four is greater than that for instruction three at cycle three. Thus, the network also displays a recency effect in its activation levels. Overall, the network exhibits a bowed pattern of recall.

In the second version of our sequence generator network, we studied the role of a nonspecific arousal signal on recall as a neurologically feasible, compensating mechanism for diminishing peak activations. At the beginning of recall, such a signal stimulates all instruction units (Figure 2.2b). In addition to the nonspecific arousal signal, the first instruction unit receives a priming signal mentioned above. The effects of combined priming and nonspecific arousal on recall are illustrated in Figure 2.3c.

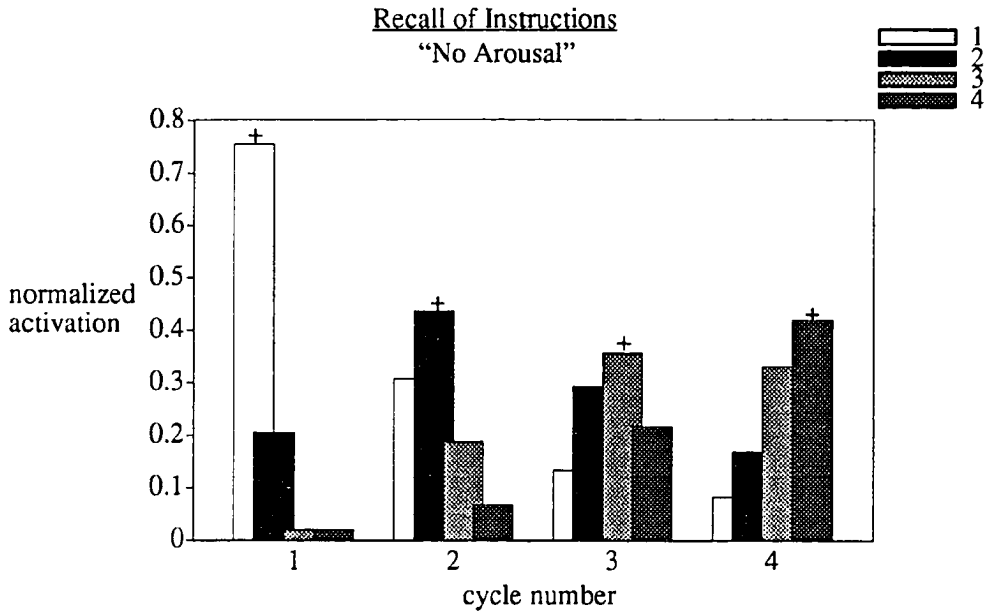


Figure 2.3b. Normalized recall of instruction units in sequence generator network. A priming signal stimulates the first unit during the beginning of recall. A nonspecific arousal signal is absent in this case. "+" above a bar indicates correct recall of the corresponding entity.

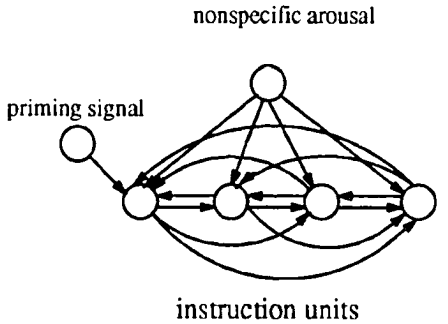


Figure 2.2b. A sequence generator neural network to learn a sequence of instructions, with a nonspecific arousal signal. As before, a priming signal stimulates the network to recall the learned sequence.

With nonspecific arousal, the network recalls the first, second, and fourth instructions correctly and does not recall instruction three correctly; thus, the network has a 75% recall accuracy (Figure 2.3c). The inspection of activations of instruction units over time indicates that nonspecific arousal has indeed boosted their values including peaks. The primacy effect

has become weaker as indicated by a drop in the normalized activation of instruction one, as seen in Figure 2.3c. At the same time, the normalized activation of instructions three and four has increased at cycle one, two, and three, indicating a stronger recency effect. It is known that children differ in the degree of primacy and recency which they demonstrate in actual experiments. Our simulation suggests that primacy and recency may be influenced by cognitive and /or neural mechanisms similar to nonspecific arousal.

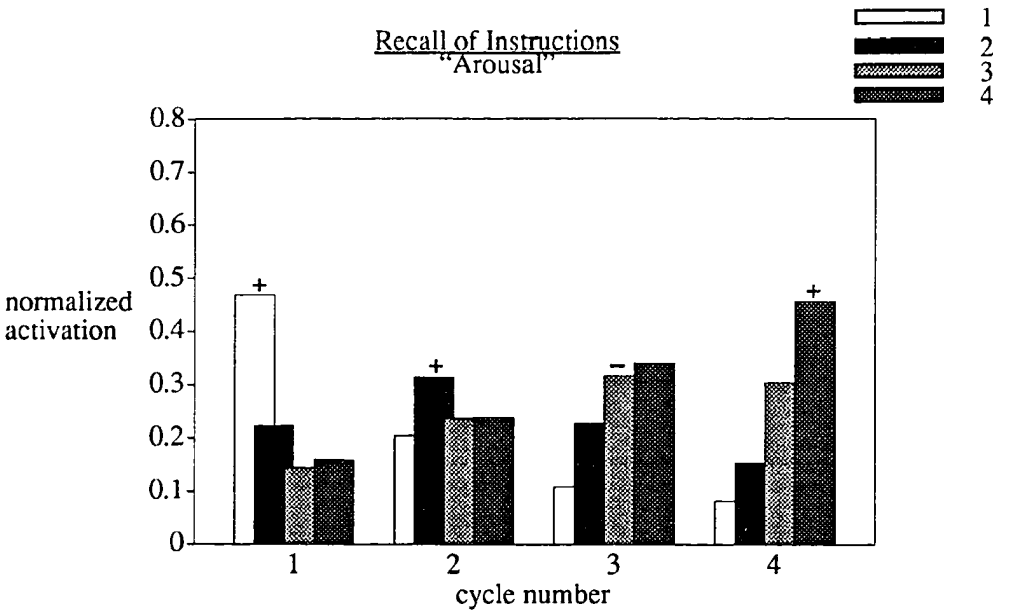


Figure 2.3c. Normalized recall of instruction units in sequence generator neural network in the presence of a nonspecific arousal signal. "+" above the bar indicates correct recall of the corresponding entity and "–" indicates incorrect recall.

2.3 A Model to Execute a Sequence of Instructions

The hybrid model that we have developed and tested consists of two components: a sequence generator component which learns to store and recall a sequence of instructions and an associator neural component which learns to store and recall different cognitive maps for an arbitrary set of instruction units (Figure 2.4).

Our hybrid model, referred to as "sequence–associator neural network," has features similar to an avalanche model, which is used for learning arbitrary spatio–temporal patterns

(Grossberg, 1978). Unlike Grossberg's model, the sequence–associator is concerned with temporal control of several spatial maps rather than a single spatial map. Each spatial map represents a cognitive mapping for a class made up of several items, for example, objects. Within a given cognitive map, lateral inhibitions exist among the items of that class so that at any given time only a few items are activated.

The architecture of the sequence–associator neural network is illustrated in Figure 2.4. Each instruction unit sends excitatory connections to all object, preposition, and target units. Each item within a class is inhibited by other items within that class. This lateral inhibition serves to enhance the activity of units with high activation and to suppress the activity of units with low activation.

2.3.1 Training

At the beginning of the training phase, the connections among instruction units in the sequence generator component are initialized to those weights which have been learned during the training phase for the sequence generator neural network previously. This corresponds to a child having already learned the concept of seriality before participating in an experiment in the psychologist' laboratory. These weights do not change during training.

The weights between instruction units and item units, however, do undergo modification during training. They are initialized to small random values (ranging from 0.0 to 0.05). The inhibitory connections among items of the same class have fixed weights, which are each assigned a value of –0.2. The predefined units of the sequence generator are effectively static, while the units of the associator undergo changes, illustrating the hybrid nature of the learning mechanisms.

In a typical simulation, the network is trained on the following sequence of instructions in the given order: (1) Put the apple on the couch, (2) Put the penny in front of the TV, (3) Put the rock in front of the table, (4) Put the stamp on the refrigerator.

Whenever an instruction is presented to the hybrid network, the corresponding instruction unit, object unit, preposition unit, and target unit receive external input. Activations and modifiable weights are updated using equations (2.4) and (2.5), respectively.

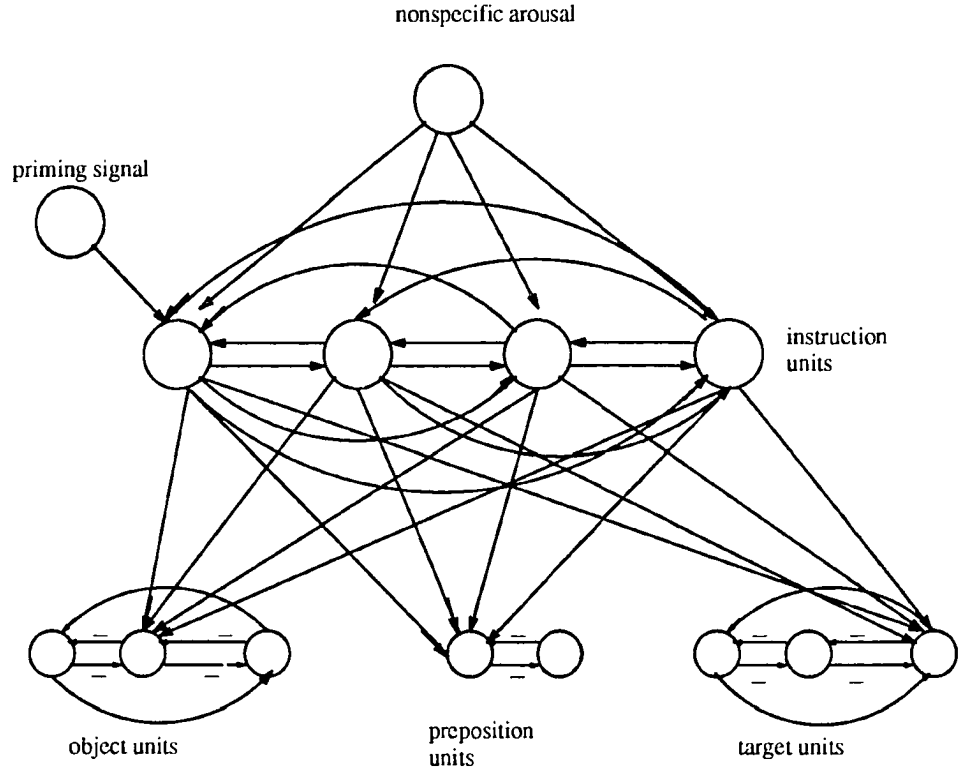


Figure 2.4. Architecture of the sequence–associator neural network. This is a hybrid of sequence generator, associator, and nonspecific arousal.

The learning rate for connections between instruction units and item units is set high so that one cycle is enough to partially learn associations between instructions and corresponding items that make up the instructions. Training consists of only one cycle just as the child in the psychological experiment hears the instructions only once. Some of the parameter values used in a typical simulation are given below:

2.3.2 Recall

As with the sequence generator model, we have studied the role of priming and nonspecific arousal on the normalized activations for item units in the sequence–associator model. The first instruction unit is stimulated with a priming signal at the beginning of recall (Figure 2.4). Then the network is run without any further external input. The activation values of objects, prepositions and targets are observed by the modeler. The normalized activations of various objects used in a typical simulation (apple, penny, rock, and stamp) are illustrated in Figure 2.5a.

Activation update equation for item units:

$$a_i(t+1) = (1-\alpha)\, a_i(t) + \beta \sum_j [a_j(t)]^{\Gamma_1}\, w_{j\,i}(t) + \gamma \sum_k q\, [a_k(t)]^{\Gamma_1} + I_i(t) \qquad (2.4)$$

where

$w_{ji}(t)$ = *connection strength from instruction unit j to item unit i at time t.*
γ = *net inhibition factor*
q = *fixed inhibitory strength among item units of the same class*
Other terms are same as specified in eq. (1).

Weight update equation:

$$w_{i\,j}(t+1) = w_{i\,j}(t) + \delta_2\, [a_i(t+1)]^{\Gamma_4}\, [a_j(t+1)]^{\Gamma_5} \qquad (2.5)$$

where

$w_{ij}(t)$ = *connection strength from instruction unit i to item unit j at time t.*
$a_i(t+1)$ = *activation of instruction unit i at time* $t+1$
$a_j(t+1)$ = *activation of item unit j at time* $t+1$
Γ_4, Γ_5 = *presynaptic* and *postsynaptic thresholds.*
δ_2 = *learning rate*

$\alpha = 0.7$, $\beta = 0.7$, $\gamma = 0.2$, $q = 0.2$, $\Gamma 1 = 0.1$, $\Gamma_4 = 0.1$, $\Gamma_5 = 0.2$, $\delta_2 = 0.2$.

We note here that there is variation neither in the external inputs presented to the units nor in the activation processing of the units, in both sequence generator and sequence–associator. Moreover, each object and target is repeated only once in a given instruction sequence in the simulations of sequence–associator neural network. For these

reasons, the performance of the models as measured by recall accuracy is independent of any particular combination of objects and targets. This independence is not valid in the case of prepositions where they are repeated twice in a given instruction sequence and thus, the order of prepositions determines the accuracy of recall.

With no arousal component, the hybrid network recalls the objects accurately in correct order with the exception of rock which has higher normalized activation than stamp in the fourth cycle (as indicated by unequal heights of the coarsely and densely shaded bars in Figure 2.5a). Figure 2.5a shows that three out of four objects are being correctly recalled and hence, the network exhibits a recall accuracy of 75% for objects.

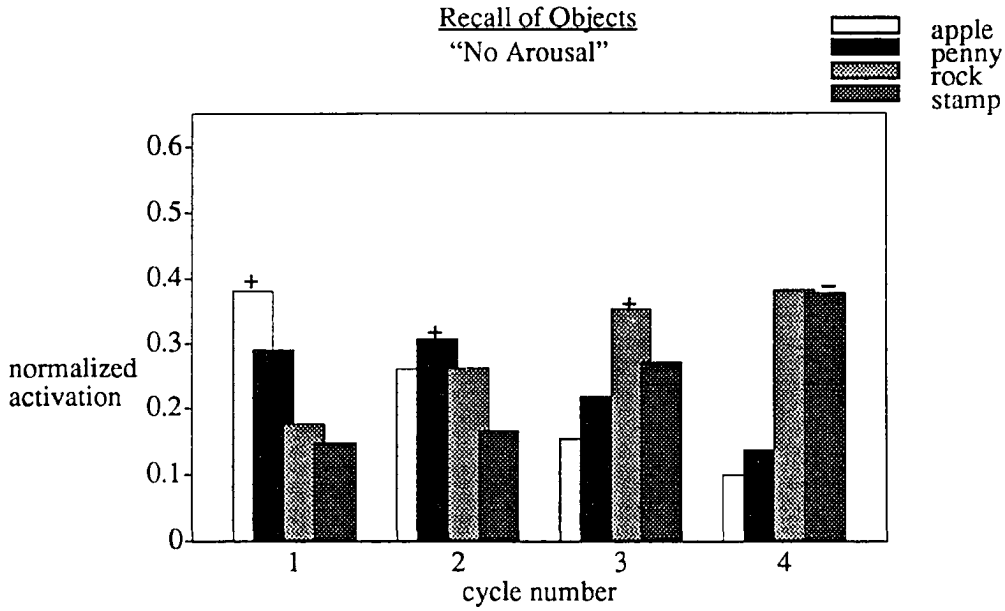


Figure 2.5a. Normalized recall of objects in the sequence–associator model in the absence of a nonspecific arousal signal.

The normalized activation of object one (apple) is significantly larger than that of other objects at the beginning of recall (as indicated by the height of the white bar in Figure 2.5a). Thus, the activation levels of the model exhibit a primacy effect. The normalized activation of the object in the second instruction (penny) is slightly less than that of the other three objects (as indicated by the height of the black bar in Figure 2.5a). Thus the activation levels

of the model exhibit slight bowing in a manner similar to that obtained for actual recall in the experiment by Bray et al. (1993).

The hybrid model with no arousal component exhibits identical responses for targets and objects and exhibits a recall accuracy of 75% for targets. This is because objects and targets are treated identically in the training phase: each object or target is not repeated more than once and an object and a target are presented to the network with equal amounts of external inputs.

In this particular simulation run, as mentioned above, the following combination of prepositions has been used in that order in training the sequence–associator network: (on, in front of, in front of, on). The network recalls prepositions accurately in correct order except for the last instruction, where "in front of" has a higher normalized activation than "on" (Figure 2.6a). Figure 2.6a also illustrates the absence of a bowing pattern in the activation levels of prepositions.

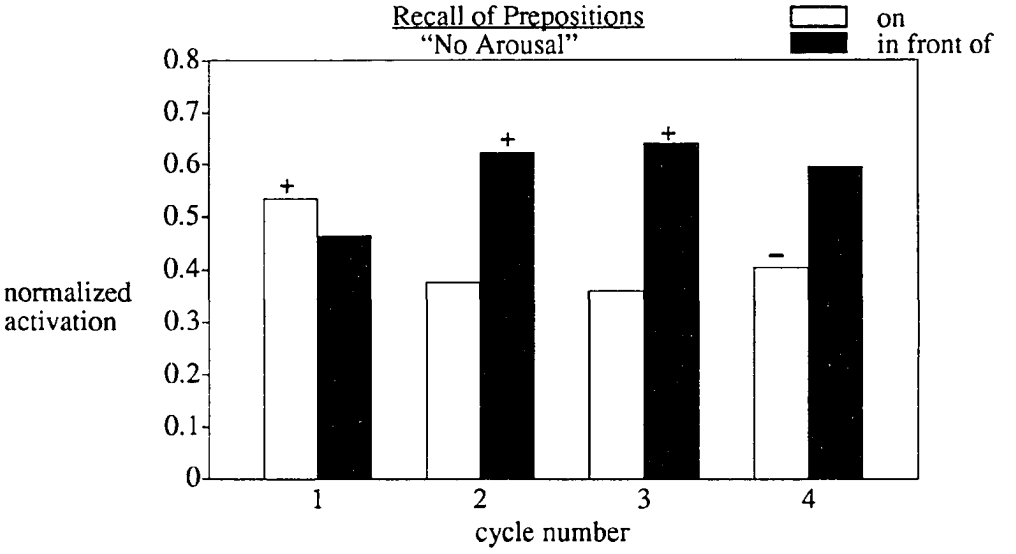


Figure 2.6a. Normaized recall of prepositions in the sequence–associator model in the absence of a nonspecific arousal signal. This particular recall is obtained after training the network with an instruction sequence that contained the combination of (on, in front of, in front of, on).

The performance of the sequence–associator neural network on prepositions is summarized in Table 2.2 which indicates that according to the normalized activation interpretation, the recall accuracy of prepositions in the absence of nonspecific arousal varies from 75% to 100%. The variation in accuracy is due to the fact that a preposition is used multiple times in the same instruction sequence. The persistent activity in a preposition unit as a result of multiple usage of that preposition causes interference in the recall of the other preposition by the network. Thus, the order of prepositions determines whether the interference is favorable or not to correct recall.

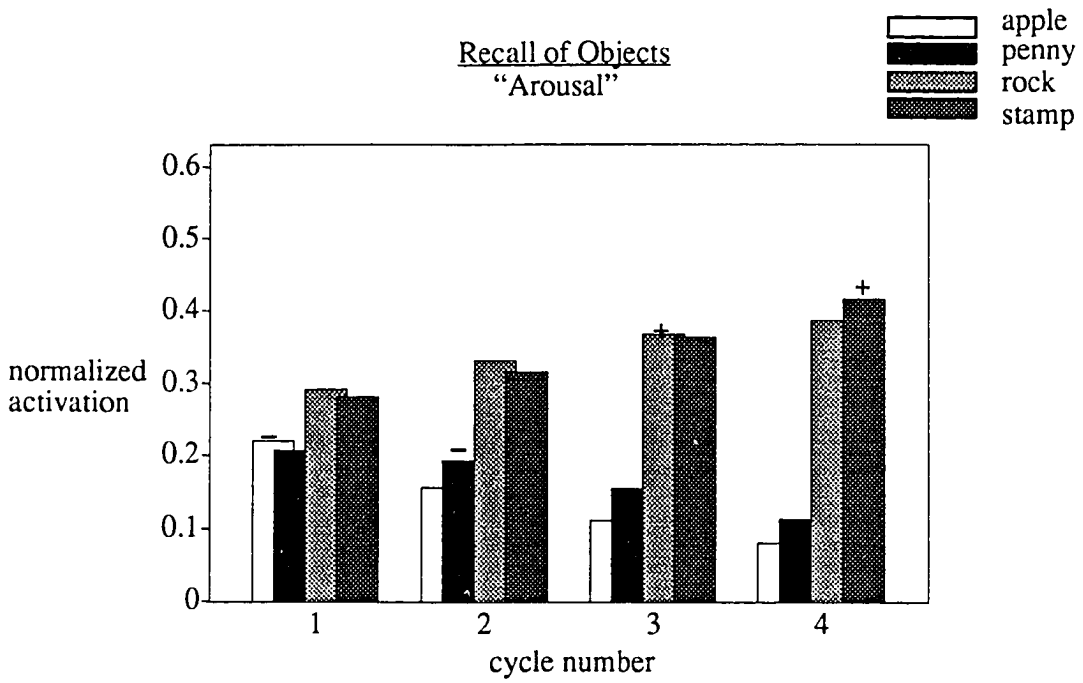


Figure 2.5b. Normalized recall of objects in the sequence–associator model in the presence of a nonspecific arousal signal.

In the sequence–associator model with an arousal component, all instruction units were stimulated with a nonspecific arousal signal at the beginning of recall, along with a priming signal to the first instruction unit. Then the network was run without any external input. The normalized activations for objects and prepositions are illustrated in Figures 2.5b and 2.6b respectively. Objects in the third and fourth instructions dominate the other two objects

throughout the duration of recall. The preposition "in front of" dominates the preposition "on" throughout the duration of recall for this particular combination of (on, in front of, in front of , on) used in training the network. Targets exhibit response patterns that are identical to those of objects.

Table 2.2. Recall of prepositions in the sequence–associator neural network when a nonspecific arousal signal is absent. Evaluation of recall is based on the normalized activation scheme.

Number	Combination	Accuracy
1	on, on, front, front	100%
2	on, front, on, front	75%
3	on, front, front, on	75%
4	front, front, on, on	100%
5	front, on, front, on	75%
6	front, on, on, front	75%

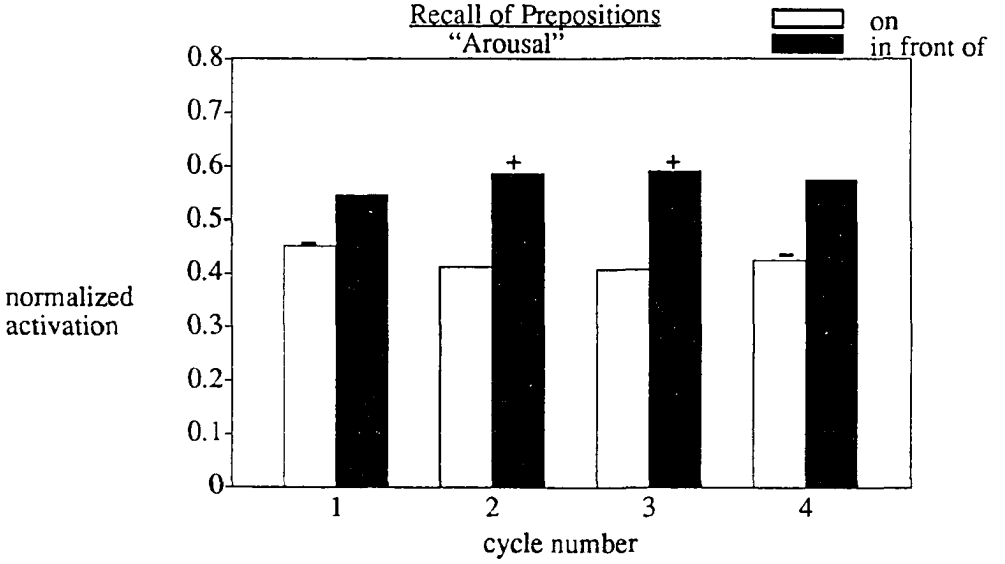


Figure 2.6b. Normalized recall of prepositions in the sequence–associator model in the presence of a nonspecific arousal signal. This particular recall is obtained after training the network with an instruction sequence that contained the combination of (on, in front of, in front of, on).

Overall, presence of the nonspecific arousal mechanism causes a strong recency effect in the recall of all items. In its presence, the network exhibits a recall accuracy of 50% each for objects and targets. The accuracy of recall by the network on various combinations of prepositions varies from 50% to 75%, as summarized in Table 2.3.

Table 2.3. Recall of prepositions in the sequence–associator neural network when a nonspecific arousal signal is present. Evaluation of recall is based on the normalized activation scheme.

Number	Combination	Accuracy
1	on, on, front, front	50%
2	on, front, on, front	75%
3	on, front, front, on	50%
4	front, front, on, on	50%
5	front, on, front, on	75%
6	front, on, on, front	50%

2.4 Interpretation of the Results Using Maximum Activation

The activations of units in the sequencer model as well as the sequence–associator model, have so far been interpreted using the normalized activation scheme stated by equation (2.3). As mentioned previously, this scheme is based on Luce's (1959) choice model and has been successfully applied to modeling of cognitive tasks such as word perception and speech perception (McClelland & Rumelhart, 1988). In more recent analysis, McClelland (1991) has observed that the normalized interpretation leads to a distortion in the correspondence between interactive activation neural network models that have been used in studying these tasks, and classical models of perception such as the signal detection model and the Luce's choice model. He has attributed the distortion to competition and nonlinearity processes that act upon the units within a pool in the neural network models.

McClelland (1991) has further observed that if a unit with the highest activation in its pool is chosen as the response choice among the many alternatives, then the distortion in the correspondence between the classical and neural network models is eliminated. In light of

the new study, we reinterpret the results of sequence generator and sequence–associator models using maximum activation as the response criterion. A unit in a pool is deemed to be correctly recalled if (i) its activation exceeds the firing threshold, (ii) it has the highest activation in its pool, and (iii) it meets conditions (i) and (ii) at the appropriate time interval during recall.

Figures 2.7a and 2.7b show recall of instructions using the new interpretation in the absence and presence of nonspecific arousal respectively. Figure 2.7a indicates that in the absence of nonspecific arousal, the activations of second and third instruction units are approximately equal in the third cycle. Also, the activations of third and fourth instruction units are approximately equal in the fourth cycle. Figure 2.7b indicates that the activations of all instructions are discriminated from each other by the presence of nonspecific arousal. However, the recall accuracy for instructions both in the absence and presence of nonspecific arousal is 100%. When these results are compared to the previous results obtained using the normalized activation interpretation (refer to Figures 2.3b and 2.3c), it may be noted that the newer results show weaker recency effect and stronger primacy effect. This is due to the fact that the normalized activation interpretation has the effect of smoothing caused by the averaging of activations whereas the "maximum activation interpretation" does not require averaging and, therefore, does not show smoothing effect.

Using the maximum activation interpretation, the recall of objects in the absence of nonspecific arousal is 25% (refer to Figure 2.8a). Even though the pattern of activation exhibits recall similar to that in the normalized activation interpretation (refer to Figure 2.5a), Figure 2.8a indicates that many of the activations fall below the firing threshold of 0.1 which leads to the low accuracy rate. During recall, the nonspecific arousal signal boosts the activations of most of the object units above the firing threshold and causes a strong recency effect (Figure 2.8b). The accuracy of recall of objects in this case is 50%. This outcome is analogous to that obtained using the normalized activation interpretation (refer

to Figure 2.5b). The results for targets using the maximum activation interpretation are identical to those for objects because of the fact that equal amounts of external inputs are used in training both objects and targets.

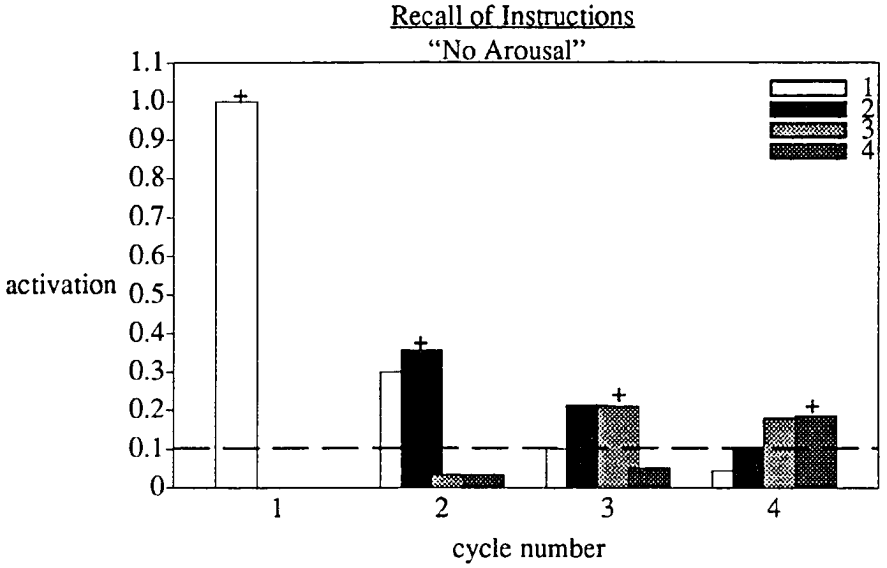


Figure 2.7a. Recall of instruction units in the sequence generator model in the absence of a nonspecific arousal signal. The model is interpreted using the maximum activation scheme. The dashed line indicates the firing threshold, "+" above a bar indicates correct recall of corresponding entity and "–" indicates incorrect recall.

The recall of prepositions in the absence of nonspecific arousal when the combination of (on, in front of, in front of, on) is used in the instruction presentation phase is shown in Figure 2.9a.

The order of recall as indicated by the pattern of activation, which is evidenced in this figure, is similar to that in the previous interpretation (compare to Figure 2.6a). However, recall accuracy in the current interpretation is 50% compared to 75% in the previous interpretation because the activation of "in front of" unit falls below the firing threshold in cycle two. Overall, the network recalls prepositions with an accuracy of 50% to 75% in the absence of nonspecific arousal, as summarized in Table 2.4. The exact amount of accuracy depends on the order of prepositions used in training the network.

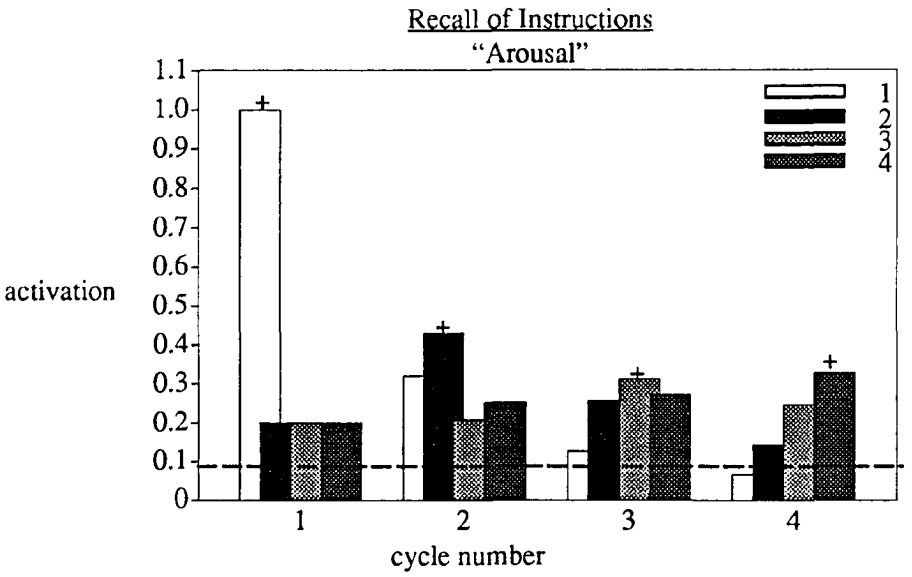


Figure 2.7b. Recall of instruction units in the sequence generator model in the presence of a nonspecific arousal signal. Refer to Figure 2.7a for an explanation of notation.

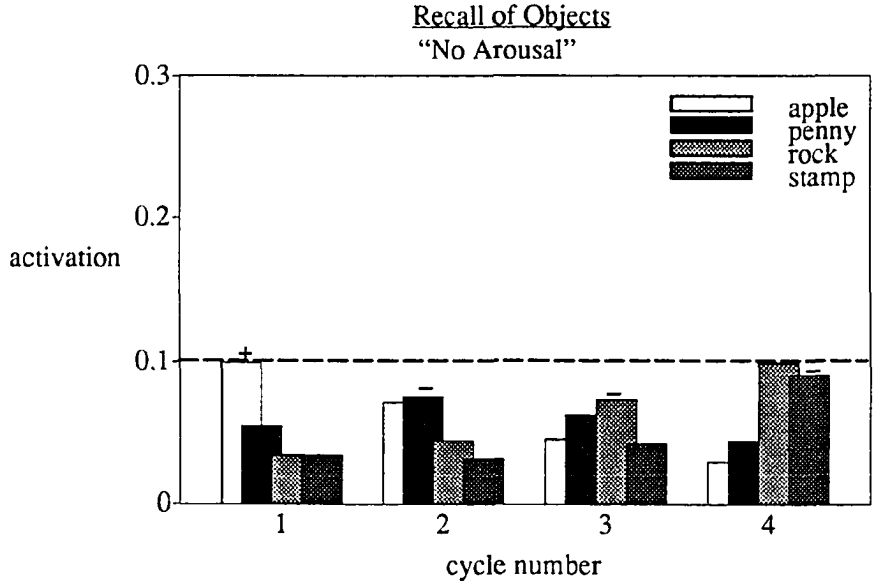


Figure 2.8a. Recall of objects in the sequence–associator model in the absence of a nonspecific arousal signal. The model is interpreted using maximum activation scheme. Refer to Figure 2.7a for further explanation of notation.

The recall of prepositions in the presence of nonspecific arousal for the combination of (on, in front of, in front of, on) is shown in Figure 2.9b. The order of recall as indicated by

the pattern of activation in this figure is similar to that in the previous interpretation (compare Figure 2.9b to Figure 2.6b). The network recalls this particular combination of prepositions with an accuracy of 50%. Overall, the recall accuracy varies from 50% to 75%, as summarized in Table 2.5, based on the order of prepositions used in training the network.

2.5 Discussion and Conclusions

The sequence generator neural network that we have presented self–organizes to store and recall a sequence of instructions in the order in which they were presented. The postsynaptic threshold term in the learning rule plays an important role in attenuating backward associations. This term, when later used in the hybrid sequence–associator neural network, offers greater flexibility in the control of associations than the presynaptic threshold term alone as used in the Hebbian learning rule.

Table 2.4. Recall of prepositions in the sequence–associator neural network when a nonspecific arousal signal is absent. Evaluation of recall is based on maximum activation scheme.

Number	Combination	Accuracy
1	on, on, front, front	75%
2	on, front, on, front	50%
3	on, front, front, on	50%
4	front, front, on, on	75%
5	front, on, front, on	50%
6	front, on, on, front	50%

The associations between instructions and item units are learned in a single time step in contrast to many time steps that are needed to learn the temporal associations among consecutive instructions. Our observation regarding the difficulty of learning temporal associations compared to spatial associations agrees with the work of other researchers. Sompolinsky and Kanter (1986) and Kleinfeld (1986) in accordance with this observation make use of fast and slow connections for automatic generation of temporal sequences.

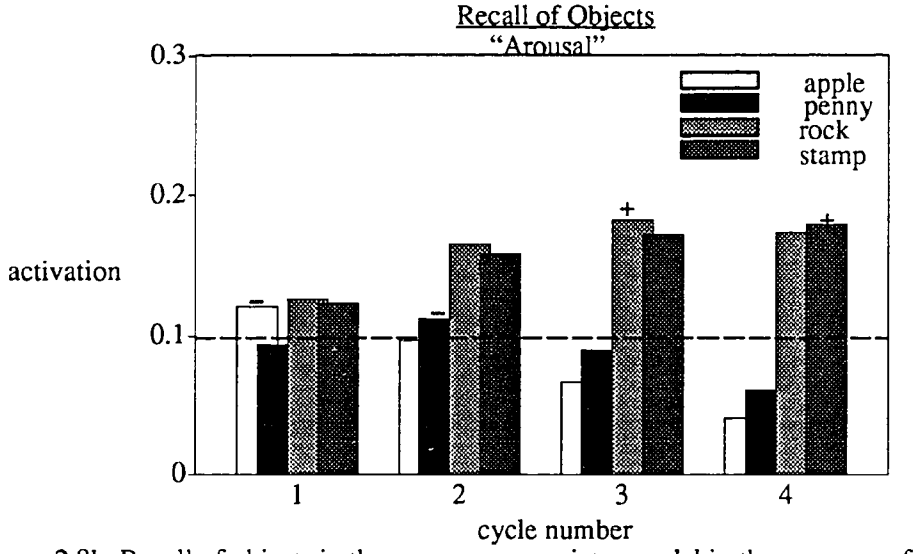


Figure 2.8b. Recall of objects in the sequence–associator model in the presence of a nonspecific arousal signal. The model is interpreted using maximum activation scheme. Refer to Figure 2.7a for further explanation of notation.

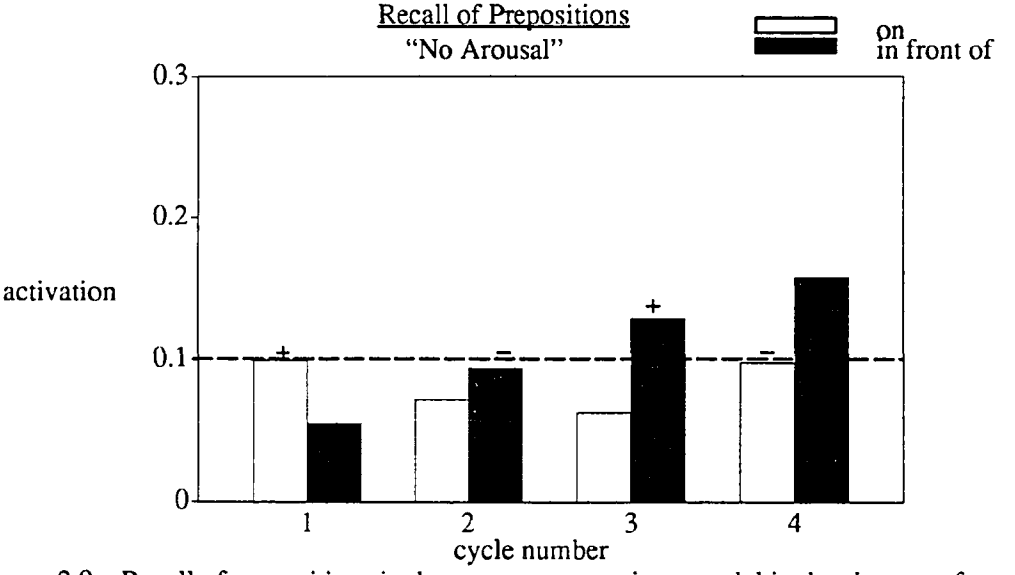


Figure 2.9a. Recall of prepositions in the sequence–associator model in the absence of a nonspecific arousal signal. This particular recall is obtained after training the network with an instruction sequence that contained the combination of (on, in front of, in front of, on). The model is interpreted using maximum activation scheme. Refer to Figure 2.7a for further explanation of notation.

The recall accuracy of instructions in the absence of nonspecific arousal in both readout schemes is 100%. The presence of arousal causes strong recency effect in activation levels in both schemes. This effect, in turn, results in a drop in accuracy to 75% in the normalized activation scheme. However, the recency effect enhances discrimination of instructions in the maximum activation scheme and the recall accuracy remains at 100%.

The normalized activation scheme resulted in an accuracy of 75% for objects and targets, and of 75 – 100% for prepositions in the absence of nonspecific arousal signal. The maximum activation scheme resulted in an accuracy of 25% for objects and targets, and of 50 – 75% for prepositions under the same condition. The lower accuracy that resulted from using maximum activation scheme is explained by the fact that the network parameters in this scheme were not tuned for best performance. The same set of parameters that were used in maximum activation scheme were kept constant for comparative study. As a result, many activations did not exceed the firing threshold and thus lower recall accuracy was obtained in the maximum activation scheme.

Table 2.5. Recall of prepositions in the sequencer–associator neural network when a nonspecific arousal signal is present. Evaluation of recall is based on maximum activation scheme.

Number	Combination	Accuracy
1	on, on, front, front	75%
2	on, front, on, front	75%
3	on, front, front, on	50%
4	front, front, on, on	75%
5	front, on, front, on	75%
6	front, on, on, front	50%

Thus, based on the computer simulations in this chapter, we could not conclude whether one readout scheme is better than the other. However, as discussed by McClelland (1991) in detail, the normalized activation scheme is based on classical perception models and is not well–suited for interpreting the output of neural network models if the units in them

involve competition and nonlinearity processes. For this reason, we adopt exclusively the maximum activation scheme for interpreting the results of neural network models on strategy selection and evolution to be presented in Chapters three and four.

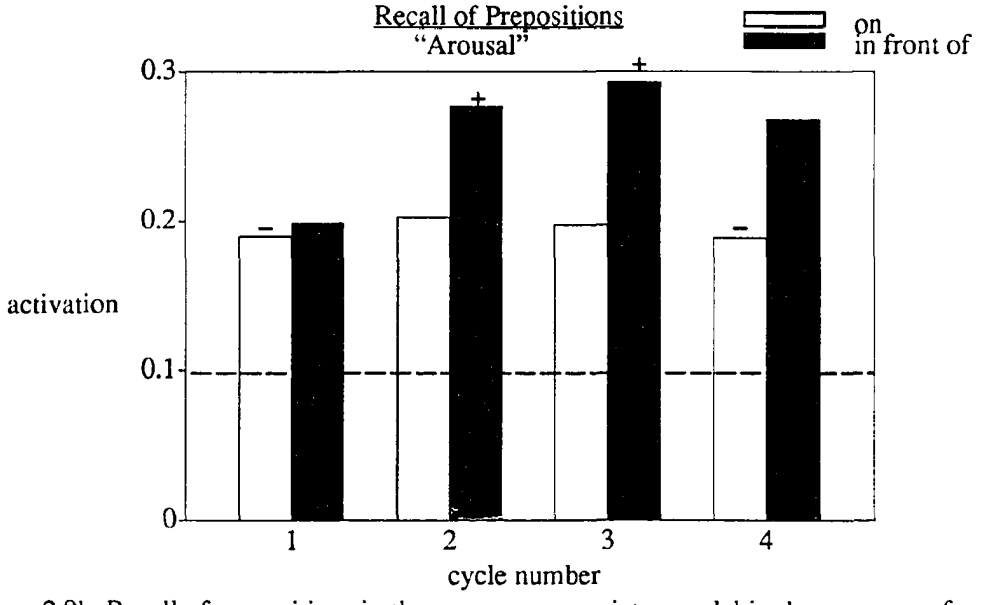


Figure 2.9b. Recall of prepositions in the sequence–associator model in the presence of a nonspecific arousal signal. This particular recall is obtained after training the network with an instruction sequence that contained the combination of (on, in front of, in front of, on). The model is interpreted using maximum activation scheme. Refer to Figure 2.7a for further explanation of notation.

Priming in the sequence generator network causes strong primacy effect on the recall of instructions, whereas it causes a somewhat diminished primacy effect on the recall of objects, targets, and prepositions in the hybrid sequence–associator network.

Nonspecific arousal has the effect of uniformly boosting activations of all units, which otherwise have diminished peaks toward the end of recall. It causes strong recency effects in both models. The recency effect is so strong in the hybrid sequence–associator model that it interferes with correct recall of initial items (see Figures 2.5b, 2.6b, 2.8b, and 2.9b). For this reason, the nonspecific arousal mechanism is not invoked in extensions of the sequence–associator neural network, which are undertaken in Chapters three and four for the purpose of studying strategy development.

The hybrid model in its present status neglects the time delays between the presentations of objects and prepositions and targets that are evident in the experiment. It offers a framework for analyzing the role of external strategies on memory recall. The hybrid of models for previously learned and currently learned information provides a promising framework for further development of cognitive and/or neural models for humans and robots.

CHAPTER 3

A NOVELTY BIAS NEURAL NETWORK MODEL OF STRATEGY SELECTION AND EVOLUTION

In this chapter, we extend the hybrid sequence–associator model presented in Chapter two to account for strategy selection and evolution behavior in children in the "object–target matching task." The model is referred to as the "novelty bias neural network" and contains three new components: strategy selection, accuracy, and novelty bias. The choice of these additional components is based on Siegler's theory of strategy selection and evolution, which postulates that strategy development is controlled by factors such as accuracy, speed, and novelty bias. We first elaborate on the relevance of Siegler's theory to the development of the novelty bias neural network model. We follow this with a discussion of the architecture of the model. Last, we present a discussion of computer simulations and compare outcomes of the simulations with observed strategy behavior in children.

3.1 Motivation

Children use a variety of strategies in the object–target matching task described in Chapter two. Examples of some commonly used strategies are pointing at an object, moving an object with orientation toward a target, and placing an object in front of or on top of a wooden separator directly across from a target. All these strategies may be incorporated into three categories: object encoding only, object–target encoding, and object–target–preposition encoding. The investigations of Bray et al. (1993) show that older children use "object–target encoding" and/or "object–target–preposition encoding" more frequently than younger children. Younger children tend to use "object encoding only" most

frequently and, to a lesser degree, "object–target encoding." Even though Bray et al. (1993) observe that the subjects use a combination of these strategies in a given instruction sequence, we assume in our computer simulations that they use a single strategy throughout the instruction sequence. In a broader context, we are interested in identifying the various cognitive mechanisms that mediate such developmental differences in strategy use.

Siegler and his colleagues empirically determined that for a given task a child will use multiple strategies. This is true in a number of different tasks such as addition, subtraction, multiplication, reading, time telling, and serial recall (McGilly & Siegler, 1989; Siegler & Jenkins, 1989; Siegler, 1991). According to their findings, children switch from one strategy to another on a given problem even though a particular strategy may work well on that problem. Siegler (1991) proposes that cognitive factors such as accuracy of outcome, speed of execution, and novelty of strategy are responsible for the variability in strategy selection and for the evolution of strategies. Strategies that result in higher accuracy, those that are faster and those that are newer are generally preferred over others, but there is variability in strategy selection, and the evolution of strategy selection is gradual.

We hypothesize that factors similar to these may also be responsible for the selection and evolution of strategies in the object–target matching task. A model constructed on this hypothesis should explain the differences in strategy choices of different age groups on this task. We turn to neural network models because they exhibit the characteristics of adaptability, generalization, and neurological plausibility. In the current effort, we construct a neural network model based on the assumption that accuracy and novelty are two key cognitive factors that play a role in the selection and evolution of strategies in the object–target matching task.

3.2 Architecture of the Neural Network

Figure 3.1 illustrates the various functional components in the neural network model for the selection and evolution of strategies in the object–target matching task and the

interactions among them. A sequence generator is responsible for storing and recalling a

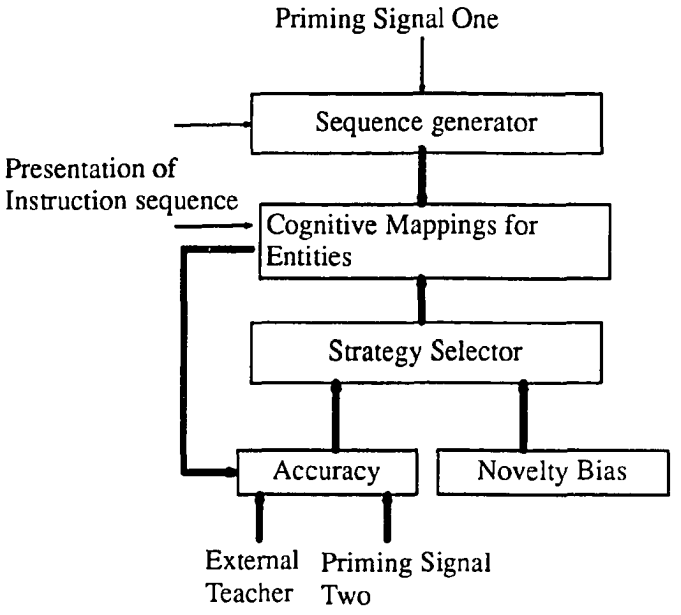


Figure 3.1. Block diagram of the novelty bias neural network model. The various neural components involved in strategy selection and evolution in the object–target matching task are illustrated here. The bold arrows represent multiple inputs/ connections from/to the components. The light arrows represent single inputs.

sequence of instructions. It is assumed that a subject has become familiar with the concept of seriality before attempting the matching task. Hence a neural network which has already been trained on seriality and thus embodies knowledge of seriality is used as a sequence generator.

Cognitive mappings are representations of the objects, targets, and prepositions used in the experiment. An outstar–like associator couples the sequence generator with cognitive mappings for the entities. Construction of the sequence generator and associator is motivated by Grossberg's work on serial recall and the outstar (Grossberg, 1978). Further details of the model for the storage and recall of instruction sequences without the strategy component are discussed in Chapter two.

The strategy selector contains multiple strategies, each of which receives a bias for novelty and accuracy input based on previous experience of the network with the matching

task. An external teacher provides the network with correct answers during the recall of instruction sequences in each trial. A priming signal stimulates the accuracy component to prompt the selection of a strategy. Another priming signal, corresponding to the cue to recall in Bray et al. (1993), stimulates the sequence generator to prompt the recall of an instruction sequence after its presentation.

The specifications and equations that govern the various neurons which constitute the functional components in the neural network are described below. The neural network is illustrated in Figure 3.2.

3.2.1 Instruction Units

The sequence generator consists of four instruction units corresponding to the number of instructions presented in each trial. Instruction units are connected to every other instruction unit and send excitatory connections to the entity units. As mentioned above, the sequence generator component has been trained separately to control a sequence. Training results in strong forward associations being established among the instruction units. Henceforth these associations remain fixed. The first instruction unit receives a priming signal at the beginning of recall. Further details on training and recall are discussed in Chapter two.

3.2.2 Entity Units

Cognitive mappings for entities are represented by entity units. There are three entity pools, one each for objects, targets, and prepositions. The object pool has 12 object units; the target pool, 6 target units; and the preposition pool, 2 preposition units. These numbers correspond to the actual numbers of objects, targets, and prepositions used in the experiment of Bray et al. (1993). Each entity unit receives adjustable excitatory connections from all instruction units, and fixed inhibitory connections from other entity units within its pool. Units in the object pool also receive excitatory connections from all strategy units. Units in the target pool receive adjustable excitatory connections from strategy units two and three

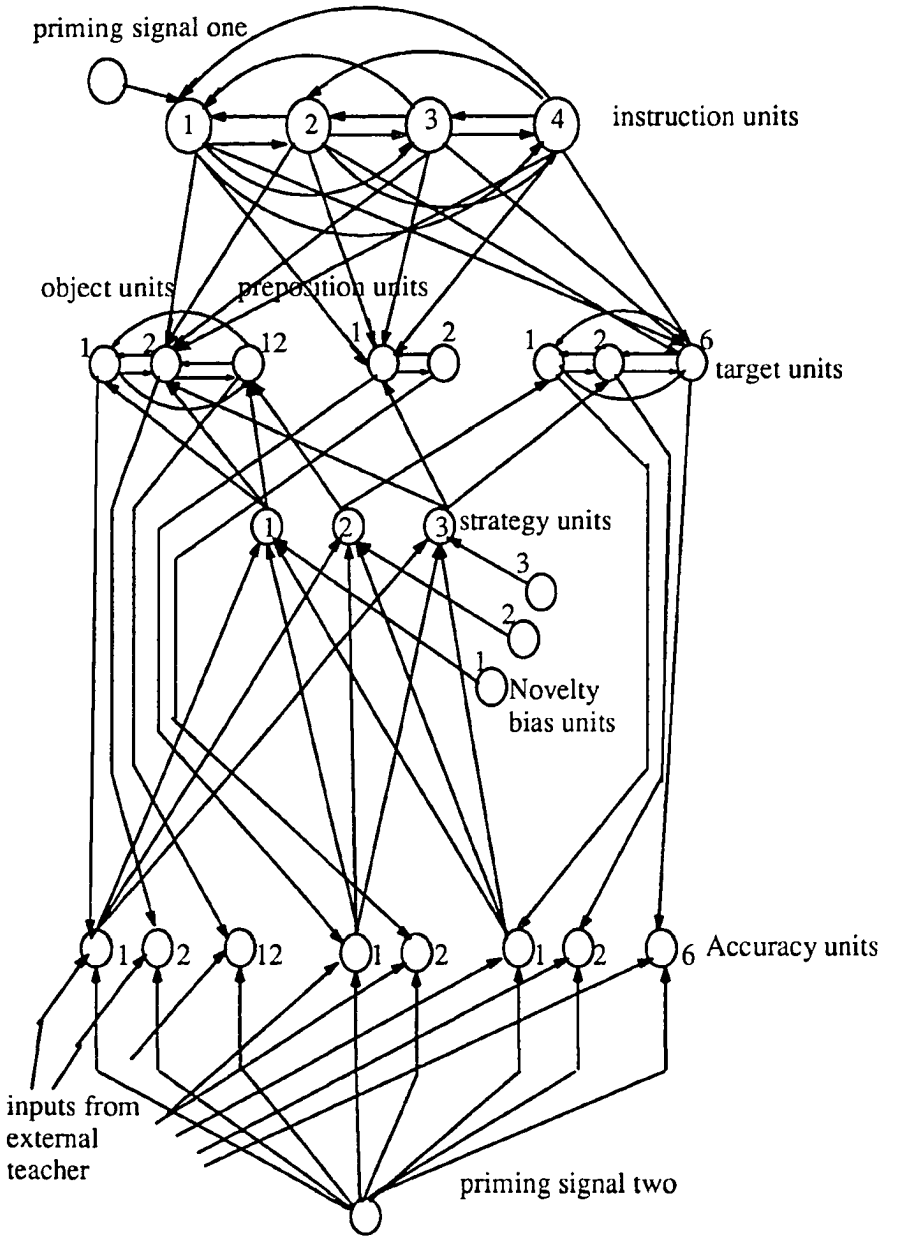


Figure 3.2. Architecture of the novelty bias neural network model for strategy selection and evolution. For the sake of clarity, only a few units and connections are shown here. All instruction units send excitatory connections to all object, preposition, and target units ("entity units"). All accuracy units send excitatory connections to all strategy units. Each entity unit sends an excitatory connection to a corresponding accuracy unit. All instruction units send excitatory connections to every other instruction unit. Entity units and strategy units receive inhibitory connections from all other units within their pool. See text for further details.

only. Units in the preposition pool receive adjustable excitatory connections from strategy unit three only. The rationale for such selective connectivity comes from the fact that strategy one encodes objects only, strategy two encodes objects and targets, and strategy three encodes all three entity classes.

3.2.3 Strategy Units

The strategy selector consists of three strategy units. The first unit represents the object encoding only strategy, the second unit the object–target encoding strategy, and the third unit the object–target–preposition encoding strategy. Each strategy unit receives an excitatory connection from a corresponding novelty bias unit and adjustable excitatory connections from all accuracy units, which are described below. In the current implementation, the strategy unit with the highest activation retains its value and the activations of other units in the pool are set to zero. It would be possible to accomplish this using a "winner take all" mechanism such as lateral inhibition within the pool.

3.2.4 Novelty Bias Units

The novelty bias component consists of three novelty bias units. The connection from a novelty bias unit to its strategy is assigned a random weight each time the activations of strategies are updated, reflecting the probabilistic nature of strategy selection. The activation of a novelty bias unit starts at a given value and decays with time. It represents the degree of novelty a subject may place on a strategy. A higher activation value indicates that the strategy corresponding to the bias unit is newer and is strongly biased in favor of selection, and vice–versa.

3.2.5 Accuracy Units

The accuracy component consists of three pools of units. The first pool contains 12 units, the second 6 units, and the third 2 units. Thus, there is a one–to–one correspondence between accuracy units and entity units. Each accuracy unit receives an excitatory connection from a corresponding entity unit. It also receives input from an external teacher

during recall. It sends out its excitatory connections to all strategy units. An accuracy unit shunts the input from an external teacher with excitatory input from its entity unit.

3.2.6 Activation and Weight Update

The activations of all units except accuracy units are updated using a general equation (3.1). This equation contains a decay term, a net excitatory input term, a net inhibitory term, and an external input term. The activation update for accuracy units takes place according to equation (3.2), which contains a shunting term and a term for priming strategy selection.

The adjustable weights are updated using equation (3.3), which is motivated by the dual learning rule proposed by Edelman (1987) in his theory of neuronal group selection. This equation is a generalized version of Hebbian learning, and synaptic weights may increase or decrease based on presynaptic and postsynaptic thresholds.

Activation update equation:

$$a_i(t+1) = (1-\alpha)\, a_i(t) + \beta \sum_j [a_j(t)]^{\Gamma_1}\, w_{j\,i}(t) + \gamma \sum_k q\, [a_k(t)]^{\Gamma_1} + I_i(t) \qquad (3.1)$$

where

$a_i(t)$ = *activation of unit i at time t*
α = *decay rate*
β = *net input factor*
$w_{ji}(t)$ = *connection strength from unit j to item unit i at time t*
γ = *net inhibition factor*
q = *fixed inhibitory strength among units of the same pool*
Γ_1 = *presynaptic threshold*
$[x]^{\Gamma_1}$ = x *only* if $x \geq \Gamma_1$
= 0.0 *otherwise*
$I_i(t)$ = *external input to unit i at time t.*

Activation update equation for accuracy units:

$$a_i(t+1) = a_k(t) * E_i(t) + I_i(t) \qquad (3.2)$$

where

$a_i(t)$ = *activation of accuracy unit i at time t*
$a_k(t)$ = *activation of corresponding entity unit k at time t*
$E_i(t)$ = *input from external teacher to accuracy unit i at time t*
$I_i(t)$ = *priming signal to accuracy unit i at time t*

Weight update equation:

$$w_{i\,j}(t+1) = w_{i\,j}(t) + \delta\,[a_i(t+1)]^{\Gamma_2}\,[a_j(t+1)]^{\Gamma_3} \qquad (3.3)$$

where

$w_{ij}(t)$ = *connection strength from unit i to item unit j at time t.*
$a_i(t+1)$ = *activation of unit i at time* $t+1$
$a_j(t+1)$ = *activation of unit j at time* $t+1$
Γ_2, Γ_3 = *presynaptic* and *postsynaptic thresholds.*
δ = *learning rate*

Table 3.1. Parameters used in computer simulation of the novelty bias neural network model.

$\alpha = 0.9,\ \beta = 0.7,\ \gamma = 0.2,\ q = -0.5,\ \Gamma_1 = 0.1;$
for instructions to entity connections :
$\Gamma_2 = 0.1,\ \Gamma_3 = 0.4,\ \delta = 0.4;$
for strategy to entity connection:
$\Gamma_2 = 0.1,\ \Gamma_3 = 0.4,\ \delta = 0.2;$
for accuracy to strategy connections :
$\Gamma_2 = 0.1, \Gamma_3 = 0.2;$
when accuracy unit $\geq \Gamma_2 : \delta = 0.2;$
when accuracy unit $< \Gamma_2 : \delta = -0.075;$
Priming signal one = 1., priming signal two = 0.15;
Initial weights for all connections are set at 0.05;

3.3 Computer Simulation

Computer simulations of the neural network model have been carried out to evaluate its utility as a predictor of strategy selection and strategy evolution in children. The simulation consists of several trials, each made up of a presentation phase followed by a recall phase. Multiple trials in the computer simulation are analogous to multiple learning epochs, each involving exposure by children to tasks similar to object–target matching in everyday life, which elicits changes in strategy choice in them. The current simulation is intended to model the evolution of strategy selection across a number of years in a child's life, and therefore, is not to be confused with the multiple trials that are presented to children in the course of the experiment performed by Bray et al. (1993).

In the presentation phase of the simulation, a priming signal is given to accuracy units to prompt selection of a strategy. An instruction sequence is presented starting the next cycle. Whenever an instruction is presented, the proper instruction unit, the object unit, the target unit, and the preposition unit each receive an external input. A preposition, e.g., "on," is uniquely used exactly twice in an instruction sequence in order to maintain a correspondence with the experiment of Bray et al. (1993). Each presentation cycle consists of an update of the activations of all units in the network followed by an update of the connection strengths from instruction units to entity units and from strategy units to entity units. All other connection strengths remain fixed during the presentation phase.

In the recall phase, a priming signal is given to the first instruction unit to prompt the recall of instruction sequence. As the recall of entities takes place, a constant external input from the teacher is given to accuracy units. This input is positive if the entity corresponding to that accuracy unit should be on in the current recall cycle and negative otherwise. Activations of all units in the network and connection strengths from accuracy units to strategy units are updated in that order in each recall cycle.

It should be noted here that learning occurs in the network in both presentation and recall phases. The associations between a particular instruction sequence and a strategy are learned in the presentation phase. Feedback concerning accuracy is provided by the teacher to the accuracy units during the recall phase. It is assumed that sufficient time elapses between trials so that the associations learned between a strategy and a particular instruction sequence during a presentation phase decay to resting value and, therefore, do not cause residual associations during the next trial. However, the associations learned between accuracy units and strategy units have negligible decay so that they last the entire life span of the network.

At the beginning of the computer simulation, the novelty bias unit, N1, which excites strategy unit, S1, receives an external input. This input leads to the selection of strategy one of object encoding only over the others. We henceforth refer to the process of preferentially

setting the external input to a bias unit as "strategy initiation." This corresponds to a subject's "decision" to try a new type of strategy. After a few trials (12 in this sample simulation), strategy two of object–target encoding is initiated. After a few more trials (12 in this sample simulation), strategy three of object–target–preposition encoding is initiated. The rationale for such orderly initiation of strategies is that the first strategy of object encoding only is the most basic of all strategies and the first to occur in the development of external strategies in children. Strategy two is discovered after strategy one and strategy three after strategy two by most children.

The computer simulation is terminated when any one of the strategy units reaches its maximum activation value (in the current implementation, 1.0). Based on experience with the computer simulations of the model, we know that when this criterion is satisfied, the network has completed the process of strategy evolution and has converged on the strategy with the maximum activation.

3.4 Results

The results of a typical simulation run are presented in Table 3.2 which lists net bias input and net accuracy input to each strategy, the winning strategy and its activation for every trial for 50 trials. "Net bias input" to a strategy unit is defined as the net influence of its bias unit in a given trial. "Net accuracy input" to a strategy unit is defined as the net influence of all accuracy units to the strategy unit when a priming signal is given to accuracy units.

After "object encoding only" strategy was initiated at the beginning of the simulation, it continues, being strongly biased over other strategies, to be selected until trial twelve. A steady increase in its net accuracy input is noticeable.

After object–target encoding strategy was initiated at the beginning of trial thirteen, the selection shifts between strategy one and strategy two until trial twenty–four. A steady increase in the net accuracy input is noticeable for both strategy one and strategy two.

Table 3.2 Strategy evolution observed in one of the computer simulation runs of the novelty bias neural network model.

Tr	Strategy 1		Strategy 2		Strategy3		SS	ASS
	NB	AI	NB	AI	NB	AI		
				< Strategy one is initiated >				
1	0.682	0.150	0.000	0.150	0.000	0.150	1	0.832
2	0.431	0.169	0.000	0.150	0.000	0.150	1	0.600
3	0.269	0.182	0.000	0.150	0.000	0.150	1	0.450
4	0.277	0.191	0.000	0.150	0.000	0.150	1	0.468
5	0.412	0.200	0.000	0.150	0.000	0.150	1	0.613
6	0.280	0.217	0.000	0.150	0.000	0.150	1	0.496
7	0.381	0.227	0.000	0.150	0.000	0.150	1	0.607
8	0.332	0.240	0.000	0.150	0.000	0.150	1	0.572
9	0.067	0.252	0.000	0.150	0.000	0.150	1	0.319
10	0.166	0.258	0.000	0.150	0.000	0.150	1	0.424
11	0.202	0.267	0.000	0.150	0.000	0.150	1	0.468
12	0.388	0.276	0.000	0.150	0.000	0.150	1	0.664
				< Strategy two is initiated>				
13	0.130	0.297	0.614	0.150	0.000	0.150	2	0.764
14	0.252	0.297	0.080	0.177	0.000	0.150	1	0.549
15	0.010	0.310	0.572	0.177	0.000	0.150	2	0.748
16	0.074	0.310	0.300	0.205	0.000	0.150	2	0.505
17	0.029	0.310	0.067	0.221	0.000	0.150	1	0.340
18	0.097	0.317	0.550	0.221	0.000	0.150	2	0.771
19	0.094	0.317	0.411	0.248	0.000	0.150	2	0.659
20	0.046	0.317	0.172	0.278	0.000	0.150	2	0.450
21	0.190	0.317	0.425	0.288	0.000	0.150	2	0.712
22	0.229	0.317	0.414	0.316	0.000	0.150	2	0.730
23	0.170	0.317	0.098	0.343	0.000	0.150	1	0.487
24	0.030	0.327	0.111	0.343	0.000	0.150	2	0.454
				< Strategy three is initiated >				
25	0.073	0.327	0.093	0.353	0.584	0.150	3	0.734
26	0.169	0.327	0.097	0.353	0.160	0.197	1	0.496
27	0.023	0.337	0.074	0.353	0.615	0.197	3	0.812
28	0.159	0.337	0.102	0.353	0.314	0.252	3	0.567
29	0.147	0.337	0.001	0.353	0.164	0.276	1	0.484
30	0.057	0.347	0.284	0.353	0.131	0.276	2	0.637
31	0.050	0.347	0.172	0.381	0.380	0.276	3	0.656
32	0.008	0.347	0.227	0.381	0.197	0.320	2	0.608
33	0.011	0.347	0.097	0.396	0.096	0.320	2	0.493
34	0.111	0.347	0.058	0.408	0.389	0.320	3	0.709
35	0.045	0.347	0.139	0.408	0.026	0.366	2	0.546
36	0.067	0.347	0.118	0.424	0.046	0.366	2	0.543
37	0.008	0.347	0.146	0.441	0.163	0.366	2	0.587
38	0.093	0.347	0.061	0.456	0.259	0.366	3	0.626
39	0.069	0.347	0.182	0.456	0.080	0.407	2	0.637
40	0.025	0.347	0.053	0.483	0.192	0.407	3	0.599
41	0.003	0.347	0.064	0.483	0.141	0.434	3	0.574
42	0.012	0.347	0.062	0.483	0.305	0.459	3	0.765
43	0.061	0.347	0.107	0.483	0.132	0.508	3	0.641
44	0.064	0.347	0.090	0.483	0.197	0.549	3	0.747
45	0.036	0.347	0.010	0.483	0.172	0.600	3	0.771
46	0.025	0.347	0.063	0.483	0.240	0.650	3	0.890
47	0.010	0.347	0.108	0.483	0.163	0.706	3	0.870
48	0.007	0.347	0.079	0.483	0.156	0.761	3	0.917
49	0.051	0.347	0.002	0.483	0.078	0.822	3	0.900
50	0.024	0.347	0.028	0.483	0.131	0.880	3	1.000

Tr: Trial Number, NB: Net novelty Bias, AI: net Accuracy Input, SS: Strategy Selected, ASS: Activation of Strategy Selected. Net novelty Bias = (random number between 0.0 and 1.0) * (activation of bias unit); net Accuracy Input = Σ Priming Signal two * activation of accuracy unit A strategy was initiated by setting the activation of its novelty bias unit to 0.8.

After object–target–preposition encoding strategy was initiated at the beginning of trial twenty–five, all strategies were used at least twice. This behavior of the network of using multiple strategies in close time intervals is similar to strategy selection by children in arithmetic, serial recall, time telling and other task domains (McGilly & Siegler, 1989; Siegler & Jenkins, 1989; Siegler, 1991).

Even though the net accuracy input for the object encoding only strategy unit, for example, at trial twenty–eight is greater than that for the object–target–preposition encoding strategy unit, the novelty factor allows the latter strategy to be selected. As the simulation continues, the net novelty bias input diminishes for all strategies and net accuracy input becomes the deciding factor in strategy selection. Since the object–target–preposition encoding strategy gains the highest net accuracy input after trial forty–three, when the net novelty biases for other strategies have nearly diminished to zero, it becomes the only strategy selected by the network. This behavior is similar to most adults always selecting a "retrieval" strategy over a "count from one" strategy or a "count from smaller addend" strategy in simple addition as in Siegler and Jenkins (1989).

Due to randomness in the contribution of novelty bias units to the strategy units, the results of strategy evolution and recall accuracy vary with each simulation run. The results from 11 simulation runs are tabulated in Appendix A. These results indicate that strategies, in general, advance from the object encoding only type to the object–target encoding type to the object–target–preposition encoding type and the network converges to the object–target–preposition encoding strategy. However, some simulation runs indicate that the network occasionally converges to the object–target encoding strategy. Simulation run eleven listed in Appendix A demonstrates this kind of behavior. The examination of the simulation runs reveals that the randomness in the contribution of novelty bias units is responsible for the differences in strategy convergence patterns.

We observe from Appendix A that the number of trials when the activation of any strategy reaches its peak value of 1.0 varies from one simulation run to another due to randomness in the contribution of novelty bias. The number of runs when the network converged on the object–target–preposition encoding strategy varies from 46 in simulation run five to 65 in simulation run ten. The network converges on the object–target encoding strategy in 79 trials as indicated by simulation run eleven.

The network is tested for recall accuracy of objects, targets, and prepositions using minimum and maximum activations of strategy units which are obtained from simulation runs one to eleven listed in Appendix A. The object encoding only strategy has a minimum activation of 0.190 and a maximum activation of 0.9, the object–target encoding strategy of 0.304 and 1.0, and the object–target–preposition encoding strategy of 0.431 and 1.0 respectively. The testing for recall accuracy included all possible combinations of prepositions.

Activations of entities with regard to recall are interpreted using the maximum activation scheme, discussed in Chapter two. To review, recall of an entity is deemed correct if its activation exceeds the firing threshold of 0.1 (same as all other units in the network), its activation is the highest in its pool, and it fires in the order of its training. Recall accuracy of objects, for example, is determined by the ratio of the number of objects correctly recalled to the number of objects used in the sequence of four instructions.

Graphs S0.1, S0.2, and S0.3 in Figures 3.3, 3.5, and 3.7, respectively, illustrate that without any strategy use, activations of all strategy units equal zero and only first entity units exceed the firing threshold of 0.1. Recall accuracy for objects, targets, and prepositions is 25% each in this case.

With the use of an object encoding only strategy, the object units receive a boost in their activation value (refer to graph S1.1 in Figures 3.3 and 3.4). However, when the activation of the strategy is at a minimum (i.e., 0.190), the boost is not significant enough to raise the

activations of object units above the firing threshold except that of the first object unit (refer to graph S1.1 in Figure 3.3). When the activation of the strategy is at its maximum (i.e., 0.900), the boost raises the activations of all object units above the firing threshold (refer to graph S1.1 in Figure 3.4). Thus, the recall accuracy for objects using the object encoding only strategy varies from 25% to 100%. The activations for target units and preposition units using the object encoding only strategy (graphs S1.2 and S1.3, respectively, in Figures 3.5, 3.6, 3.7, and 3.8) are the same as those without any strategy use. The lack of a boost in activations of these units is explained by the fact that they do not receive any excitatory connections from strategy one unit. The recall accuracy for targets and prepositions is 25% each.

When the activation of the object–target encoding strategy is at its minimum (i.e., 0.304), the activations of objects, targets and prepositions during recall are illustrated in graphs S2.1, S2.2 and S2.3, respectively (refer to Figures 3.3, 3.5, and 3.7). Even though the object and target units receive a boost in their activation, at its minimum, it does not raise the activations above the firing threshold. When the activation of the object–target encoding strategy is at its maximum (i.e., 1.0), the activations of objects and targets exceed the firing threshold in correct order, as illustrated in graphs S2.1 and S2.2, respectively (see Figures 3.4 and 3.6). Thus, the recall accuracy of objects and targets when the object–target encoding strategy is used varies from 25% to 100%. Since this strategy does not encode prepositions, its use does not have any effect on the recall accuracy of prepositions which remains at 25%.

Graphs S3.1 and S3.2 portray the effect of the use of the object–target–preposition encoding strategy on the recall of objects and targets, respectively (refer to Figures 3.3 and 3.5), when the activation of the strategy is at its minimum (i.e., 0.431). The activations of objects and targets receive a boost from the strategy but the boost is not significant enough to raise them above the firing threshold. Thus, the recall accuracy of objects and targets

remains at 25% each when activation of the object–target–preposition encoding strategy is at its minimum value.

Graph S3.3 in Figure 3.7 portrays the effect of the use of the object–target–preposition encoding strategy on the recall of the following combination of prepositions: (on, in front of , in front of, on) when the activation of the strategy is at its minimum (i.e., 0.431). Both prepositions receive a boost in their activations except the "on" unit in the last cycle of recall. Thus, the recall accuracy of the network on this particular combination of prepositions is 75%. We tested the network for recall accuracy on other possible combinations of prepositions. The use of the object–target–preposition encoding strategy, when its activation is at its minimum, has the effect of boosting the activations of prepositions in these combinations except in one cycle. Thus, the recall accuracy for any given combination of prepositions, when the activation of the strategy is at its minimum, is 75%.

Graphs S3.1 and S3.2 portray the effect of the use of object–target–preposition encoding strategy on the recall of objects and targets, respectively (refer to Figures 3.4 and 3.6), when the activation of the strategy is at its maximum (i.e., 1.0). The activations of all objects and targets receive a boost from the object–target–preposition encoding strategy unit to raise them above the firing threshold in the correct order. Thus, the recall accuracy of objects and targets is at 100% each when activation of the object–target–preposition encoding strategy is at its maximum value.

Graph S3.3 in Figure S3.8 portrays the effect of the use of the object–target–preposition encoding strategy on the recall of the following combination of prepositions: (on, in front of, in front of, on) when the activation of the strategy is at its maximum (i.e., 1.0). The graph indicates that the network recalls prepositions correctly in the first, second, and third instructions. However, the activation of "in front of" which is used in the second and third instructions persists and dominates that of "on" in the fourth cycle. The recall accuracy in this case is 75%. We tested the network for recall accuracy on all other combinations of

prepositions. The use of the object–target–preposition encoding strategy, when its activation is at its maximum, has the effect of boosting the activations to such a large extent that the boost interferes with correct order of recall. The recall accuracy for any given combination of prepositions is found to vary from 50% to 75% when the activation of the object–target–preposition encoding strategy is at its maximum.

As noted above, the object–target–preposition encoding strategy causes insufficient boost in prepositions when its activation is at its minimum (i.e., 0.431), thus effecting a recall accuracy of prepositions of 75%. Also, it causes a large boost that interferes with correct recall of prepositions when its activation is at its maximum (i.e., 1.0), thus effecting a recall accuracy of prepositions of 50–75%. Thus, the recall accuracy has improved over the no–strategy case when the object–target–preposition encoding strategy is at its minimum but has deteriorated when the object–target–preposition encoding strategy is at its maximum. For this reason, we would like to determine if the network recalls prepositions with 100% accuracy when the object–target–preposition encoding strategy takes on intermediate values. We have tested the network for recall accuracy of prepositions when the activation of the object–target–preposition encoding strategy takes on a value of 0.793. Indeed, the network exhibits a recall accuracy of 100% at this value of the object–target–preposition encoding strategy on all other combinations of prepositions. In summary, the accuracy of network on the recall of prepositions varies from 50% to 100% based on the activation of the object–target–preposition encoding strategy and on the order of prepositions used in the instruction sequence.

The recall performance of the novelty–bias neural network on objects, targets, and prepositions using various strategies is summarized in Table 3.3.

3.5 Discussion

Computer simulations of the object–target matching task demonstrate that strategies, in general, evolve from the simple object encoding only type strategy to the sophisticated

object–target–preposition encoding type strategy. This simulated behavior is in agreement with observed behavior in children: younger children frequently use strategies such as pointing to and holding objects, while older children frequently use strategies such as moving objects with orientation toward targets and with encoding of prepositions (Bray et al., 1993).

Table 3.3 Summary of performance of the novelty bias neural network on recall accuracy.

ENTITY	STRATEGY USED			
	NONE	1	2	3
OBJ	25%	25–100%	25–100%	25–100%
TARG	25%	25%	25–100%	25–100%
PREP	25%	25%	25%	50–100%

Occasionally, the network converged on the moderately advanced object–target encoding strategy rather than on the most advanced object–target–preposition encoding strategy. This type of behavior by the novelty bias neural network resembles that of some mentally retarded individuals who do not seem to advance to the more advanced strategies. The differences in types of strategy convergence as exhibited by the network are attributed to randomness in the novelty bias factor. The issue of how novelty bias and the concomitant randomness translate into a plausible biological mechanism is not addressed in the current model. Addressing this issue could eventually lead to a more satisfactory explanation of the differences in strategy evolution between nonretarded and mentally retarded individuals.

The computer simulations are consistent with the supposition that the accuracy information gained from previous experience with strategies and novelty bias are responsible for the evolution of strategies from the simple to the advanced. With the use of more advanced strategies, more accuracy input is gained for later use. For example, the activations of preposition units are higher with the use of object–target–preposition

encoding strategy than with the use of object–target encoding strategy because the former encodes prepositions and the latter does not. Consequently, preposition accuracy units receive higher excitation from corresponding preposition units with the object–target–preposition encoding strategy than with the object–target encoding strategy. Thus, in the long run, the accuracy units contribute more input to the object–target–preposition encoding strategy than to the object–target encoding strategy. This explains why the network, in general, takes a longer number of trials to converge on the object–target encoding strategy than on the object–target–preposition encoding strategy.

We noted in the results section that strategy use causes a boost in the activations of entity units. If the boost is significantly high, then the activations of the entity units exceed the firing threshold and correct recall of the entities ensues. Based on the minimum and maximum activations of the object only encoding strategy and the object–target encoding strategy observed in the computer simulations, the recall accuracy of the novelty bias neural network on objects and targets varies from 25% to 100%. Thus, at lower values of the activations of these strategies, the network does not exhibit any improvement in recall accuracy. With higher activations, however, the network exhibits 100% recall accuracy.

Based on the minimum and maximum activations of the object–target–preposition encoding strategy observed in the computer simulations, the recall accuracy of the network on prepositions varies from 50% to 100%. For lower activation values of the object–target–preposition encoding strategy, the recall accuracy is 75% and for medium activation values, the recall accuracy improves to 100%. For higher activation values of the object–target–preposition encoding strategy, the recall accuracy deteriorates to 50%. Analysis of the simulation results reveals that at higher values, spurious associations are established between incorrect preposition units and instruction units because the preposition units receive higher activation values that persist for longer cycles, due to increased stimulation from the object–target–preposition encoding strategy units. These spurious

associations lead to incorrect recall of prepositions. This anomaly in the recall of prepositions needs to be addressed in future work.

In the current implementation of the model, strategies act as a uniform boosting signal and do not increase recall discrimination within a class of entities – objects, for example. Future implementations of strategy mechanism should address this issue.

As far as we know, long–term empirical studies dealing with the discovery of various strategies in the same individuals have not yet been undertaken for the object–target matching task. So the prediction of the model that children evolve their strategies motivated by accuracy of outcome and novelty is yet to be tested empirically. The neural network model suggests that cognitive factors such as accuracy of outcome and novelty of strategy are plausible in the object–target matching task, and agrees with the postulation of Siegler (1991) that such factors are general to strategy selection and strategy evolution.

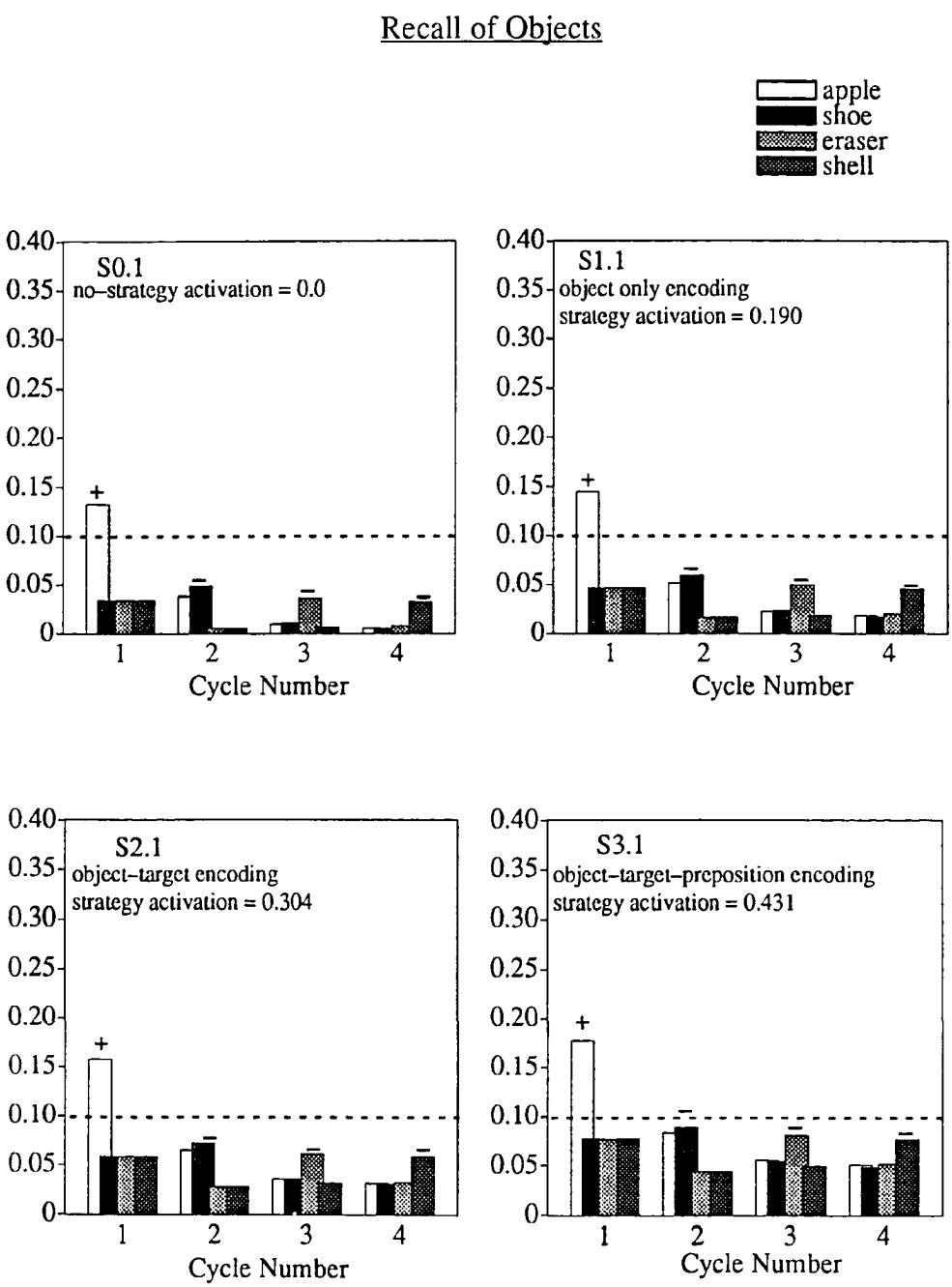


Figure 3.3 Effects of strategy use on recall of objects when strategies have minimum activations. X axis denotes cycle number and Y axis activation of an object unit. Plots S0.1, S1.1, S2.1 and S3.1: Recall of objects when "no–strategy," strategy one, strategy two, and strategy three were respectively used.

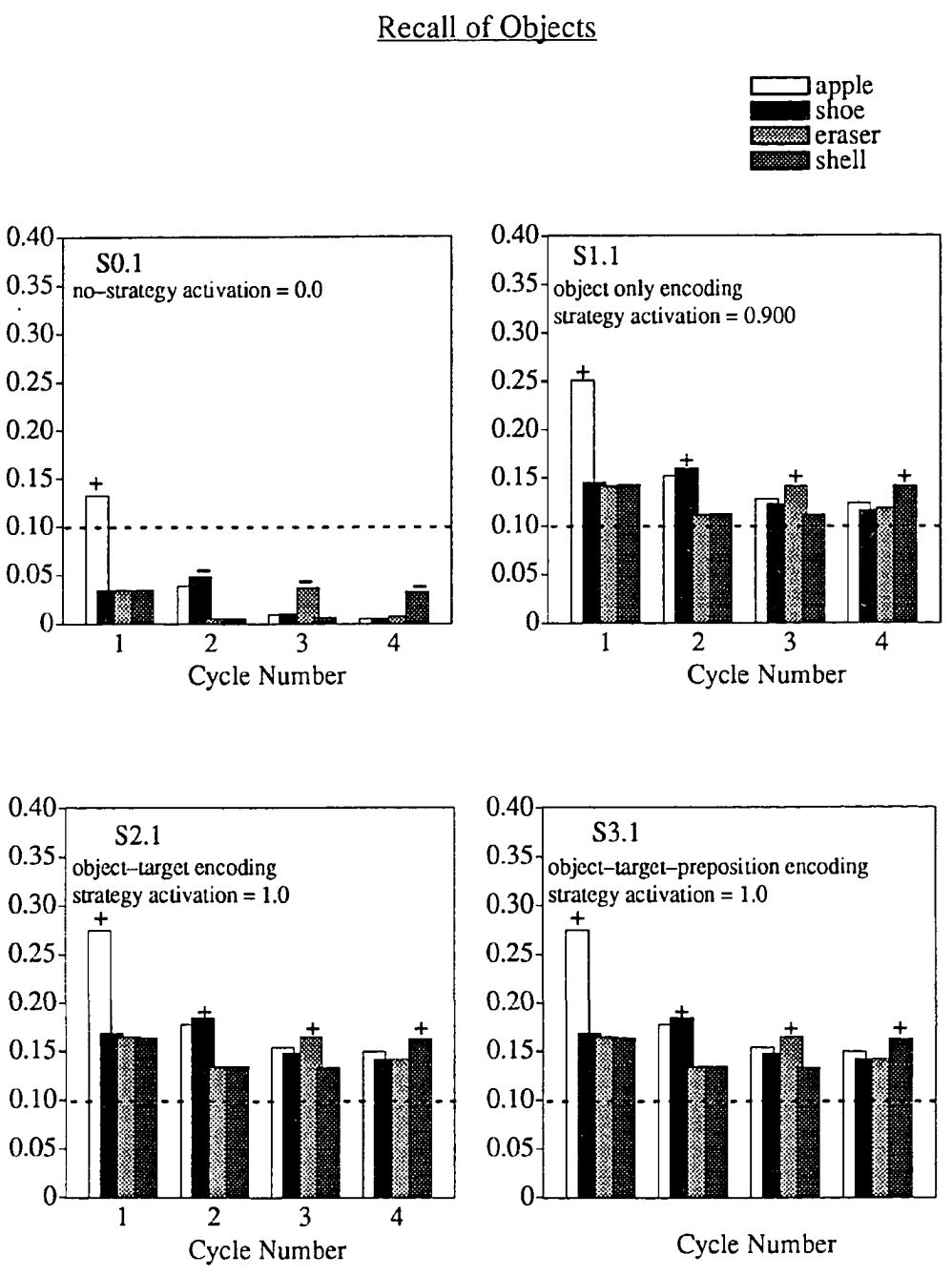


Figure 3.4 Effects of strategy use on recall of objects when strategies have maximum activations. X axis denotes cycle number and Y axis activation of an object unit. Plots S0.1, S1.1, S2.1 and S3.1: Recall of objects when "no–strategy," strategy one, strategy two, and strategy three were respectively used.

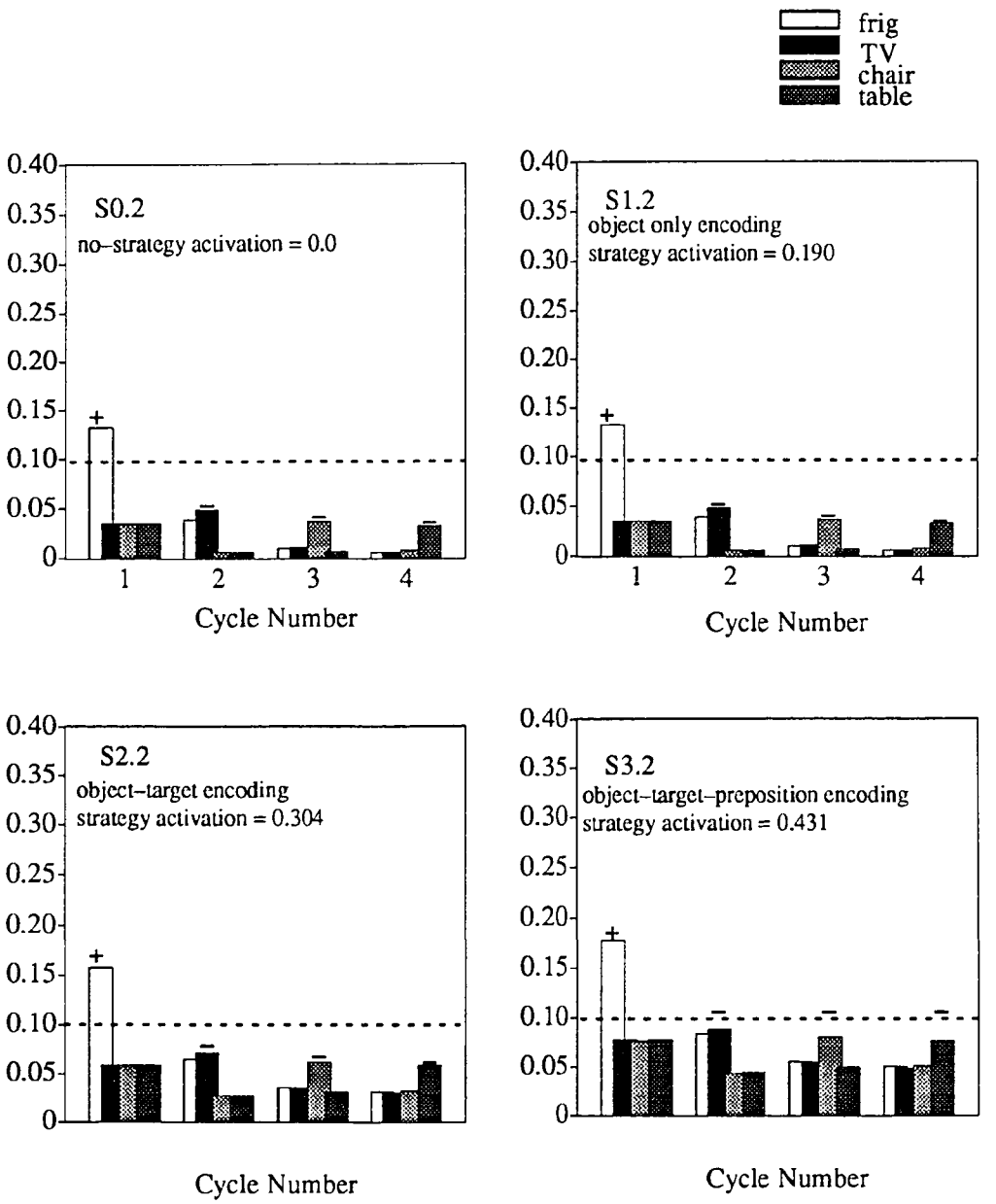


Figure 3.5. Effects of strategy use on recall of targets when strategies have minimum activations. X axis denotes cycle number and Y axis activation of an entity unit. Plots S0.2, S1.2, S2.2 and S3.2: Recall of targets with "no–strategy," strategy one, strategy two, and strategy three respectively.

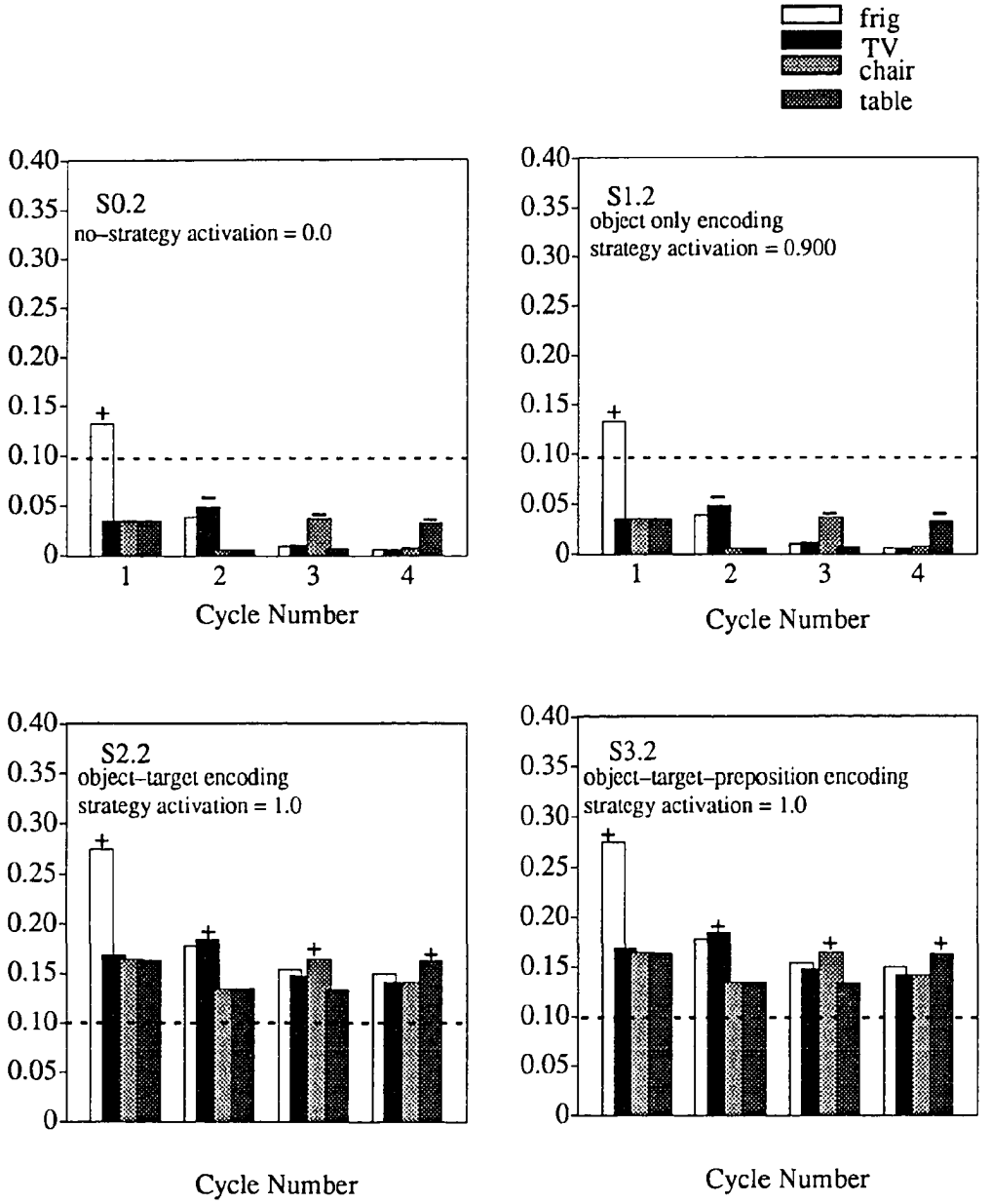


Figure 3.6. Effects of strategy use on recall of targets when strategies have maximum activations. X axis denotes cycle number and Y axis activation of an entity unit. Plots S0.2, S1.2, S2.2 and S3.2: Recall of targets with "no–strategy," strategy one, strategy two, and strategy three respectively.

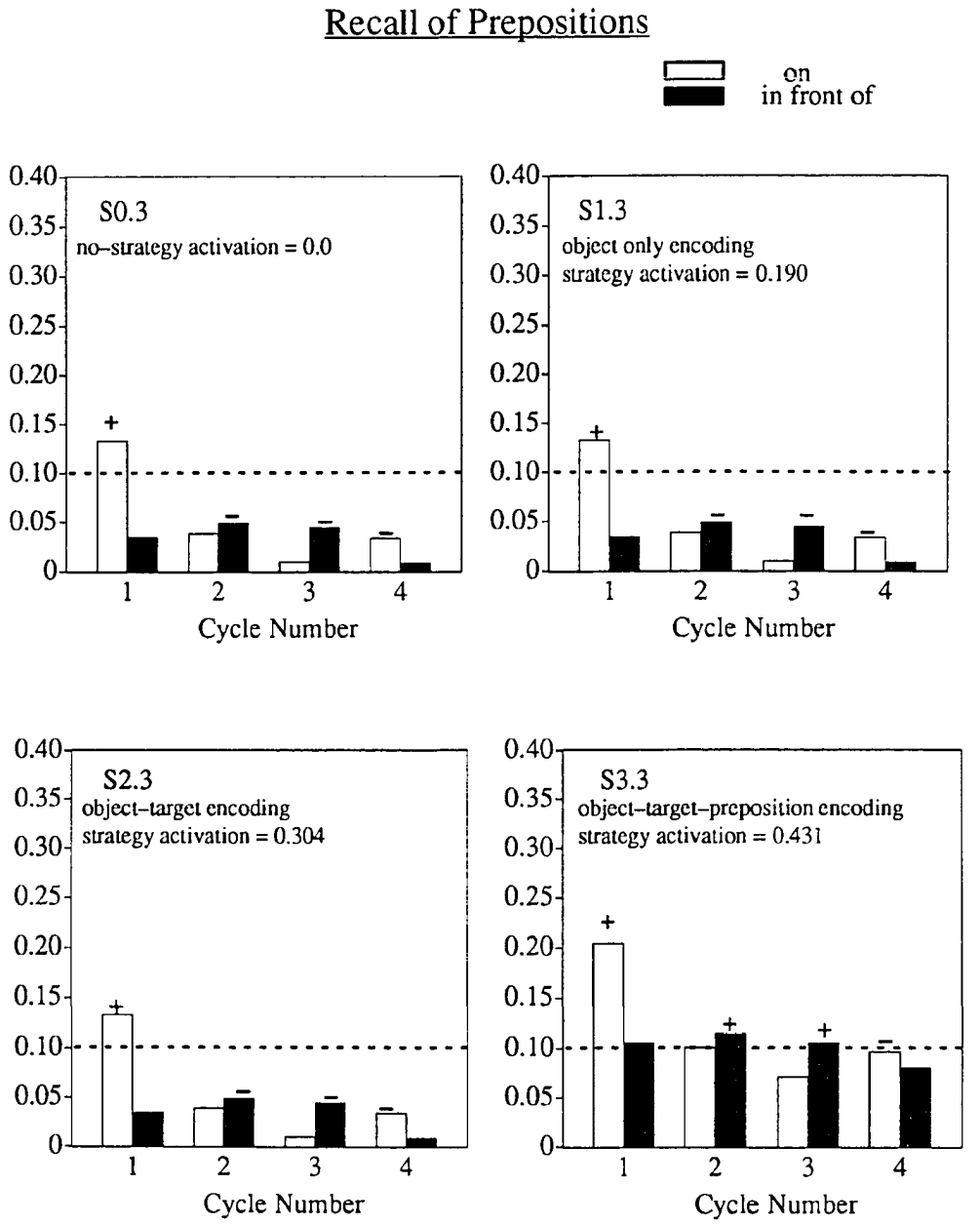


Figure 3.7. Effects of strategy use on recall of prepositions when strategies have minimum activations. The network has been tested for recall accuracy on the combination of (on, front, front, on). X axis denotes cycle number and Y axis activation of an entity unit. Plots S0.3, S1.3, S2.3 and S3.3: Recall of prepositions with "no-strategy," strategy one, strategy two, and strategy three, respectively.

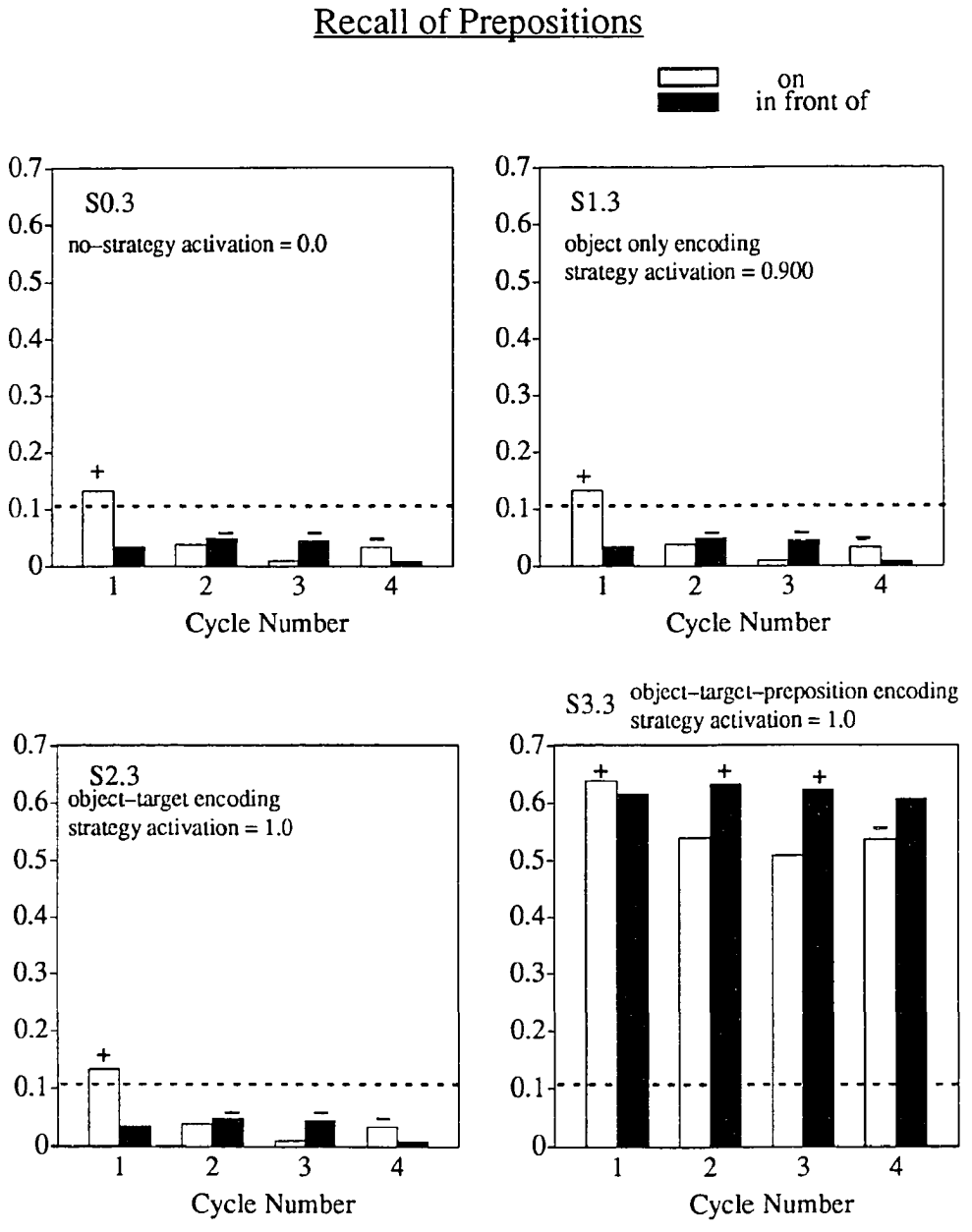


Figure 3.8. Effects of strategy use on recall of prepositions when strategies have maximum activations. The network has been tested for recall accuracy on the combination of (on, front, front, on). X axis denotes cycle number and Y axis activation of an entity unit. Plots S0.3, S1.3, S2.3 and S3.3: Recall of prepositions with "no–strategy," strategy one, strategy two, and strategy three, respectively.

CHAPTER 4

A COMPONENTS NEURAL NETWORK MODEL OF STRATEGY SELECTION AND EVOLUTION

In this chapter, we introduce the "components neural network model" for strategy selection and evolution. We construct this model to overcome the drawbacks of the novelty bias neural network model presented in Chapter three and to explore the notion that subjects are able to encode more sophisticated information with development. First, we discuss the motivation for the components neural network model in detail and follow up with discussions of its architecture, equations for activation and weight update, computer simulation of the model, and simulation results.

4.1 Motivation

The novelty bias neural network model as discussed in Chapter three, though useful as an initial prototype, is not completely satisfactory for the following reasons. First, the orderly introduction of novelty biases at an arbitrary number of trials for the selection of strategies seems less biologically plausible, even though this does emulate the formulation of strategy discovery in Siegler's model (Siegler & Jenkins, 1989). Second, the random variation in the contribution of novelty toward strategy selection may be very large compared to the contribution of accuracy. Our objective is to control the contributions of randomness, eliminating them if desirable and including them when necessary, to make the model more biologically plausible. Third, the novelty bias neural network model does not incorporate the notion of strategy components, which as we proceed, will be shown to be a useful formulation. Each of the strategy types considered in Chapter three may be broken down into finer elements, called "strategy components." For example, the object–target

encoding strategy is made up of at least three components: looking at the object, grasping the object, and moving the object toward the target.

Further, we want to explore the idea that children can encode increasingly sophisticated information with age, in order to explain the evolution of external memory strategies with development. This simple idea is evident in the work of Piaget (Siegler, 1991). For example, in a "balancing of weights" experiment, younger children normally take into account only one dimension, i.e., weight; whereas, older children normally take into account two dimensions, i.e., weight and distance of the weight from the fulcrum.

Sternberg and Rifkin (1979) present the differences in encoding between adults and children on analogy problems. According to their findings, adults spend a longer time on encoding and encode more features of a given problem than children. As a result, the solution times for adults are shorter than those for children.

Levine and Prueitt(1989), in their neural network modeling effort, adopt a similar idea to explain the differences in the performance of frontal lobe damaged patients and normal subjects. They attribute the differences in the performance of the two groups to differences in the strength of signals from sensory loci to reinforcement loci. In the context of the experiment of Bray et al. (1993), we postulate that children at earlier ages pay more attention to feedback about recall of objects than to feedback about recall of targets or prepositions. They gradually pay more attention to feedback about recall of targets and lastly, more attention to feedback about recall of prepositions. In other words, the ability to assimilate feedback about recall of objects, targets, and prepositions increases in that order.

As we shall see later, the components neural network model based upon this postulation helps explain the differences in strategy use not only among younger and older children but also among mentally retarded and nonretarded children. We may note here that feedback information about the recall of entities may originate internally from another part of the brain or externally from an experimenter.

4.2 Architecture of the Neural Network

The components neural network model for strategy selection and evolution is illustrated in Figure 4.1. Its architecture differs from that of the novelty bias neural network model presented in Chapter three as follows: the newer model lacks novelty bias units and has additional units that represent strategy components. The novelty bias units are eliminated in the new model because they contribute a large amount of randomness as discussed in the previous section. Further, according to Occam's razor, a model with minimal mechanisms is to be preferred if it can lead to equivalent behavior: novelty bias units are not needed if differences in encoding of information at the accuracy units can alone lead to the evolution of strategies. Specifications for instruction units, and entity units remain the same as provided in the section "architecture of neural network" in Chapter three. Specifications for strategy units, component units and accuracy units are presented below.

4.2.1 Component Units

Component units represent those parts that are postulated to make up strategies. In a bottom–up information processing perspective, which is commonly subscribed to in psychology, components are abstracted or automatized into strategies. In the current implementation of the model, three component units are used: object encoding unit, target encoding unit, and preposition encoding unit. The object encoding unit represents, for example, grasping an object in the hand. The target encoding unit represents, for example, the orientation of an object toward a target. The preposition encoding unit represents, for example, placing an object above or in front of the yellow board, corresponding to the preposition in the instruction.

It is quite conceivable to have component units in addition to these, e.g., pointing at an object, pointing at a target. The component units receive adjustable excitatory connections from accuracy units of a given type and send excitatory signals to appropriate strategy units through fixed excitatory connections, as illustrated in Figure 4.1.

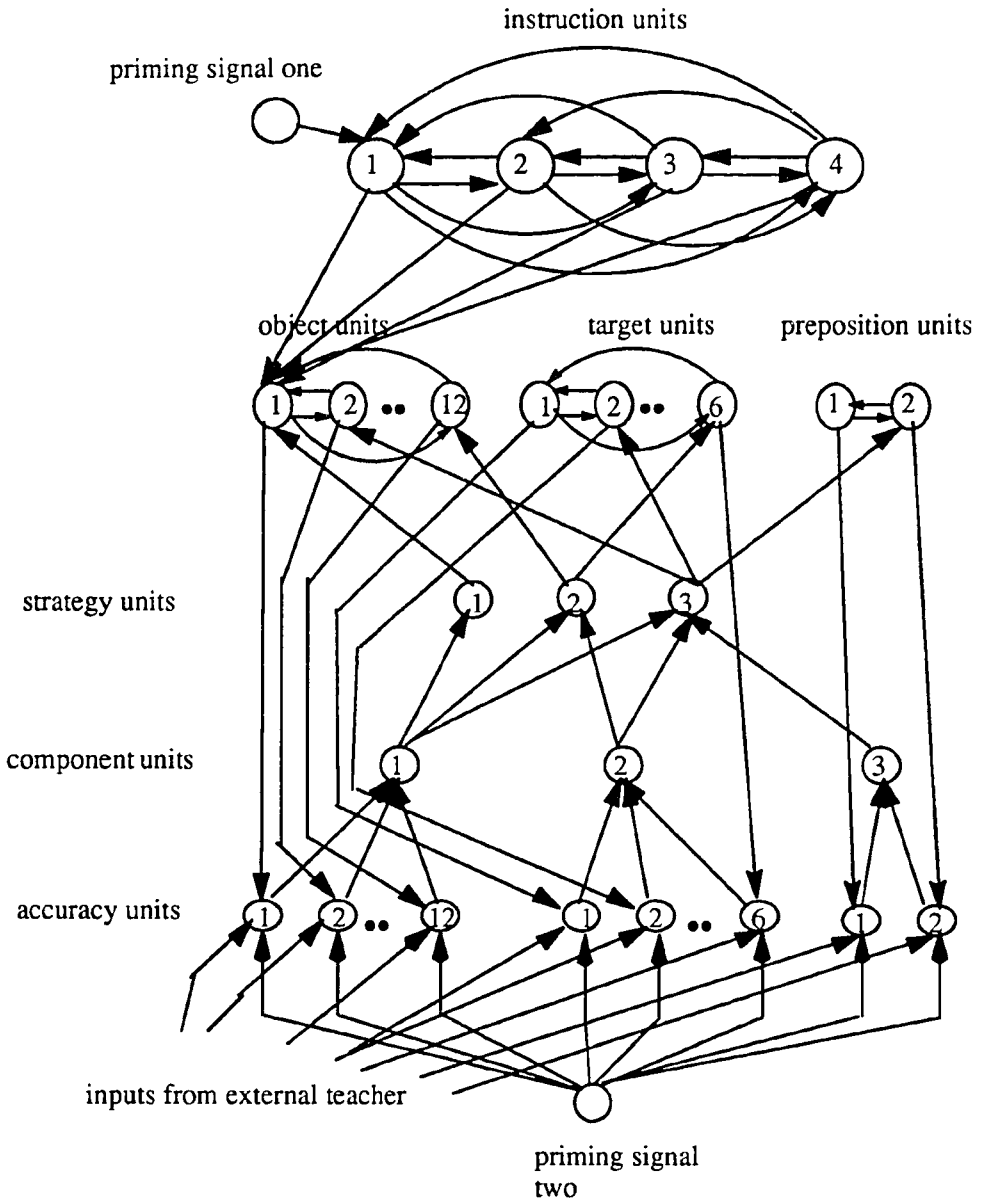


Figure 4.1. Architecture of the components neural network model for strategy selection and evolution.

4.2.2 Strategy Units

As in the novelty bias neural network model described in Chapter 3, the strategy unit pool consists of three units that represent three strategy types: object encoding only,

object–target encoding and object–target–preposition encoding. Each strategy unit receives excitatory connections from appropriate component units. In the current implementation, the strategy unit with the highest activation retains its value, and the activations of other units in the pool are set at zero. It would be possible to accomplish this using a "winner take all" mechanism such as lateral inhibition within the pool.

4.2.3 Accuracy Units

In the novelty bias model, each accuracy unit sends excitatory signals to every strategy unit; the underlying assumption is that all types of accuracy units potentially contribute to the evolution of strategies in the absence of components. However, with the introduction of component units in the current model, accuracy units send excitatory signals to relevant component units only: object accuracy units send excitatory input to the object encoding component only, target accuracy units to the target encoding component only, and preposition accuracy units to the preposition encoding component only.

4.2.4 Activation and Weight Update

The activation update equation for all the units in the components model except for the accuracy units and the strategy units takes the generalized form described by (3.1). The activation update for the accuracy units is given by equation (4.1). This equation differs from the accuracy equation (3.2) of the novelty bias model primarily through the new $D_k(t)$ term postulated to account for differences in encoding of information in the different types of accuracy units. The object accuracy units reach their maximum encoding ability most rapidly, the target accuracy units next, with the preposition units last. Characteristic $D_k(t)$ curves for the three accuracy unit types are displayed in Figure 4.2.

In addition to decay, excitatory input, and inhibitory input terms, the activation update equation for strategy units contains a term, Noise. Noise is assumed to be uniformly distributed over the interval [0.0, MaxNoise]. MaxNoise is small compared to the possible range of activation values in the network. In our assumptions, we are following researchers

such as Reeke, Sporns and Edelman (1990) and McClelland(1991) who apply randomness in neuronal processing to explain variation in behavior which arises even in the presence of constant external stimulus input. As we shall see, the noise term in the activation update equation of the strategy units allows for randomness in the selection of a strategy when two strategy units have nearly equal activation values. It would be possible to incorporate the same randomness term for all neuronal units in the neural network. The behavior of the network when a randomness term is used in all activation update equations needs further exploration.

Activation update equation for accuracy units:

$$a_i(t+1) = D_k(t) * a_k(t) * E_i(t) + I_i(t) \qquad (4.1)$$

where

$a_i(t)$ = *activation of accuracy unit i at time t*

$a_k(t)$ = *activation of corresponding entity unit k at time t*

$E_i(t)$ = *input from external teacher to accuracy unit i at time t*

$I_i(t)$ = *priming signal to accuracy unit i at time t*

$D_k(t)$ = *degree of encoding of information at accuracy unit i of type k at time t*

$$= C_1 * \left\{1./\left(1. + e^{-\lambda(t)/T_k}\right) - 0.5\right\}$$

$\lambda(t)$ = *degree of experience of subject in object − target matching task*

$$= \lambda(t-1) + C_2$$

$\lambda(0) = 0.0$

C_1, C_2 = *constants*

T_k = *temparature constant for accuracy unit of type k*

$k = 1, 2, 3$ *for object, target and prepostion respectively;*

In the current implementation, $T_1 < T_2 < T_3$

The weight update equation for adjustable connections is the same as the generalized Hebbian equation previously described by equation (3.3). We note here that the weights from accuracy units to component units are normalized to discount the effect of the number of units in each accuracy type. In other words, the total contribution of an accuracy type to its component unit is independent of the number of units within that accuracy type. Normalization is accomplished by dividing learning rate by the number of units within an accuracy type. The connections from component units to strategy units have fixed weights.

These weights are chosen under the assumption that components contribute in equal proportion to a given strategy.

4.3 Computer Simulation

Computer simulations of the components neural network model have been carried out to develop its performance characteristics. Each simulation run consists of several trials. Each trial consists of three phases: strategy selection, instruction presentation, and instruction recall. Simulation is halted when activations of strategy units reach stable values.

At the beginning of a trial, a priming signal to all accuracy units (priming signal two in Figure 4.1) results in the activation of accuracy units, component units, and strategy units, ultimately leading to the selection of a strategy. It is postulated that the activation of accuracy units, component units, and strategy units takes place in one cycle, without intermediate time delays. The selected strategy is in effect over a trial, i.e., throughout the instruction presentation and the instruction recall phases. We note here that this particular assumption of strategy use does not cover the cases wherein a subject uses multiple strategies in a single trial as seen, for example, in the experiment of Bray et al. (1993).

A sequence of four instructions is presented to the network with one instruction per cycle. In each cycle, the activations of all the instruction units and entity units in the network are adjusted, followed by the update of weights between instruction units and entity units, and of weights between strategy units and entity units.

At the end of the instruction presentation phase, the priming signal to the first instruction unit (priming signal one in Figure 4.1) triggers recall of the instruction sequence. When an instruction is being recalled, a "teacher," external to the model, provides reinforcement to accuracy units. For example, if a particular object is supposed to have been recalled in a given cycle and has been recalled (by exceeding the firing threshold), then the recall is deemed correct; otherwise, the recall is deemed incorrect. The teacher provides positive reinforcement to accuracy units corresponding to entities that have been recalled correctly.

It provides negative reinforcement to accuracy units corresponding to entities that have been recalled incorrectly. The teacher as discussed here may be external, such as the experimenter, or internal, such as signals received from another part of the brain.

The components model, unlike the novelty bias model presented in Chapter three, incorporates a $D_k(t)$ term that defines the degree of encoding of reinforcement information at a given accuracy type as illustrated in equation (4.1). The $\lambda_k(t)$ term which defines the experience of the subject with the object–target matching task is incremented during the strategy selection phase at the beginning of every trial. The T_k term, which stands for a Boltzman constant for each accuracy unit of type k, defines the differences in encoding of reinforcement among the various accuracy types. As may be noted from equation (4.1), $\lambda_k(t)$ and T_k jointly determine the effectiveness of the encoding of reinforcement for a given trial. Various parameters used in the computer simulation are listed in Table 4.1.

4.4 Results

The results of several computer simulation runs are presented in this section. The results of a typical run are given in Table 4.2. The table displays the strategy selected and its strength corresponding to a trial. In this particular simulation run, the object only encoding is discovered in trial forty–four. Discovery of a strategy occurs when its strength exceeds the firing threshold. We may note here that a strategy unit has an effect on an entity unit only when it exceeds the firing threshold. A firing threshold of 0.1 has been uniformly applied to all neurons in the network. Of course, due to the presence of the noise term in the activation update equation for strategy units, we observe a variation in the results from one simulation run to another.

The behavior of the model is analogous to that of human subjects in several ways. Children discover their first strategy only when they attain some chronological age (not at the embryonic stage!). In the model too, it takes a "long time," forty–three trials in this particular run, before any strategy is discovered. Furthermore, the selection of various

strategies before trial forty–four in the model illustrates that neuronal processing related to eventual strategy discovery may be taking place in the brain even though it does not become apparent through outward action.

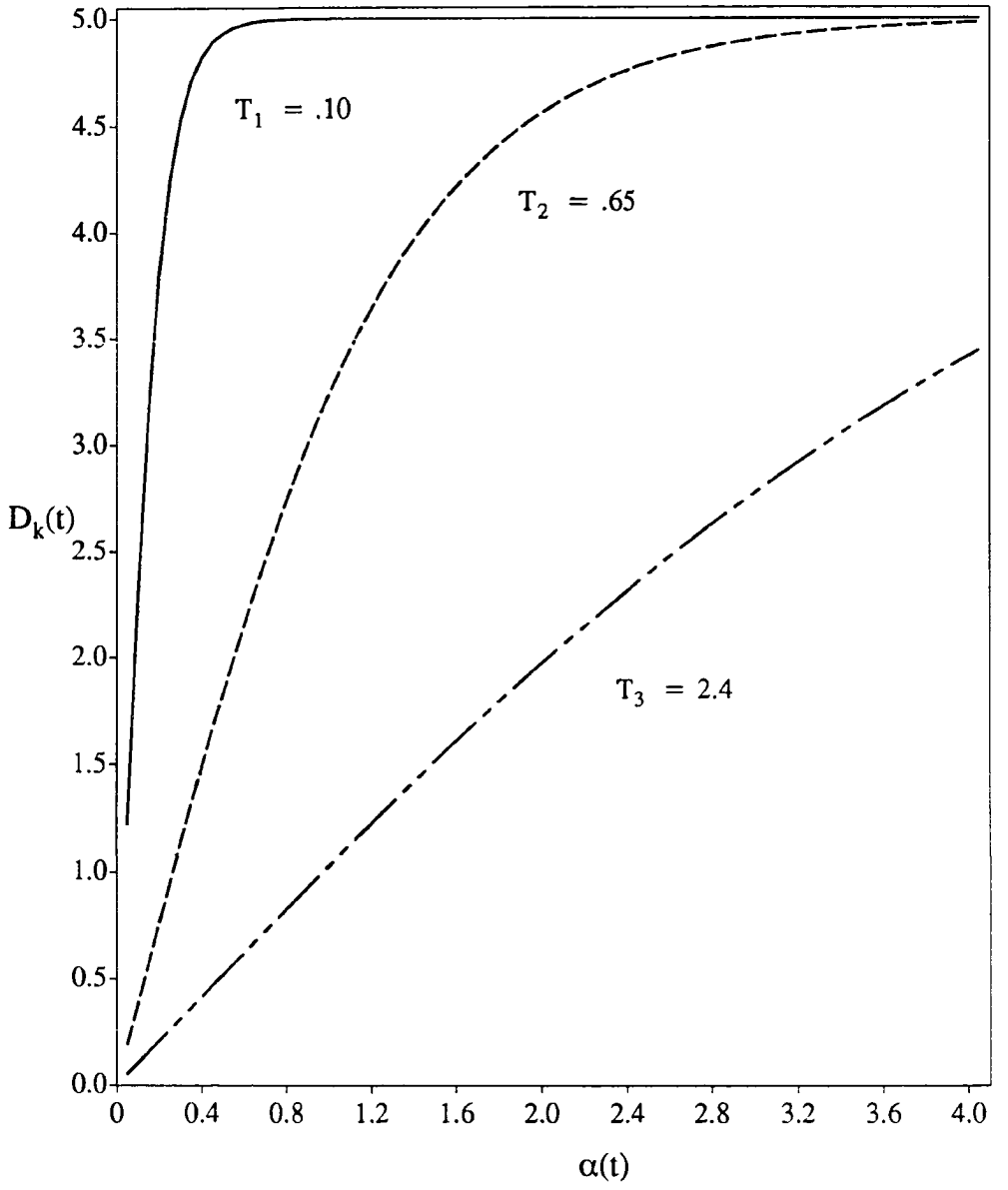


Figure 4.2. Characteristic curves for D(k)(t), ability to encode information at accuracy units for each type, k as a function of α (t) and T(k). See text for further details.

We may notice from Table 4.2 that after the object only encoding was discovered in trial forty–four, it is not used in trial forty–five and trial forty–seven because its strength falls below the firing threshold. The nonuse of a strategy after its discovery is reminiscent of what Siegler and Jenkins (1989) report, i.e., that children do not use a strategy because they are not certain about its execution. The study of Bray et al. (1993) also reveals that children do not use a particular strategy even though they are cognizant of it. Thus, the model effects a "nonuse of strategy" behavior after its discovery, similar to its being seen in children.

Table 4.1. Parameters used in computer simulation of the components neural network model.

$\alpha = 0.9,\ \beta = 1.0,\ \gamma = 0.4,\ q = -0.5,\ \Gamma_1 = 0.1;$

for instructions to entity connections :
$\Gamma_2 = 0.1,\ \Gamma_3 = 0.8,\ \delta = 0.4;$

for strategy to entity connections :
$\Gamma_2 = 0.1,\ \Gamma_3 = 0.8,\ \delta = 0.2;$

for accuracy to component connections :
$\Gamma_2 = 0.1, \Gamma_3 = 0.01;$
when accuracy unit $\geq \Gamma_2 : \delta = 0.8;$
when accuracy unit $< \Gamma_2 : \delta = -0.4;$

$C_1 = 10.0,\ C_2 = 0.025;$
$T_1 = 0.1,\ T_2 = 0.65\ ,\ T_3 = 2.4;$

Priming signal one = 1., priming signal two = 0.3;
Initial weights for all connections are set at 0.05;
Noise in activation update equation for strategy units is uniformly distributed in the interval 0.0 and 0.025.

At the beginning of the simulation, specifically until trial seventeen, all three strategies are being alternately selected at subthreshold levels due to the contribution of the noise term in the strategy update equation (refer to Table 4.2). Right after trial seventeen, the object only encoding strategy is frequently selected over the other two strategies. The object only encoding strategy exceeds threshold for the first time (is "discovered") on trial forty–four. This is caused by a higher degree of the encoding of feedback information for objects than

for targets or prepositions. The object–target encoding strategy is discovered in trial fifty–two and the object–target–preposition encoding strategy in trial fifty–nine. With the experience of more trials, the network encodes more feedback information for targets similar to objects as illustrated in Figure 4.2. This results in more frequent selection of the object–target encoding strategy between trials fifty–two and sixty–six. Lastly, the network encodes more feedback information for all three entities. This results in the eventual selection of the object–target–preposition encoding strategy only. Thus, during the "lifetime" of the network, its strategy choice progresses from the simplest, i.e., the object only encoding strategy, to the most advanced, i.e., the object–target–preposition encoding strategy.

This run and other runs presented in Appendix B illustrate the use of diverse strategies within short intervals. As Siegler and Jenkins (1989, p.27) note, "at any one time, individual children use diverse strategies to solve arithmetic problems." Bray et al (1993) also observe the use of diverse strategies in the object–target matching task. Examination of Table 4.2 reveals that all three strategies have been selected within the close interval of trial fifty–nine to trial sixty–seven. This clearly demonstrates the diverse nature of strategy selection in the model.

4.4.1 Recall Accuracy and Strategy Use

In this section, we deal with the effects of strategy use on recall accuracy. When a strategy is used, the entity units that are connected to the strategy unit receive a boost in their activations. The amount of boost depends on the activation of the strategy units. The recall accuracy of the network using a particular strategy, therefore, depends on the activation of the strategy. Among the entities, objects and targets are uniquely used in the instruction sequences without repetition. Thus, the recall accuracy of the network is independent of the order of the objects and targets. However, the two prepositions, "on" and "in front of," are each repeated twice in an instruction sequence and the recall accuracy of the network

depends on the particular combination of the prepositions used in training the network. Thus, we tested the components neural network using different values of strategy activations and all possible combinations of prepositions.

The object only encoding strategy has a minimum activation of 0.100 in simulation run four and a maximum activation of 0.203 in simulation run five (refer to Appendix B); The object–target encoding strategy of 0.102 each in runs one and two, and 0.392 in run eight, respectively; and the object–target–preposition encoding strategy of 0.139 in run two and of 1.0 in all runs.

We illustrate the results on recall accuracy using the following instruction sequence:

(1) "Put the apple on the refrigerator."

(2) "Put the shoe in front of the TV."

(3) "Put the eraser in front of the chair."

(4) "Put the shell on the table."

Recall accuracy of an entity using a strategy, as noted in Chapter three, is defined as the ratio of number of entities correctly recalled to the number of instructions (kept at four throughout this investigation). An entity is said to be correctly recalled if it exceeds the firing threshold, has the highest activation in its pool, and fires in the order of its training during the instruction presentation phase.

The performance of the network with regard to the recall accuracy of objects is illustrated in Figure 4.3 when strategies have minimum activations. When no strategy is used, the network recalls apple, eraser, and shell correctly during cycles one, three, and four, respectively (Chart S0 in Figure 4.3). The "shoe" object has the highest activation in its pool in cycle two; however, it does not exceed the firing threshold. Thus, the recall accuracy of objects for "no–strategy use" case is 75%. As charts S1, S2, and S3 illustrate, use of strategies one, two, and three has the effect of boosting the activation of previously trained objects. However, the boost is not great enough to enhance the recall accuracy. Thus, the

recall accuracy of objects, when the strategies are used with minimum strengths, remains at 75%.

Table 4.2. Results of a typical simulation run that indicate strategy evolution in the components neural network model.

Trial#	SS	Strength	Comments	Trial#	SS	Strength	Comments
1	2	0.030		40	1	0.082	
2	2	0.038		41	1	0.089	
3	3	0.038		42	1	0.083	
4	1	0.040		43	2	0.084	
5	1	0.039		44	1	0.102	discovered
6	2	0.039		45	1	0.099	not used
7	3	0.031		46	1	0.109	
8	3	0.033		47	1	0.098	not used
9	3	0.040		48	1	0.114	
10	1	0.038		49	1	0.117	
11	1	0.030		50	1	0.121	
12	2	0.029		51	1	0.118	
13	1	0.041		52	2	0.135	discovered
14	2	0.043		53	1	0.148	
15	2	0.034		54	2	0.149	
16	2	0.041		55	1	0.140	
17	3	0.041		56	1	0.166	
18	1	0.044		57	1	0.162	
19	1	0.048		58	1	0.168	
20	1	0.048		59	3	0.180	discovered
21	1	0.046		60	1	0.191	
22	1	0.053		61	3	0.204	
23	1	0.054		62	2	0.227	
24	2	0.050		63	3	0.241	
25	3	0.049		64	3	0.252	
26	1	0.048		65	2	0.286	
27	1	0.057		66	2	0.310	
28	2	0.056		67	3	0.331	
29	1	0.060		68	3	0.360	
30	1	0.052		69	3	0.412	
31	1	0.071		70	3	0.419	
32	1	0.055		71	3	0.457	
33	1	0.071		72	3	0.513	
34	1	0.056		73	3	0.569	
35	1	0.069		74	3	0.609	
36	2	0.065		75	3	0.693	
37	1	0.085		76	3	0.775	
38	1	0.075		77	3	0.882	
39	1	0.082		78	3	1.000	

Notation: SS: strategy selected: Strength refers to the activation of strategy unit selected.

Recall accuracy of the network, when strategies have maximum activations is portrayed in Figure 4.4. As charts S1, S2, and S3 illustrate, all object units receive a boost with strategy use. Activations of advanced strategies have higher maxima and the object units therefore receive greater amounts of boost from these strategies. The recall accuracy of the network

on objects when strategies are used with maximum strengths is 100%. Overall, it varies from 75% to 100% depending on the strength of the strategy.

Figure 4.5 depicts the recall accuracy of the network on targets when strategies have minimum activations. Without the use of any strategy, the recall accuracy is 75% (see Chart S0 in Figure 4.5). The object only encoding strategy has no effect on the recall of targets. Thus, it does not bring about any changes in the activations of targets as compared to the "no–strategy use" case (contrast Chart S1 with Chart S0 in Figure 4.5). Use of the object–target encoding strategy or the object–target–preposition strategy results in a boost in the activations of previously trained targets. However, this boost is not big enough to enhance the recall accuracy in either case. Thus, the recall accuracy of targets when the strategies are used with minimum strengths remains at 75%.

Figure 4.6 depicts the recall accuracy of targets when the activations of strategy units are at their maximum values. Charts S2 and S3 indicate that with the use of the object–target encoding strategy and the object–target–preposition encoding strategy, the activations of targets receive a boost that raises them above the firing threshold. Thus, the recall accuracy using these two strategies is 100%. Overall, the recall accuracy of the network on targets varies from 75% to 100%.

The performance of the network on the recall of prepositions is tested using the minimum and maximum activations of various strategies (Figures 4.7 and 4.8). Neither the object only encoding strategy nor the object–target encoding strategy encodes prepositions. Use of the object only encoding strategy (see Chart S1 in Figures 4.7 and 4.8) and use of the object–target encoding strategy (see Chart S2 in Figures 4.7 and 4.8), thus, do not bring about any changes in the activations of prepositions as compared to the "no–strategy use" case (see Chart S0 in Figures 4.7 and 4.8). The recall accuracy for any combination of prepositions without any strategy use and with use of the object only encoding strategy or the object–target encoding strategy is 75%.

Use of the object–target–preposition encoding strategy results in a boost in the activations of prepositions in appropriate cycles. When the network is trained on the particular combination of (on, in front of, in front of, on) as part of the instruction sequence and the object–target–preposition encoding strategy has a minimum activation, the "in front of" unit receives a boost in cycle two but does not exceed the firing threshold (Chart S3 in Figure 4.7). We tested the network on all possible combinations of prepositions with the object–target–preposition encoding strategy at its minimum activation value and found that the recall accuracy varies from 75% to 100%.

Table 4.3 Summary of performance of the components neural network on recall accuracy.

ENTITY	STRATEGY USED			
	NONE	1	2	3
OBJ	75%	75–100%	75–100%	75–100%
TARG	75%	75%	75–100%	75–100%
PREP	75%	75%	75%	50–100%

When the network is trained on the particular combination of (on, in front of, in front of, on) and the object–target–preposition encoding strategy has a maximum activation, the activations of both preposition units receive significant boost in all four cycles of recall but the activation of "on" dominates that of "in front of" in cycles two and three, leading to incorrect recall in these latter cycles (Chart S3 in Figure 4.8). The recall accuracy, in this case, is 50%. Testing of the network on all other combinations of prepositions revealed that recall of one preposition dominates the recall of the other, and that the recall accuracy in all these cases is 50% each.

As noted above, the performance of the network deteriorates when the activation of the object–target–preposition encoding strategy increases from minimum to maximum. For this reason, we have tested the network on all possible combinations of prepositions when the activation of the object–target–preposition encoding strategy takes an intermediate value

(0.180). The recall accuracy of the network on all these combinations is 100%. In summary, the recall accuracy of the network on prepositions varies from 50% to 100%, based on the activation value of the strategy and on the particular combination of the prepositions. The performance of the network on recall of entities using the various strategies is summarized in Table 4.3.

Table 4.4 Strategy use in nonmentally–retarded children (11–year old).

RUN#	STRATEGY USE IN 16 TRIALS		
	type 1	type 2	type 3
1	0	3	13
2	0	4	12
3	0	3	13
4	0	7	9
5	1	8	7
6	1	7	8
7	0	5	11
8	1	9	6
9	1	3	12
10	0	4	12
average use	0.4	5.3	10.3
% use	2.5	33.12	64.38
EMPIRICAL DATA:			
% use	2.84	33.5	63.6

NOTE: The values of strategy use for neural network are computed starting at trial 60.

4.4.2 Comparison of Network Performance with Empirical Data

One of the motivations behind this research effort has been to account for the differences in the performance of educable mentally–retarded children (EMR) and nonmentally–retarded (NMR) children. A key difference between the two intelligence groups is the frequency of strategy use. According to the study of Bray et al. (1993), EMR children use the object only encoding strategy and the object–target encoding strategy more frequently than NMR children. On the other hand, the latter group uses the object–target–preposition strategy more frequently. Tables 4.4 and 4.5 list the empirical data for strategy use in NMR and EMR children, respectively, in addition to data for strategy use

Cases of the nonuse of any strategy and of pointing as a strategy are not included in the empirical data in order to obtain the best possible correlation between observed data and the performance of the neural network.

The use of various strategies in the network is compared to the same in the two intelligence groups by averaging the results from multiple simulation runs. Due to randomness in the processing of strategy units, strategy choice within a trial in one simulation run often differs from the strategy choice within the same trial in another simulation run. However, each simulation run is consistent with another in the manner of progression of strategies, i.e., from simple to most advanced.

The results of strategy selection in the network for 10 simulation runs starting at trial sixty and trial fifty–seven are shown in Table 4.4 and Table 4.5, respectively. Each row in both tables lists the number of times every strategy type is selected starting at trial sixty or fifty–seven for 16 successive trials. We choose 16 successive trials because it corresponds to the number of trials that each child receives in the study of Bray et al. (1993). We settle on starting trials of sixty and fifty–seven because the resultant percentages of strategy use correlate best with the empirical data for NMR children and EMR children at these starting values. We observe from Table 4.4 and Table 4.5 that the average percentages of strategy selection in network match quite well with those of the NMR children and EMR children in all strategy categories.

4.5 Discussion and Conclusions

In this chapter, we have proposed the construction of the components neural network model for strategy selection, presented its architecture and the outcomes of computer simulations, and compared the performance of the model with empirical data from the study of Bray et al. (1993).

The formulation of the components neural network model is motivated by classical Piagetian notions that children take into account more dimensions of information with

development and their cognitive maturation involves advancement from simple sensory–motor manipulation during infancy to abstract thinking during adolescence and adulthood. In the context of the "object–target matching task," these notions imply that children pay attention to feedback information about the recall of concrete entities (objects and targets) earlier, and to feedback information about the recall of abstract entities (prepositions) later. The adoption of the "differences in feedback" postulation has eliminated the necessity for novelty bias as a controlling factor in the selection and evolution of strategies. As we have seen in Chapter three, the novelty bias neural network model assumes that a strategy has already been discovered at the instant of the introduction of the novelty bias for that strategy. Thus, the previous model does not explain how strategies are discovered. In the components neural network model, a strategy is gradually formed by continual inputs from accuracy units and actually discovered when its activation exceeds the firing threshold.

Table 4.5 Strategy use in educable mentally–retarded children (11–year old).

RUN#	STRATEGY USE IN 16 TRIALS		
	type 1	type 2	type 3
1	1	5	10
2	1	4	11
3	2	4	10
4	1	9	6
5	1	11	4
6	2	9	5
7	0	8	8
8	2	11	3
9	3	3	10
10	1	5	10
average use	1.4	6.9	7.7
% use	8.75	43.12	48.12
EMPIRICAL DATA:			
% use	9.1	41.5	49.3

NOTE: The values of strategy use for neural network are computed starting at trial 57.

An advantage of neural network modeling is evident here for the purposes of studying cognitive development. The process of neural network construction forced us to abandon the notion of novelty bias, which is abruptly introduced into the network and probably biologically implausible, and to explore the idea of "differences in feedback." The latter idea seems biologically plausible in light of the modeling effort of Levine and Prueitt (1989). They attributed the differences in the manifest behavior of normal subjects and frontal lobe damaged patients to differences in the strength of signal from sensory loci to reinforcement loci. The components neural network model, as we have seen, attributes the differences in strategy selection between NMR children and EMR children to the degree of the encoding of feedback information from the entities.

Although the initial fit to empirical data shown in Tables 4.4 and 4.5 is encouraging, some ambiguity remains. It is not clear why a difference of only three trials in the beginning of the 16–trial block simulates the differences between the two groups of children. It may be that the simulated trials of the model correspond to larger psychological units of experience. The NMR children, having more experience with encoding strategy components, possess a higher proportion of more sophisticated strategies than the EMR children. This aspect of the fit of the simulation to the empirical data requires further investigation.

The components neural network model exhibits qualitative behavior observed by Siegler and Jenkins (1989) in arithmetic tasks, and by Bray et al. (1993) in the external memory task in several aspects: use of diverse strategies in short intervals, discovery of strategies, nonuse of a strategy after its discovery, evolution of strategies from simple to advanced in individuals, and use of simpler strategies by younger children and of more advanced strategies by older children.

Outcomes from computer simulations point out that the behavior of the components neural network model is not as variable as that of the children studied by Bray et al. (1993).

Children in the study use a strategy as infrequently as about 50% of the trials. The model, except for the few trials immediately following the discovery of the first strategy, always uses a strategy. Children typically use a variety of strategies in any given trial. On the other hand, the model uses the same selected strategy throughout the duration of a trial. The accuracy of recall in children varies from 0% to 100% whether a strategy is used or not. The accuracy of recall in the network is always 75% when a strategy is not used and can vary from 50% to 100% when a strategy is used. Thus, the model exhibits limited variability and further enhancements of the model should be attempted to make it more realistic.

As with the novelty bias neural network, the components neural network recalls prepositions with lower accuracy at high activations of the "object–target–preposition encoding" strategy. Analysis of the simulation results reveals that high activations of the object–target–preposition encoding strategy lead to persistence of activity in preposition units which establishes spurious associations between preposition units and instruction units. The spurious associations cause incorrect recall of prepositions. This anomalous result needs to be addressed in future work.

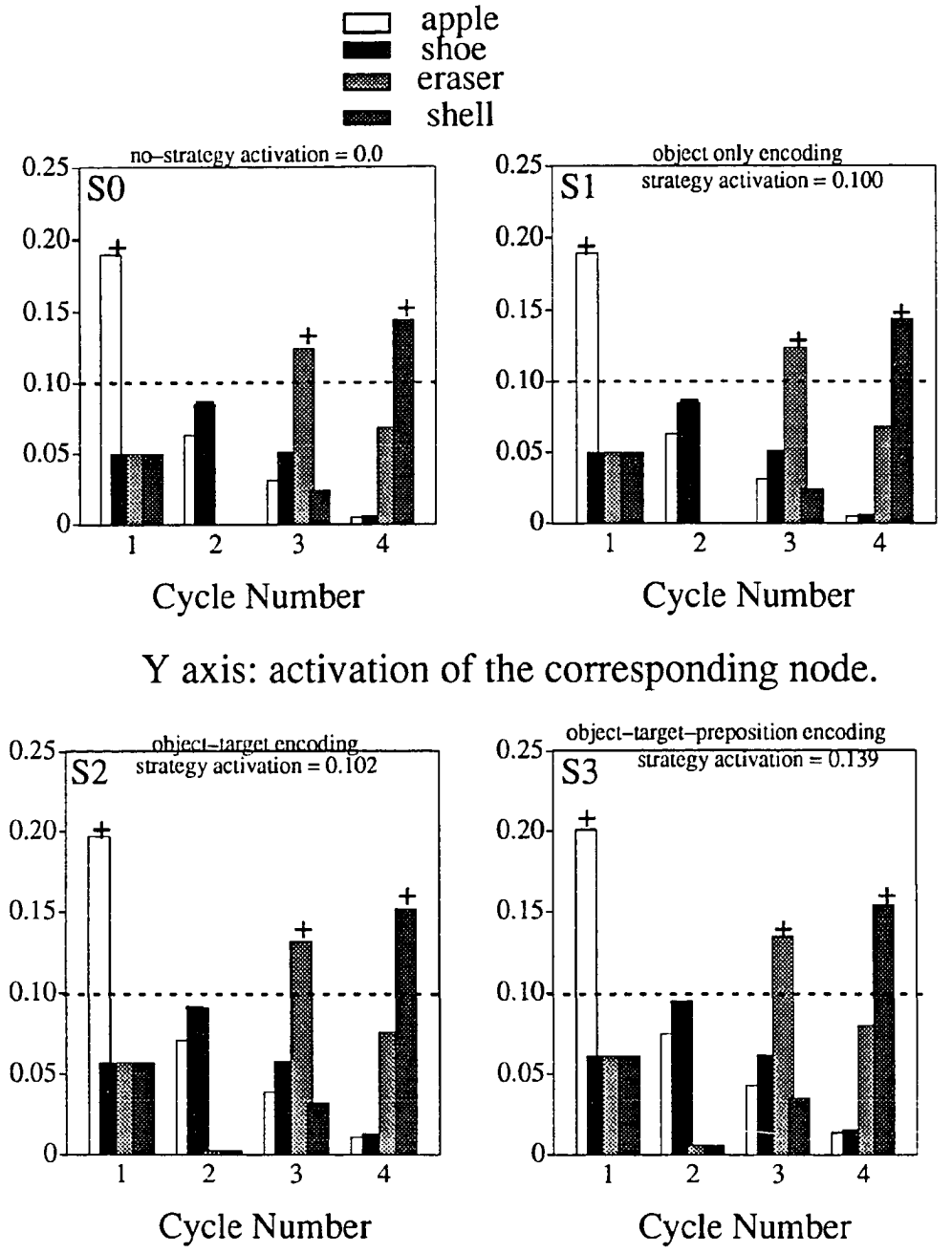


Figure 4.3 Effects of strategy use on recall of objects when strategy units have minimum activations. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object-target encoding" and "object-target-preposition encoding" cases. X axis represents cycle number and Y-axis activation of an object unit. The dashed line depicts the common firing threshold for all neurons in the network.

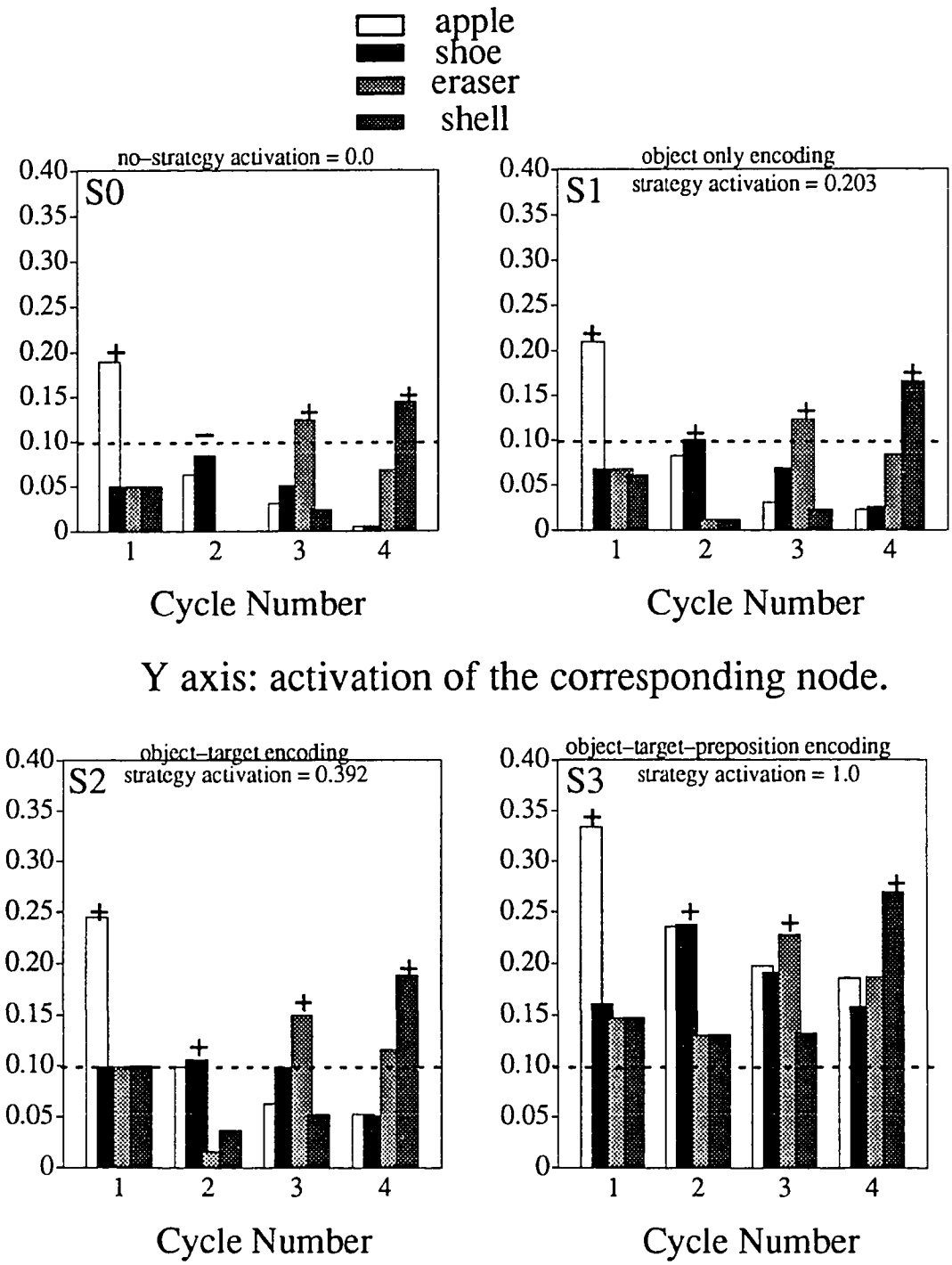


Figure 4.4. Effects of strategy use on recall of objects when strategy units have maximum activations. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object-target encoding" and "object-target-preposition encoding" cases. X axis represents cycle number and Y-axis activation of an object unit. The dashed line depicts the common firing threshold for all neurons in the network.

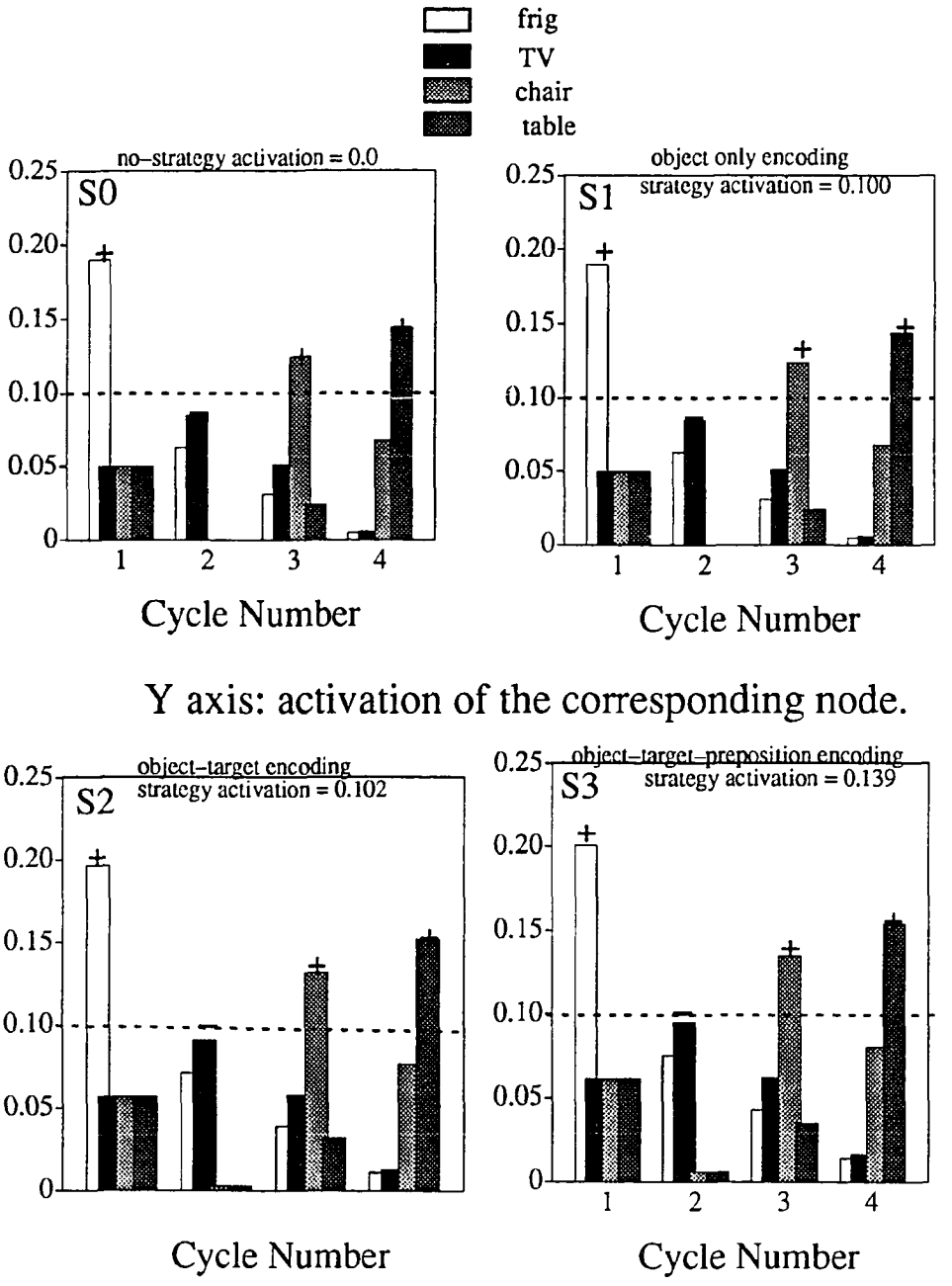


Figure 4.5. Effects of strategy use on recall of targets when strategy units have minimum activations. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object-target encoding" and "object-target-preposition encoding" cases. X axis represents cycle number and Y-axis activation of a target unit. The dashed line depicts the common firing threshold for all neurons in the network.

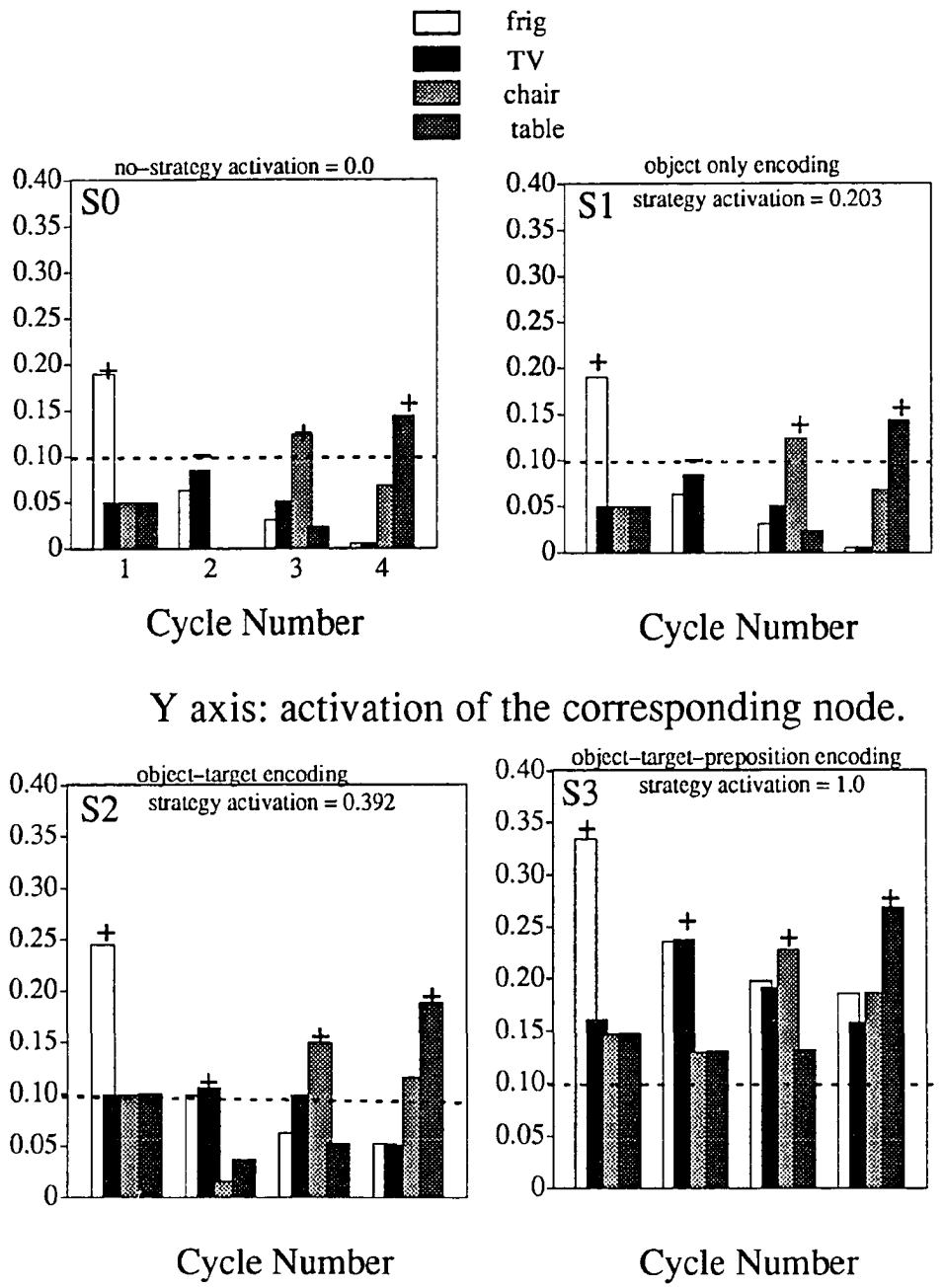


Figure 4.6. Effects of strategy use on recall of targets when strategy units have maximum activations. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object–target encoding" and "object–target–preposition encoding" cases. X axis represents cycle number and Y–axis activation of a target unit. The dashed line depicts the common firing threshold for all neurons in the network.

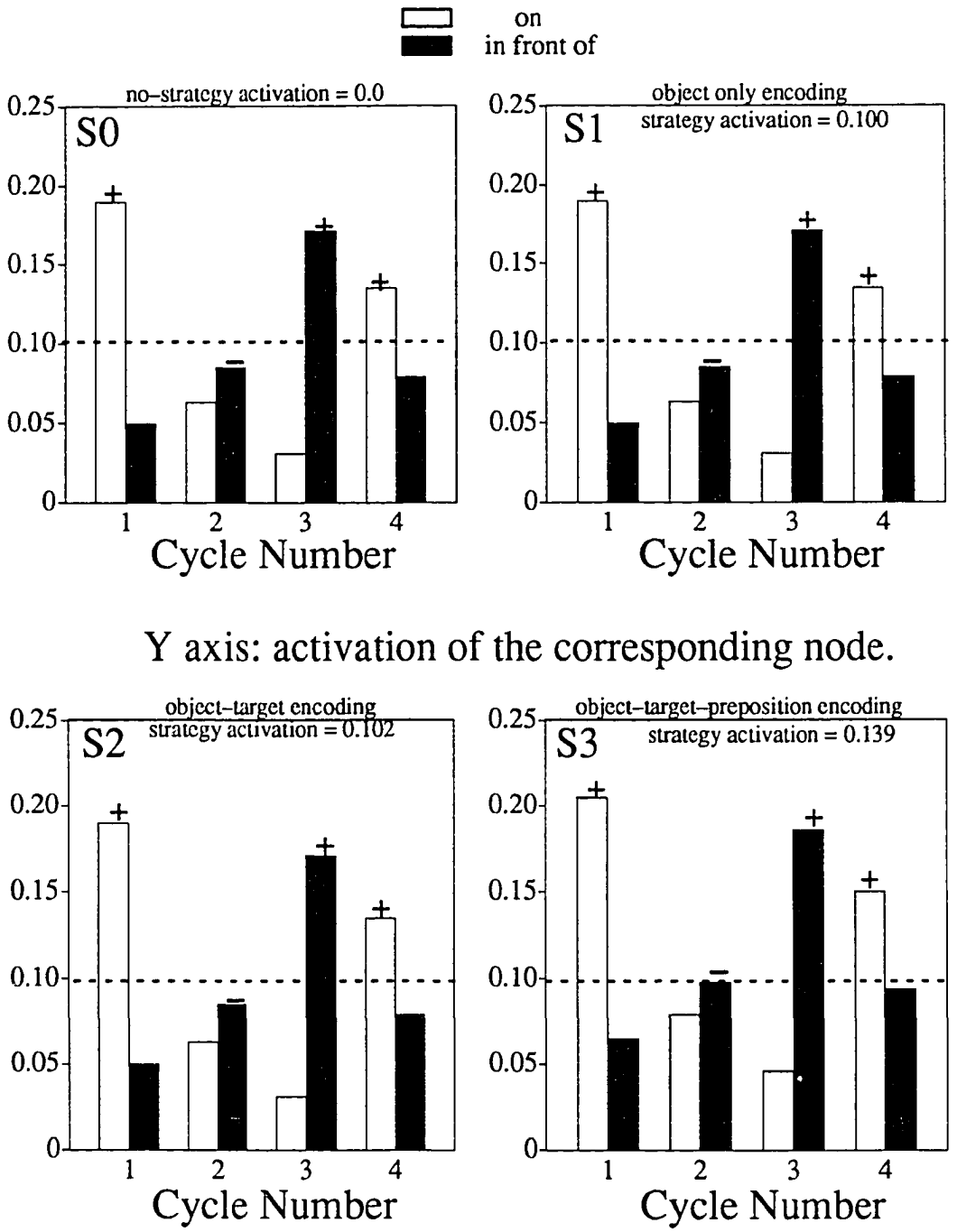


Figure 4.7. Effects of strategy use on recall of prepositions when strategy units have minimum activations. The combination of (on, in front of, on, in front of) is used in in this particular testing. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object–target encoding" and "object–target–preposition encoding" cases. X–axis represents cycle number and Y–axis activation of a preposition unit. The dashed line depicts the common firing threshold for all neurons in the network.

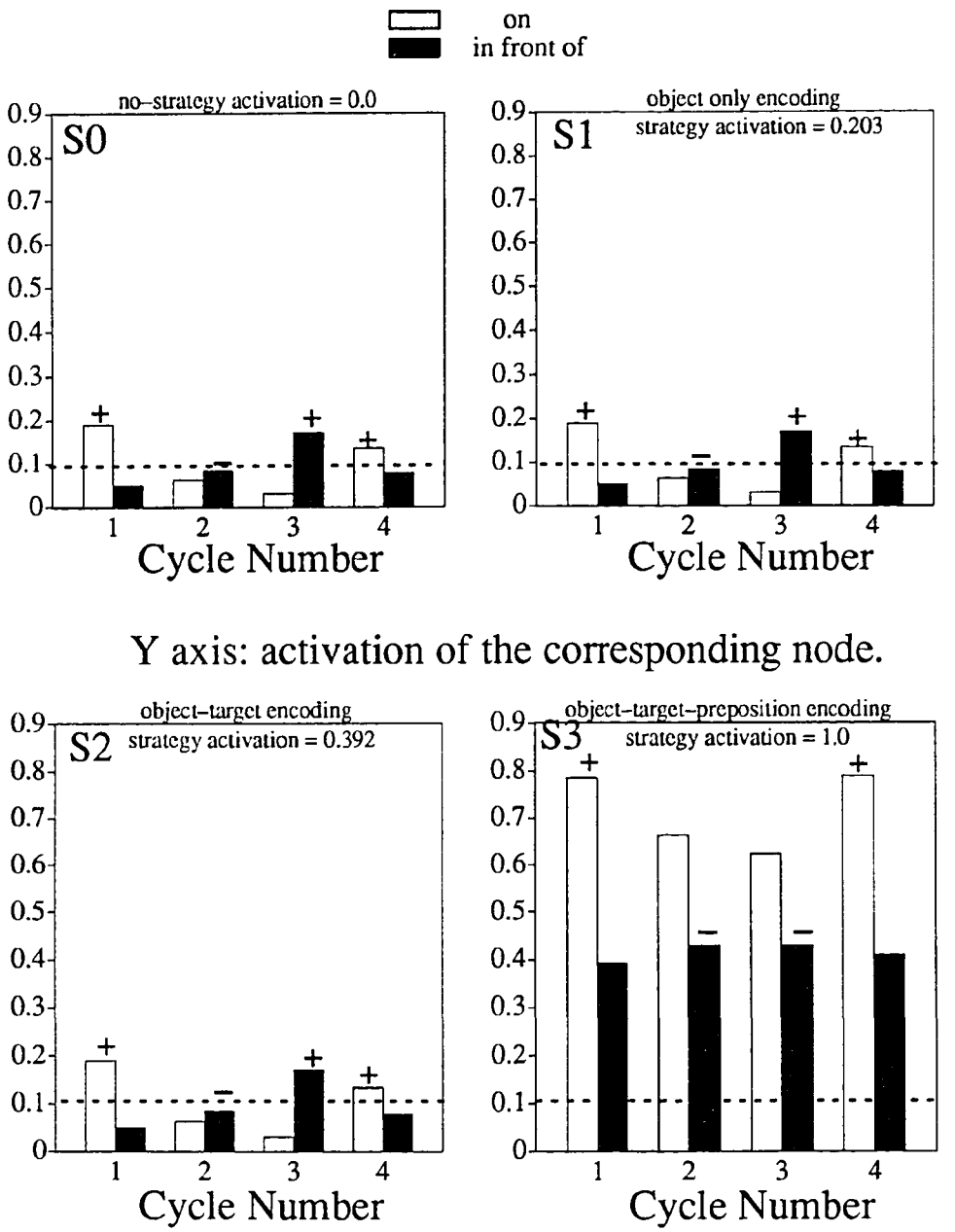


Figure 4.8. Effects of strategy use on recall of prepositions when strategy units have maximum activations. The combination of (on, in front of, on, in front of) is used in in this particular testing. S0, S1, S2 and S3 refer to "no strategy use," "object encoding only", "object–target encoding" and "object–target–preposition encoding" cases. X–axis represents cycle number and Y–axis activation of a preposition unit. The dashed line depicts the common firing threshold for all neurons in the network.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, we evaluate what has been accomplished in the previous chapters. We follow the example of Siegler and Shipley (1993) who evaluate three models of strategy evolution based on five empirical phenomena that are observed in many cognitive problem domains. The neural network models of strategy development presented in Chapters three and four are assessed on how they exhibit these phenomena. Next, we offer some alternatives with regard to the construction of neural network models which we faced during their design but were bypassed to achieve the main objectives of the dissertation. These alternatives have the potential of making the models more elegant and of suggesting other behavioral patterns. Last, we identify other cognitive tasks, and business and robotic applications that are relevant to the current investigation.

5.1 Assessment of Neural Network Models Based on the Phenomena of Strategy Development

Based on evidence from several problem domains such as arithmetic, serial learning, formation of past tense, and time telling, Siegler and Shipley(1993) observe five phenomena that accompany strategy selection and evolution. They suggest that models of strategy development exhibit these phenomena in order to be considered realistic. Here, we present a brief description of these phenomena and assess the novelty bias neural network model and the component neural network model based on them.

1. Variability: There does not exist a one–to–one correspondence between the age of a subject and his strategy choice. For example, a kindergartner, on the average, uses the retrieval strategy on 16% of the addition problems in a session and the "minimum" strategy

on 30% of the problems. A second grader, on the average, uses the retrieval strategy on 45% and "minimum" strategy on 40% of the problems in the same session (Siegler & Shipley, 1993). Thus, the second grader uses both strategies similar to the kindergartner but with different relative frequencies.

Both neural network models manifest the characteristic of variability in their of selection of strategies. Between trials twenty–five and thirty–four in Table 3.1, the novelty bias neural network selects strategy one twice, strategy two thrice and strategy three five times. Between trial fifty–five and sixty–four in Table 4.2, the components neural network selects strategy one five times, strategy two once and strategy three four times.

2. Adaptive Strategy Choices: In general, humans show adaptivity in strategy choice. The more difficult a given problem is, the more often they rely upon well–understood but slower strategies. In contrast, the easier the problem is, the faster the use of strategies. For example, second grade children fall back on the sum strategy on more difficult addition problems and use the retrieval strategy on easier addition problems.

Currently, the neural network models are trained on problems of equal difficulty. Each problem in the context of object–target matching task consists of a sequence of four instructions. The idea of difficultly may be incorporated into this task, for example, by varying the number of instructions from one trial to another in the computer simulations of the models.

3. Change: Three types of changes occur among humans in association with strategy use across time.

(i) Relative frequencies of strategy use vary with time, as mentioned in the discussion of variability above. Both neural network models show changes in relative frequencies of strategy use with increasing simulation trials as discussed in Chapters three and four.

(ii) Effectiveness of implementation of a strategy improves with experience. For example, after some experience with the "minimum" strategy, children execute the same more quickly

and more correctly. The strategy units in both neural network models show higher gains in the total accuracy input that they receive from accuracy inputs with increasing simulation trials and, thus, the activations of the entity units during recall are, in general, higher.

(iii) Acquisition of new strategies: Children as well as adults never stop learning (or discovering) new strategies. For example, children who are familiar with the "sum" and "retrieval" types of strategy proceed to discover the "minimum" strategy (Siegler & Jenkins, 1989). According to Siegler and Shipley (1993), the current models of strategy evolution lack a satisfactory explanation of mechanisms that lead to strategy discovery.

The novelty bias neural network implements the novelty bias mechanism to explain the acquisition of new strategies. In this model, the initiation of a novelty bias for a strategy corresponds to the discovery of the strategy. Thus, this model does not provide a satisfactory explanation of strategy discovery.

However, the components neural network model provides some insights into strategy discovery. According to this model, selective encoding of the accuracy information that results from the use of various strategy components allows strategies to exceed firing thresholds at different time periods. A strategy is discovered when its activation transcends the firing threshold. Thus, this model attributes strategy discovery to accuracy and selective encoding factors in a neural realm. An advantage of the neural network modeling endeavor is highlighted here. The constraints that accompany the design of neural networks have led us to finding these factors.

4. Generalization: It refers to the use of strategies on new problems. Children, for example, extend the "minimum" strategy to new addition problems after they are faced with challenge problems. In the context of the object–target matching task, new problems could imply sequences with variable number of instructions. Currently, the structure of the neural network models does not permit these types of instruction sequences and generalization of the neural network models on strategy use could not, therefore, be tested.

5. Individual Differences: Younger children differ from older children in patterns of strategy use, and mentally retarded individuals from nonretarded individuals on the same. In both neural network models, the activation update equations for strategy units each have a noise term. This term leads to variability in computer simulation of the models from one run to another with respect to strategy selection. As a consequence of this variability, each simulation run may be treated as an "individual" who makes a series of strategy choices, different from another "individual."

The differences between chronological groups, as well as between intelligence groups, may be emulated using the neural network models, as noted in Chapter four. Both models select simpler strategies during the initial phase of a simulation and more advanced strategies during the later phase of a simulation. The mentally retarded individuals and the younger children use simpler strategies and, therefore, their strategy development corresponds to the initial phase in a computer simulation. The nonretarded individuals and the older children use more advanced strategies and, therefore, their strategy development corresponds to the later phase in the simulation.

Thus, the neural network models, investigated in this dissertation, exhibit three of the five phenomena: variability, change, and individual differences. Further work is necessary to determine whether the models exhibit the phenomena of adaptive strategy choice and generalization. Most importantly, it provides an important insight into the acquisition of strategies. According to this insight, strategies are discovered when the components of strategies are continually evaluated for their effectiveness e.g., accuracy and speed. This insight may be of practical use, for example, in training the mentally–retarded individual on strategy use.

5.2 Scope for Further Development of Neural Network Models

1. The strategy pools in the novelty bias neural network and the component neural network are currently implemented by selecting the strategy with highest activation and

setting the other two strategies to zero. For example, a suitable winner–take–all mechanism can be implemented to achieve similar effect.

2. Currently, a noise term is included in the activation update equation of strategy units. This noise term has led to diversity in strategy choice from trial to trial. Without the term, we expect the strategies to evolve from the most simple to the most advanced but without showing diversity behavior. The presence of a similar noise term in the activation update equation of all units in every neural network that we considered should also be explored for its effects on the manifest behavior of the network. For example, a noise term in the activation update equation of accuracy units may simulate the effects of noise in encoding accuracy information on strategy behavior.

3. The generalized Hebbian rule that is used in learning associations in all the neural network models does not lead to the convergence of weights. For this reason, the following rather arbitrary criterion is used in halting the simulation of the sequence generator neural network in Chapter two : stop the simulation when one of the weights in the network reaches a value of 1.0. Further research effort should be devoted in developing halting criteria for learning of weights.

4. Currently, we get anomalous results with respect to the recall of prepositions when "object–target–preposition encoding" strategy has high activation values. As discussed in Chapters three and four, spurious associations occur due to the persistent activity in preposition units. Further work should explore possible solutions, e.g., larger competition within the preposition pool to rectify this anomaly.

5. In the current implementation of strategies, it is assumed that a single strategy is in effect throughout the duration of a trial. However, this assumption is not realistic because subjects in the investigation of Bray et al. (1993) apply multiple strategies in a single trial. Future work should eliminate this restriction.

5.3 Applications

The topic of strategy development and hybrid neural network methodology as discussed in this dissertation is of interest to a variety of problems in cognitive psychology, business and robotics. We identify these problems in this section.

Siegler and Shipley (1993) mention several cognitive problems where the subjects exhibit the same characteristics of strategy selection and evolution as pursued in the neural network models in this dissertation. These problems include arithmetic, time telling, spelling, formation of past tense, causal reasoning, and number conservation. Fletcher and Bray (1993) have more recently devised a "ghost task" that involves placing objects in a given spatial order around the ghost. This task is a modification of the object–target matching problem investigated by Bray et al. (1993) and is supposed to further facilitate the use of external memory strategies by the mental retarded and nonretarded subjects. All the cognitive tasks referred to here are candidates for the development of hybrid neural network models. During the course of current investigation, we gained the insight that selective encoding of accuracy information as it pertains to strategy components may result in the differences in strategy choice among the various chronological and intelligence groups. The generality of this finding should be tested by the construction of neural network models for the aforementioned cognitive tasks.

Waxman and Bachelder (1992) and Grefenstette (1992) discuss the application of neural networks and genetic algorithms to the design of an adaptive autonomous system. Such a system typically operates in an environment that can only be partially modeled or that permits only limited sensing. Constraints such as these introduce uncertainty into the decision–making process of the system. The work carried out in this dissertation offers a framework that enables an autonomous system to overcome these constraints and to adapt its decisions appropriately.

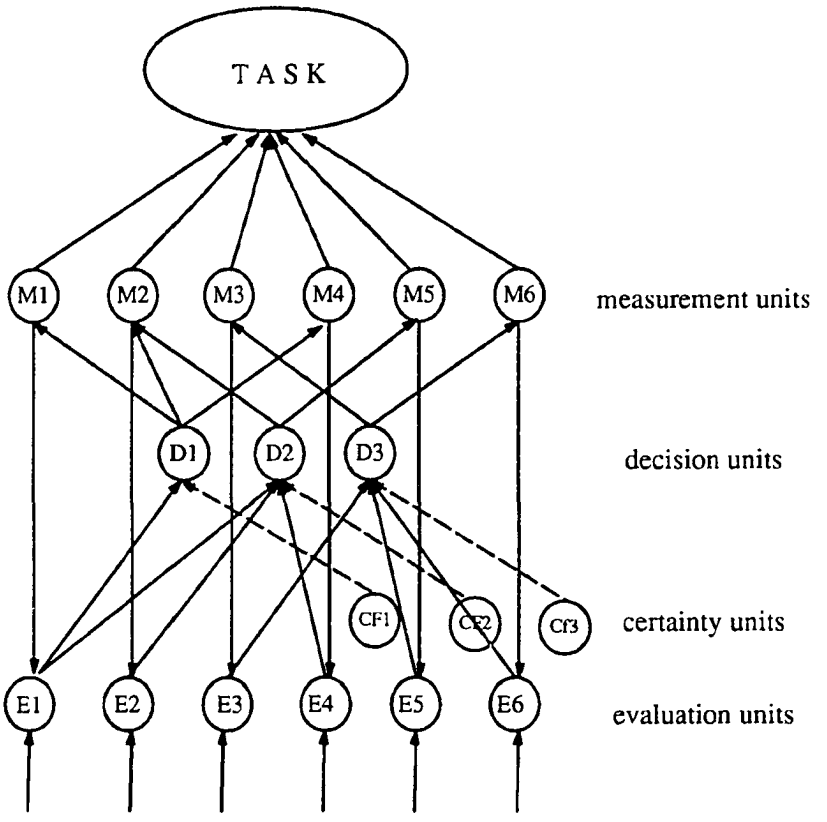


Figure 5.1. A hypothetical neural network for decision–switching in an autonomous system. The connections between certainty units and decision units are shown dashed to distinguish them from the connections between the evaluation units and the decision units.

This framework is illustrated by a conceptual neural network, which is analogous to the novelty bias neural network presented in Chapter three (Figure 5.1). Measurement variables, M1, M2, .., M6 are sensory values after the system applies one or more decisions, D1, D2, and D3 on the task at hand. Certainty factors, CF1, CF2 and CF3, which are analogous to the novelty bias factors in the novelty bias model, are associated with the three decisions D1, D2, and D3, respectively. The evaluation variables, E1, E2, .., E6 corresponding to the measurement variables are built into the system at the time of commissioning of the system. Evaluation may be based on performance criteria such as accuracy and speed. In a semi–autonomous system, the operator of the system interactively supplies inputs to the

evaluation units. At the end of each evaluation cycle, the weights between the decision units and the evaluation units are updated. These weights are responsible for the switch in decision–making in the long–term.

Levine and Leven (1993) illustrate the application of hybrid neural network methodology for modeling irrational behavior of consumers. Specifically, they construct a neural network that simulates consumer behavioral patterns where consumers favor the introduction of "new coke" during market survey but reject it in a buying situation. Our work on strategy development further confirms the utility of the hybrid neural network methodology. Similar to the work of Levine and Leven, models analogous to those developed in the course of this research can be used in predicting shifts in consumer preferences based on criteria such as taste and novelty.

REFERENCES

Albus, J.S. (1991). Outline for a theory of intelligence. *IEEE Trans. Systems, Man and Cybernetics*, 21, No. 3, 473–509.

Anderson, J.A., Silverstein, J.W., Ritz, S.A., & Jones, R.S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, 84, 413–451.

Bjorklund, D.F. & Harnishfeger, K.K.(1990). Children's strategies: Their definitions and origins. In D.F. Bjorklund (Ed.), *Children's strategies: Contemporary views of cognitive development* (pp. 309–323). Hillsdale, N.J.: Lawrence Erlbaum.

Braintenberg, V. (1984). *Vehicles: Experiments in synthetic psychology.* Cambridge, MA: MIT Press.

Bray, N.W., Saarnio, D.A., Borges, L.M., & Hawk, L.W. (1993). Intellectual and developmental differences in external memory strategies. *American Journal on Mental Retardation* (in press).

Brown, A. (1987). Metacognition, executive control, self – regulation and other more mysterious mechanisms. In F.E. Weinert & R.H. Kluwe (Eds.), *Metacognition, motivation, and understanding,* (pp. 65–116). Hillsdale, N.J.: Lawrence Erlbaum.

Edelman, G.M. (1978). Group selection and phasic re–entrant signalling: A theory of higher brain function. In G.M.Edelman & V.B.Mountcastle (Eds.), *The Mindful Brain* (pp.51–100). Cambridge, MA: MIT Press.

Edelman, G.M. (1987). *Neural Darwinism: the theory of neuronal group selection.* Basic Books, New York.

Edelman, G.M. & G.N. Reeke, Jr., (1982). Selective networks capable of representative transformation, limited generalizations, and associative memory. In *Proceedings of National Academy of Sciences (USA),* 79, 2091–2095.

Flavell, J.H. (1987). Speculations about the nature and development of metacognition. In F.E. Weinert & R.H. Kluwe (Eds.), *Metacognition, motivation, and understanding,* (pp. 21–29). Hillsdale, N.J.: Lawrence Erlbaum.

Fletcher, K. & N.W. Bray (1993). External memory strategies in children with and without mild mental retardation. *American Journal of Mental Retardation* (in press).

Grefenstette, J.J. (1992). The evolution of strategies for multiagent environments. *Adaptive Behavior,* 1, No. 1, 65–90.

Grossberg, S. (1969). On the serial learning of lists. *Mathematical Biosciences.* 4, 201–253.

Grossberg, S. (1974). Classical and instrumental learning by neural networks. In R.Rosen & F.Snell (Eds.), *Progress in theoretical biology*, 3, 51–141.

Grossberg, S. (1978). A theory of human memory : self–organization and performance of sensory–motor codes, maps and plans. In R.Rosen & F.Snell (Eds.), *Progress in theoretical biology*, 5, 233–374.

Grossberg, S. & Pepe, J. (1971). Spiking Threshold and overarousal effects in serial learning. *Journal of Statistical Physics*, 3, No.2, 95–125.

Harris, J.E. (1980). Memory aids people use: Two interview studies. *Memory and Cognition*, 8, 31–38.

Hertz, J., A. Krogh, & R.G. Palmer (1991). *Introduction to the theory of neural computation.* Redwood City, CA: Addison – Wesley.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79, 2554–2558.

Hrycej, T. (1992). *Modular learning in neural networks: a modularized approach to neural network classification.* New York: Wiley.

Kleinfeld, D. (1986). Sequential state generation by model neural networks. *Proceedings of the National Academy of Sciences, USA*, 83, 9469–9473.

Levine, D.S. (1991). *Introduction to neural and cognitive modeling.* Hillsdale, N.J.: Erlbaum.

Levine, D.S. & Leven, S. (1993). A gated dipole architecture for multi–drive, multi–attribute decision making. *World Congress on Neural Networks,* International Neural Network Society, Washington, D.C., 1, 495–499.

Levine, D.S. & Prueitt, P.S. (1989). Modeling some effects of frontal lobe damage–novely and perseveration. *Neural Networks*, 2, 103–116.

Luce, R.D. (1959). *Individual choice behavior.* New York: Wiley.

McClelland, J.L. (1991). Stochastic interactive processes and the effects of context on perception. *Cognitive Psychology*, 23, 1–44.

McClelland, J.L. & Rumelhart, D.E. (1988). *Explorations in parallel distributed processing: a handbook of models, programs, and exercises.* MIT Press, Cambridge, MA.

McGilly, K. & Siegler, R.S. (1989). How children choose among serial recall strategies. *Child Development*, 60, 172–182.

Norman, D.A. (1980). Twelve issues for cognitive science. *Cognitive Science*, 4, 1–32.

Pearson, J.C., L.H. Finkel, & G.M. Edelman (1987). Plasticity in the organization of adult cortical maps: A computer model based on neuronal group selection. *Journal of Neuroscience.* 7, No.12, 4209–4223.

Reeke, G.N., Jr., Sporns, O. & Edelman, G.M. (1990). Synthetic neural modeling: the Darwin series of recognition automata. *Proceedings of the IEEE*, 78, No.9, 1498–1530.

Reilly, K. D., Amthor, F., McAnulty, M. A., Thurston, P. W., Villa, M. F., & Wainer, M. S. (1987). Neural Network Modeling and the Neuronal Robot. In Jordan Q. B.Chou, (Ed.) *Proc. 1987 Summer Computer Simulation Conference* (pp. 448–453). San Diego, CA: Simulation Councils.

Reilly, K.D. & Villa, M. (1990). A barrels organization for data fusion and communication in neural network systems. *Procs. First Workshop on Neural Networks: Academic–Industrial–NASA–Defense 90–WNN–AIND* (pp.115–122). San Diego, CA: Simulation Councils.

Siegler, R.S. (1991). *Children's Thinking*. Prentice Hall, Englewood Cliffs, NJ.

Siegler, R.S. & Jenkins, E. (1989). *How children discover new strategies*. Lawrence Erlbaum, Hillsdale, N.J.

Siegler, R.S. & Shipley, C. (1993). Variation, selection, and cognitive change. In G. Halford and T. Simon (Eds.), *Developing cognitive competence: new approaches to process modeling*. Lawrence Erlbaum Associates, Hillsdale, N.J. (in press).

Sternberg, R.J. (1985). *Beyond IQ: A triarchic theory of human intelligence*. Cambridge University Press.

Sternberg, R.J. & Rifkin, B. (1979). The development of analogical reasoning processes. *Journal of Experimental Child Psychology*, 27, 195–232.

Sompolinsky, H. & Kanter, I. (1986). Temporal association in asymmetric neural networks. *Physical Review Letters*, 57, 2861–2864.

Waxman, A.M. & Bachelder, I. (1992). Neural networks for mobile robot visual navigation and behavioral conditioning. *Research conference on Neural networks for Learning, Recognition, and Control*, Department of Cognitive and Neural Systems, Boston University, May 14–16, p.10.

Villa, M.F. & Reilly, K.D. (1992). Hierarchical structures in hybrid systems. In A.Kandel and G.Langholz (Eds.) *Hybrid architectures for intelligent systems* (pp.221–254). Boca Raton, FL: CRC Press.

APPENDIX A

SIMULATIONS FOR THE NOVELTY BIAS NEURAL NETWORK

In the following simulation runs, the entries represent the following:

Column	Representation
1	Trial number
2	net novelty bias input to strategy unit 1
3	net accuracy input to strategy unit 1
4	net novelty bias input to strategy unit 2
5	net accuracy input to strategy unit 2
6	net novelty bias to strategy unit 3
7	net accuracy input to strategy unit 3
8	winning strategy
9	activation of winning strategy

Simulation Run 1

1	0.682	0.150	0.000	0.150	0.000	0.150	1	0.832
2	0.431	0.169	0.000	0.150	0.000	0.150	1	0.600
3	0.269	0.182	0.000	0.150	0.000	0.150	1	0.450
4	0.277	0.191	0.000	0.150	0.000	0.150	1	0.468
5	0.412	0.200	0.000	0.150	0.000	0.150	1	0.613
6	0.280	0.217	0.000	0.150	0.000	0.150	1	0.496
7	0.381	0.227	0.000	0.150	0.000	0.150	1	0.607
8	0.332	0.240	0.000	0.150	0.000	0.150	1	0.572
9	0.067	0.252	0.000	0.150	0.000	0.150	1	0.319
10	0.166	0.258	0.000	0.150	0.000	0.150	1	0.424
11	0.202	0.267	0.000	0.150	0.000	0.150	1	0.468
12	0.388	0.276	0.000	0.150	0.000	0.150	1	0.664
13	0.130	0.297	0.614	0.150	0.000	0.150	2	0.764
14	0.252	0.297	0.080	0.177	0.000	0.150	1	0.549
15	0.010	0.310	0.572	0.177	0.000	0.150	2	0.748
16	0.074	0.310	0.300	0.205	0.000	0.150	2	0.505
17	0.029	0.310	0.067	0.221	0.000	0.150	1	0.340
18	0.097	0.317	0.550	0.221	0.000	0.150	2	0.771
19	0.094	0.317	0.411	0.248	0.000	0.150	2	0.659
20	0.046	0.317	0.172	0.278	0.000	0.150	2	0.450
21	0.190	0.317	0.425	0.288	0.000	0.150	2	0.712
22	0.229	0.317	0.414	0.316	0.000	0.150	2	0.730
23	0.170	0.317	0.098	0.343	0.000	0.150	1	0.487
24	0.030	0.327	0.111	0.343	0.000	0.150	2	0.454
25	0.073	0.327	0.093	0.353	0.584	0.150	3	0.734
26	0.169	0.327	0.097	0.353	0.160	0.197	1	0.496
27	0.023	0.337	0.074	0.353	0.615	0.197	3	0.812
28	0.159	0.337	0.102	0.353	0.314	0.252	3	0.567
29	0.147	0.337	0.001	0.353	0.164	0.276	1	0.484
30	0.057	0.347	0.284	0.353	0.131	0.276	2	0.637
31	0.050	0.347	0.172	0.381	0.380	0.276	3	0.656
32	0.008	0.347	0.227	0.381	0.197	0.320	2	0.608
33	0.011	0.347	0.097	0.396	0.096	0.320	2	0.493
34	0.111	0.347	0.058	0.408	0.389	0.320	3	0.709
35	0.045	0.347	0.139	0.408	0.026	0.366	2	0.546
36	0.067	0.347	0.118	0.424	0.046	0.366	2	0.543
37	0.008	0.347	0.146	0.441	0.163	0.366	2	0.587
38	0.093	0.347	0.061	0.456	0.259	0.366	3	0.626
39	0.069	0.347	0.182	0.456	0.080	0.407	2	0.637
40	0.025	0.347	0.053	0.483	0.192	0.407	3	0.599
41	0.003	0.347	0.064	0.483	0.141	0.434	3	0.574
42	0.012	0.347	0.062	0.483	0.305	0.459	3	0.765
43	0.061	0.347	0.107	0.483	0.132	0.508	3	0.641
44	0.064	0.347	0.090	0.483	0.197	0.549	3	0.747
45	0.036	0.347	0.010	0.483	0.172	0.600	3	0.771
46	0.025	0.347	0.063	0.483	0.240	0.650	3	0.890
47	0.010	0.347	0.108	0.483	0.163	0.706	3	0.870
48	0.007	0.347	0.079	0.483	0.156	0.761	3	0.917
49	0.051	0.347	0.002	0.483	0.078	0.822	3	0.900
50	0.024	0.347	0.028	0.483	0.131	0.880	3	1.000

Simulation Run 2

1	0.499	0.150	0.000	0.150	0.000	0.150	1	0.649
2	0.193	0.163	0.000	0.150	0.000	0.150	1	0.356
3	0.719	0.168	0.000	0.150	0.000	0.150	1	0.887
4	0.141	0.178	0.000	0.150	0.000	0.150	1	0.319
5	0.027	0.182	0.000	0.150	0.000	0.150	1	0.209
6	0.120	0.185	0.000	0.150	0.000	0.150	1	0.305
7	0.525	0.189	0.000	0.150	0.000	0.150	1	0.713
8	0.417	0.201	0.000	0.150	0.000	0.150	1	0.618
9	0.076	0.212	0.000	0.150	0.000	0.150	1	0.288
10	0.084	0.215	0.000	0.150	0.000	0.150	1	0.299
11	0.275	0.219	0.000	0.150	0.000	0.150	1	0.494
12	0.492	0.226	0.000	0.150	0.000	0.150	1	0.717
13	0.256	0.238	0.042	0.150	0.000	0.150	1	0.495
14	0.425	0.245	0.416	0.150	0.000	0.150	1	0.670
15	0.397	0.258	0.083	0.150	0.000	0.150	1	0.655
16	0.257	0.271	0.697	0.150	0.000	0.150	2	0.847
17	0.232	0.271	0.340	0.160	0.000	0.150	1	0.503
18	0.214	0.279	0.249	0.160	0.000	0.150	1	0.493
19	0.218	0.286	0.066	0.160	0.000	0.150	1	0.504
20	0.240	0.295	0.368	0.160	0.000	0.150	1	0.534
21	0.132	0.303	0.311	0.160	0.000	0.150	2	0.471
22	0.175	0.303	0.120	0.167	0.000	0.150	1	0.478
23	0.084	0.310	0.392	0.167	0.000	0.150	2	0.559
24	0.060	0.310	0.349	0.176	0.000	0.150	2	0.525
25	0.248	0.310	0.012	0.186	0.217	0.150	1	0.557
26	0.263	0.317	0.433	0.186	0.221	0.150	2	0.619
27	0.252	0.317	0.071	0.200	0.354	0.150	1	0.570
28	0.108	0.325	0.289	0.200	0.540	0.150	3	0.690
29	0.110	0.325	0.404	0.200	0.641	0.178	3	0.818
30	0.040	0.325	0.100	0.200	0.378	0.212	3	0.589
31	0.224	0.325	0.359	0.200	0.570	0.228	3	0.798
32	0.102	0.325	0.053	0.200	0.016	0.262	1	0.427
33	0.212	0.331	0.291	0.200	0.476	0.262	3	0.738
34	0.183	0.331	0.283	0.200	0.419	0.292	3	0.710
35	0.146	0.331	0.273	0.200	0.260	0.321	3	0.580
36	0.092	0.331	0.222	0.200	0.336	0.338	3	0.674
37	0.177	0.331	0.242	0.200	0.132	0.365	1	0.508
38	0.090	0.339	0.197	0.200	0.352	0.365	3	0.718
39	0.036	0.339	0.062	0.200	0.404	0.394	3	0.798
40	0.150	0.339	0.010	0.200	0.109	0.428	3	0.537
41	0.080	0.339	0.215	0.200	0.171	0.444	3	0.615
42	0.009	0.339	0.109	0.200	0.266	0.467	3	0.733
43	0.041	0.339	0.090	0.200	0.304	0.496	3	0.800
44	0.074	0.339	0.014	0.200	0.064	0.530	3	0.594
45	0.124	0.339	0.015	0.200	0.121	0.548	3	0.669
46	0.067	0.339	0.197	0.200	0.032	0.577	3	0.609
47	0.078	0.339	0.103	0.200	0.113	0.598	3	0.711
48	0.030	0.339	0.118	0.200	0.063	0.628	3	0.691
49	0.066	0.339	0.100	0.200	0.078	0.656	3	0.734
50	0.078	0.339	0.060	0.200	0.112	0.685	3	0.797
51	0.049	0.339	0.127	0.200	0.265	0.719	3	0.984
52	0.036	0.339	0.087	0.200	0.061	0.773	3	0.834
53	0.015	0.339	0.012	0.200	0.249	0.805	3	1.000

Simulation Run 3

1	0.671	0.150	0.000	0.150	0.000	0.150	1	0.821
2	0.621	0.163	0.000	0.150	0.000	0.150	1	0.784
3	0.635	0.174	0.000	0.150	0.000	0.150	1	0.810
4	0.513	0.187	0.000	0.150	0.000	0.150	1	0.700
5	0.088	0.200	0.000	0.150	0.000	0.150	1	0.288
6	0.206	0.204	0.000	0.150	0.000	0.150	1	0.410
7	0.051	0.209	0.000	0.150	0.000	0.150	1	0.260
8	0.443	0.213	0.000	0.150	0.000	0.150	1	0.655
9	0.477	0.226	0.000	0.150	0.000	0.150	1	0.703
10	0.368	0.239	0.000	0.150	0.000	0.150	1	0.606
11	0.276	0.247	0.000	0.150	0.000	0.150	1	0.523
12	0.190	0.256	0.000	0.150	0.000	0.150	1	0.445
13	0.464	0.262	0.069	0.150	0.000	0.150	1	0.726
14	0.027	0.274	0.657	0.150	0.000	0.150	2	0.807
15	0.328	0.274	0.351	0.166	0.000	0.150	1	0.602
16	0.316	0.283	0.340	0.166	0.000	0.150	1	0.598
17	0.351	0.291	0.410	0.166	0.000	0.150	1	0.642
18	0.369	0.304	0.059	0.166	0.000	0.150	1	0.673
19	0.270	0.317	0.600	0.166	0.000	0.150	2	0.766
20	0.354	0.317	0.532	0.182	0.000	0.150	2	0.714
21	0.137	0.317	0.550	0.198	0.000	0.150	2	0.748
22	0.092	0.317	0.097	0.215	0.000	0.150	1	0.409
23	0.101	0.322	0.041	0.215	0.000	0.150	1	0.423
24	0.147	0.328	0.054	0.215	0.000	0.150	1	0.474
25	0.242	0.334	0.275	0.215	0.474	0.150	3	0.624
26	0.020	0.334	0.286	0.215	0.478	0.175	3	0.653
27	0.044	0.334	0.158	0.215	0.072	0.204	1	0.378
28	0.001	0.339	0.343	0.215	0.041	0.204	2	0.558
29	0.053	0.339	0.076	0.224	0.238	0.204	3	0.442
30	0.072	0.339	0.321	0.224	0.462	0.214	3	0.677
31	0.179	0.339	0.274	0.224	0.028	0.243	1	0.518
32	0.017	0.347	0.174	0.224	0.092	0.243	2	0.398
33	0.064	0.347	0.256	0.230	0.029	0.243	2	0.486
34	0.170	0.347	0.035	0.237	0.496	0.243	3	0.739
35	0.042	0.347	0.160	0.237	0.198	0.272	3	0.470
36	0.112	0.347	0.051	0.237	0.396	0.285	3	0.681
37	0.061	0.347	0.078	0.237	0.296	0.313	3	0.609
38	0.130	0.347	0.115	0.237	0.291	0.334	3	0.625

39	0.032	0.347	0.012	0.237	0.217	0.361	3	0.578
40	0.010	0.347	0.228	0.237	0.432	0.376	3	0.808
41	0.137	0.347	0.033	0.237	0.341	0.409	3	0.750
42	0.059	0.347	0.082	0.237	0.239	0.441	3	0.680
43	0.081	0.347	0.188	0.237	0.135	0.469	3	0.604
44	0.064	0.347	0.166	0.237	0.024	0.486	3	0.510
45	0.044	0.347	0.120	0.237	0.084	0.504	3	0.588
46	0.085	0.347	0.144	0.237	0.317	0.520	3	0.836
47	0.044	0.347	0.157	0.237	0.168	0.552	3	0.720
48	0.103	0.347	0.060	0.237	0.183	0.581	3	0.763
49	0.089	0.347	0.114	0.237	0.004	0.611	3	0.615
50	0.075	0.347	0.164	0.237	0.155	0.632	3	0.787
51	0.045	0.347	0.057	0.237	0.157	0.661	3	0.818
52	0.008	0.347	0.066	0.237	0.024	0.695	3	0.719
53	0.029	0.347	0.019	0.237	0.043	0.723	3	0.765
54	0.067	0.347	0.064	0.237	0.218	0.754	3	0.972
55	0.048	0.347	0.016	0.237	0.050	0.806	3	0.856
56	0.030	0.347	0.009	0.237	0.150	0.836	3	0.987
57	0.001	0.347	0.009	0.237	0.206	0.882	3	1.000

Simulation Run 4

1	0.092	0.150	0.000	0.150	0.000	0.150	1	0.242
2	0.386	0.153	0.000	0.150	0.000	0.150	1	0.539
3	0.513	0.162	0.000	0.150	0.000	0.150	1	0.674
4	0.313	0.175	0.000	0.150	0.000	0.150	1	0.488
5	0.125	0.181	0.000	0.150	0.000	0.150	1	0.306
6	0.210	0.185	0.000	0.150	0.000	0.150	1	0.395
7	0.234	0.190	0.000	0.150	0.000	0.150	1	0.424
8	0.521	0.196	0.000	0.150	0.000	0.150	1	0.717
9	0.391	0.208	0.000	0.150	0.000	0.150	1	0.599
10	0.477	0.217	0.000	0.150	0.000	0.150	1	0.693
11	0.418	0.229	0.000	0.150	0.000	0.150	1	0.648
12	0.047	0.243	0.000	0.150	0.000	0.150	1	0.290
13	0.481	0.246	0.623	0.150	0.000	0.150	2	0.773
14	0.164	0.246	0.328	0.166	0.000	0.150	2	0.495
15	0.311	0.246	0.425	0.174	0.000	0.150	2	0.599
16	0.074	0.246	0.128	0.184	0.000	0.150	1	0.320
17	0.231	0.250	0.510	0.184	0.000	0.150	2	0.694
18	0.299	0.250	0.591	0.201	0.000	0.150	2	0.792
19	0.306	0.250	0.398	0.214	0.000	0.150	2	0.613
20	0.015	0.250	0.535	0.228	0.000	0.150	2	0.763
21	0.320	0.250	0.394	0.244	0.000	0.150	2	0.638
22	0.240	0.250	0.059	0.262	0.000	0.150	1	0.490
23	0.224	0.257	0.282	0.262	0.000	0.150	2	0.544
24	0.197	0.257	0.065	0.273	0.000	0.150	1	0.454
25	0.048	0.263	0.164	0.273	0.210	0.150	2	0.437
26	0.179	0.263	0.202	0.279	0.482	0.150	3	0.632
27	0.144	0.263	0.376	0.279	0.074	0.177	2	0.656
28	0.035	0.263	0.283	0.298	0.210	0.177	2	0.581
29	0.166	0.263	0.135	0.307	0.656	0.177	3	0.833
30	0.084	0.263	0.249	0.307	0.448	0.209	3	0.658
31	0.188	0.263	0.302	0.307	0.161	0.238	2	0.609
32	0.179	0.263	0.270	0.321	0.589	0.238	3	0.827
33	0.079	0.263	0.037	0.321	0.262	0.274	3	0.536
34	0.121	0.263	0.071	0.321	0.477	0.290	3	0.767
35	0.161	0.263	0.048	0.321	0.306	0.321	3	0.627
36	0.175	0.263	0.294	0.321	0.078	0.347	2	0.615
37	0.139	0.263	0.242	0.334	0.040	0.347	2	0.576
38	0.086	0.263	0.197	0.343	0.458	0.347	3	0.805
39	0.139	0.263	0.022	0.343	0.417	0.380	3	0.798
40	0.069	0.263	0.127	0.343	0.298	0.414	3	0.712
41	0.114	0.263	0.179	0.343	0.176	0.444	3	0.620
42	0.076	0.263	0.221	0.343	0.198	0.466	3	0.664
43	0.129	0.263	0.150	0.343	0.256	0.495	3	0.751
44	0.103	0.263	0.052	0.343	0.010	0.527	3	0.536
45	0.036	0.263	0.128	0.343	0.013	0.544	3	0.557
46	0.125	0.263	0.116	0.343	0.330	0.558	3	0.888
47	0.022	0.263	0.175	0.343	0.050	0.591	3	0.641
48	0.033	0.263	0.030	0.343	0.011	0.619	3	0.630
49	0.070	0.263	0.158	0.343	0.236	0.644	3	0.880
50	0.062	0.263	0.029	0.343	0.204	0.676	3	0.880
51	0.065	0.263	0.011	0.343	0.049	0.709	3	0.758
52	0.026	0.263	0.032	0.343	0.231	0.740	3	0.971
53	0.031	0.263	0.084	0.343	0.100	0.795	3	0.895
54	0.090	0.263	0.066	0.343	0.133	0.829	3	0.962
55	0.007	0.263	0.020	0.343	0.187	0.877	3	1.000

Simulation Run 5

1	0.723	0.150	0.000	0.150	0.000	0.150	1	0.873
2	0.087	0.160	0.000	0.150	0.000	0.150	1	0.247
3	0.479	0.163	0.000	0.150	0.000	0.150	1	0.642
4	0.236	0.176	0.000	0.150	0.000	0.150	1	0.412
5	0.679	0.182	0.000	0.150	0.000	0.150	1	0.861
6	0.604	0.192	0.000	0.150	0.000	0.150	1	0.796
7	0.490	0.203	0.000	0.150	0.000	0.150	1	0.693
8	0.469	0.216	0.000	0.150	0.000	0.150	1	0.685
9	0.510	0.229	0.000	0.150	0.000	0.150	1	0.740
10	0.054	0.242	0.000	0.150	0.000	0.150	1	0.296
11	0.295	0.246	0.000	0.150	0.000	0.150	1	0.541
12	0.382	0.255	0.000	0.150	0.000	0.150	1	0.637
13	0.034	0.268	0.007	0.150	0.000	0.150	1	0.302
14	0.382	0.271	0.402	0.150	0.000	0.150	1	0.653

15	0.292	0.285	0.435	0.150	0.000	0.150	2	0.585
16	0.360	0.285	0.041	0.159	0.000	0.150	1	0.645
17	0.397	0.298	0.304	0.159	0.000	0.150	1	0.695
18	0.251	0.311	0.303	0.159	0.000	0.150	1	0.562
19	0.087	0.318	0.612	0.159	0.000	0.150	2	0.771
20	0.107	0.318	0.278	0.176	0.000	0.150	2	0.454
21	0.198	0.318	0.358	0.182	0.000	0.150	2	0.540
22	0.180	0.318	0.228	0.193	0.000	0.150	1	0.498
23	0.190	0.325	0.048	0.193	0.000	0.150	1	0.515
24	0.138	0.333	0.134	0.193	0.000	0.150	1	0.471
25	0.282	0.340	0.485	0.193	0.673	0.150	3	0.823
26	0.017	0.340	0.182	0.193	0.252	0.185	3	0.436
27	0.258	0.340	0.102	0.193	0.061	0.195	1	0.598
28	0.112	0.348	0.071	0.193	0.329	0.195	3	0.524
29	0.052	0.348	0.357	0.193	0.530	0.211	3	0.741
30	0.140	0.348	0.180	0.193	0.616	0.242	3	0.858
31	0.025	0.348	0.250	0.193	0.618	0.272	3	0.890
32	0.043	0.348	0.296	0.193	0.374	0.305	3	0.679
33	0.113	0.348	0.172	0.193	0.251	0.334	3	0.585
34	0.030	0.348	0.234	0.193	0.459	0.350	3	0.809
35	0.045	0.348	0.203	0.193	0.265	0.383	3	0.648
36	0.185	0.348	0.189	0.193	0.157	0.411	3	0.568
37	0.003	0.348	0.226	0.193	0.185	0.426	3	0.612
38	0.025	0.348	0.020	0.193	0.290	0.447	3	0.737
39	0.015	0.348	0.112	0.193	0.050	0.478	3	0.528
40	0.152	0.348	0.130	0.193	0.345	0.494	3	0.839
41	0.004	0.348	0.175	0.193	0.249	0.524	3	0.773
42	0.000	0.348	0.022	0.193	0.091	0.556	3	0.647
43	0.048	0.348	0.087	0.193	0.288	0.583	3	0.870
44	0.071	0.348	0.038	0.193	0.286	0.614	3	0.899
45	0.067	0.348	0.136	0.193	0.270	0.648	3	0.919
46	0.036	0.348	0.123	0.193	0.327	0.683	3	1.000

Simulation Run 6

1	0.172	0.150	0.000	0.150	0.000	0.150	1	0.322
2	0.393	0.154	0.000	0.150	0.000	0.150	1	0.547
3	0.027	0.163	0.000	0.150	0.000	0.150	1	0.190
4	0.685	0.163	0.000	0.150	0.000	0.150	1	0.848
5	0.513	0.173	0.000	0.150	0.000	0.150	1	0.686
6	0.565	0.185	0.000	0.150	0.000	0.150	1	0.751
7	0.503	0.199	0.000	0.150	0.000	0.150	1	0.701
8	0.517	0.212	0.000	0.150	0.000	0.150	1	0.729
9	0.316	0.224	0.000	0.150	0.000	0.150	1	0.540
10	0.265	0.233	0.000	0.150	0.000	0.150	1	0.497
11	0.426	0.240	0.000	0.150	0.000	0.150	1	0.666
12	0.266	0.253	0.000	0.150	0.000	0.150	1	0.519
13	0.261	0.262	0.646	0.150	0.000	0.150	2	0.796
14	0.045	0.262	0.153	0.164	0.000	0.150	2	0.316
15	0.256	0.262	0.008	0.168	0.000	0.150	1	0.518
16	0.319	0.270	0.403	0.168	0.000	0.150	1	0.588
17	0.060	0.278	0.500	0.168	0.000	0.150	2	0.668
18	0.399	0.278	0.534	0.187	0.000	0.150	2	0.721
19	0.298	0.278	0.200	0.203	0.000	0.150	1	0.575
20	0.090	0.286	0.094	0.203	0.000	0.150	1	0.376
21	0.289	0.291	0.366	0.203	0.000	0.150	1	0.580
22	0.091	0.299	0.221	0.203	0.000	0.150	2	0.425
23	0.127	0.299	0.169	0.210	0.000	0.150	1	0.425
24	0.161	0.304	0.051	0.210	0.000	0.150	1	0.465
25	0.195	0.311	0.373	0.210	0.269	0.150	2	0.583
26	0.053	0.311	0.349	0.219	0.246	0.150	2	0.567
27	0.235	0.311	0.298	0.228	0.477	0.150	3	0.627
28	0.214	0.311	0.335	0.228	0.043	0.175	2	0.562
29	0.212	0.311	0.128	0.236	0.350	0.175	3	0.525
30	0.118	0.311	0.214	0.236	0.562	0.192	3	0.754
31	0.164	0.311	0.279	0.236	0.489	0.223	3	0.712
32	0.018	0.311	0.130	0.236	0.111	0.252	2	0.366
33	0.015	0.311	0.168	0.241	0.059	0.252	2	0.410
34	0.162	0.311	0.079	0.247	0.350	0.252	3	0.602
35	0.109	0.311	0.277	0.247	0.214	0.269	2	0.524
36	0.145	0.311	0.177	0.258	0.161	0.269	1	0.455
37	0.105	0.317	0.230	0.258	0.375	0.269	3	0.644
38	0.165	0.317	0.210	0.258	0.218	0.296	3	0.514
39	0.131	0.317	0.010	0.258	0.000	0.312	1	0.448
40	0.145	0.323	0.161	0.258	0.237	0.312	3	0.550
41	0.128	0.323	0.128	0.258	0.361	0.329	3	0.690
42	0.101	0.323	0.127	0.258	0.073	0.358	3	0.431
43	0.129	0.323	0.014	0.258	0.162	0.367	3	0.528
44	0.106	0.323	0.161	0.258	0.299	0.382	3	0.681
45	0.101	0.323	0.028	0.258	0.310	0.412	3	0.722
46	0.018	0.323	0.112	0.258	0.131	0.440	3	0.571
47	0.002	0.323	0.021	0.258	0.227	0.455	3	0.682
48	0.052	0.323	0.050	0.258	0.042	0.484	3	0.526
49	0.066	0.323	0.056	0.258	0.087	0.500	3	0.587
50	0.091	0.323	0.041	0.258	0.109	0.516	3	0.625
51	0.018	0.323	0.138	0.258	0.107	0.542	3	0.649
52	0.029	0.323	0.126	0.258	0.233	0.569	3	0.803
53	0.066	0.323	0.057	0.258	0.241	0.602	3	0.843
54	0.049	0.323	0.020	0.258	0.067	0.632	3	0.699
55	0.013	0.323	0.000	0.258	0.026	0.661	3	0.687
56	0.006	0.323	0.114	0.258	0.164	0.688	3	0.852
57	0.009	0.323	0.010	0.258	0.033	0.719	3	0.753
58	0.046	0.323	0.064	0.258	0.170	0.750	3	0.920
59	0.037	0.323	0.099	0.258	0.075	0.785	3	0.861
60	0.038	0.323	0.013	0.258	0.137	0.817	3	0.954
61	0.063	0.323	0.042	0.258	0.101	0.864	3	0.964
62	0.028	0.323	0.030	0.258	0.167	0.914	3	1.000

Simulation Run 7

1	0.750	0.150	0.000	0.150	0.000	0.150	1	0.900
2	0.479	0.160	0.000	0.150	0.000	0.150	1	0.640
3	0.275	0.173	0.000	0.150	0.000	0.150	1	0.448
4	0.065	0.179	0.000	0.150	0.000	0.150	1	0.244
5	0.418	0.182	0.000	0.150	0.000	0.150	1	0.600
6	0.633	0.191	0.000	0.150	0.000	0.150	1	0.824
7	0.622	0.204	0.000	0.150	0.000	0.150	1	0.826
8	0.061	0.217	0.000	0.150	0.000	0.150	1	0.278
9	0.551	0.221	0.000	0.150	0.000	0.150	1	0.772
10	0.481	0.233	0.000	0.150	0.000	0.150	1	0.714
11	0.140	0.246	0.000	0.150	0.000	0.150	1	0.386
12	0.493	0.251	0.000	0.150	0.000	0.150	1	0.743
13	0.296	0.264	0.512	0.150	0.000	0.150	2	0.662
14	0.099	0.264	0.604	0.169	0.000	0.150	2	0.773
15	0.091	0.264	0.493	0.186	0.000	0.150	2	0.679
16	0.099	0.264	0.681	0.204	0.000	0.150	2	0.885
17	0.260	0.264	0.172	0.214	0.000	0.150	1	0.524
18	0.202	0.272	0.515	0.214	0.000	0.150	2	0.728
19	0.048	0.272	0.624	0.230	0.000	0.150	2	0.854
20	0.335	0.272	0.213	0.240	0.000	0.150	1	0.607
21	0.194	0.281	0.226	0.240	0.000	0.150	1	0.475
22	0.208	0.287	0.362	0.240	0.000	0.150	2	0.602
23	0.326	0.287	0.137	0.250	0.000	0.150	1	0.613
24	0.027	0.298	0.367	0.250	0.000	0.150	2	0.616
25	0.207	0.298	0.074	0.263	0.714	0.150	3	0.864
26	0.109	0.298	0.249	0.263	0.383	0.181	3	0.563
27	0.016	0.298	0.205	0.263	0.533	0.197	3	0.729
28	0.054	0.298	0.068	0.263	0.160	0.225	3	0.385
29	0.151	0.298	0.338	0.263	0.585	0.234	3	0.818
30	0.046	0.298	0.292	0.263	0.161	0.268	2	0.555
31	0.077	0.298	0.261	0.272	0.558	0.268	3	0.826
32	0.203	0.298	0.110	0.272	0.474	0.303	3	0.777
33	0.014	0.298	0.078	0.272	0.040	0.335	3	0.375
34	0.000	0.298	0.315	0.272	0.396	0.342	3	0.738
35	0.047	0.298	0.067	0.272	0.227	0.371	3	0.598
36	0.115	0.298	0.239	0.272	0.070	0.389	2	0.510
37	0.119	0.298	0.018	0.283	0.123	0.389	3	0.512
38	0.152	0.298	0.221	0.283	0.168	0.406	3	0.574
39	0.164	0.298	0.104	0.283	0.108	0.422	3	0.530
40	0.116	0.298	0.142	0.283	0.425	0.438	3	0.863
41	0.156	0.298	0.068	0.283	0.320	0.469	3	0.789
42	0.060	0.298	0.202	0.283	0.377	0.498	3	0.875
43	0.097	0.298	0.087	0.283	0.250	0.530	3	0.779
44	0.086	0.298	0.222	0.283	0.258	0.560	3	0.819
45	0.002	0.298	0.086	0.283	0.107	0.594	3	0.702
46	0.060	0.298	0.071	0.283	0.105	0.623	3	0.728
47	0.031	0.298	0.075	0.283	0.110	0.651	3	0.761
48	0.043	0.298	0.122	0.283	0.275	0.682	3	0.958
49	0.017	0.298	0.045	0.283	0.008	0.729	3	0.737
50	0.030	0.298	0.099	0.283	0.121	0.759	3	0.880
51	0.089	0.298	0.030	0.283	0.245	0.792	3	1.000

Simulation Run 8

1	0.747	0.150	0.000	0.150	0.000	0.150	1	0.897
2	0.103	0.160	0.000	0.150	0.000	0.150	1	0.263
3	0.715	0.164	0.000	0.150	0.000	0.150	1	0.879
4	0.415	0.174	0.000	0.150	0.000	0.150	1	0.589
5	0.254	0.182	0.000	0.150	0.000	0.150	1	0.436
6	0.432	0.188	0.000	0.150	0.000	0.150	1	0.620
7	0.253	0.199	0.000	0.150	0.000	0.150	1	0.451
8	0.100	0.205	0.000	0.150	0.000	0.150	1	0.305
9	0.577	0.209	0.000	0.150	0.000	0.150	1	0.785
10	0.177	0.220	0.000	0.150	0.000	0.150	1	0.397
11	0.311	0.225	0.000	0.150	0.000	0.150	1	0.536
12	0.437	0.234	0.000	0.150	0.000	0.150	1	0.671
13	0.215	0.247	0.274	0.150	0.000	0.150	1	0.462
14	0.442	0.253	0.098	0.150	0.000	0.150	1	0.695
15	0.063	0.266	0.213	0.150	0.000	0.150	2	0.363
16	0.268	0.266	0.642	0.155	0.000	0.150	2	0.797
17	0.048	0.266	0.249	0.173	0.000	0.150	2	0.421
18	0.151	0.266	0.496	0.179	0.000	0.150	2	0.675
19	0.158	0.266	0.211	0.197	0.000	0.150	1	0.423
20	0.102	0.271	0.113	0.197	0.000	0.150	1	0.374
21	0.209	0.276	0.169	0.197	0.000	0.150	1	0.485
22	0.299	0.283	0.177	0.197	0.000	0.150	1	0.582
23	0.244	0.291	0.491	0.197	0.000	0.150	2	0.688
24	0.254	0.291	0.018	0.214	0.000	0.150	1	0.545
25	0.079	0.299	0.115	0.214	0.650	0.150	3	0.800
26	0.254	0.299	0.417	0.214	0.025	0.184	2	0.630
27	0.130	0.299	0.349	0.231	0.143	0.184	2	0.580
28	0.123	0.299	0.174	0.240	0.626	0.184	3	0.810
29	0.102	0.299	0.314	0.240	0.676	0.218	3	0.894
30	0.222	0.299	0.134	0.240	0.186	0.252	1	0.521
31	0.061	0.308	0.232	0.240	0.448	0.252	3	0.700
32	0.021	0.308	0.141	0.240	0.576	0.280	3	0.856
33	0.022	0.308	0.344	0.240	0.066	0.312	2	0.584
34	0.158	0.308	0.055	0.250	0.475	0.312	3	0.786
35	0.057	0.308	0.075	0.250	0.148	0.341	3	0.489
36	0.141	0.308	0.100	0.250	0.262	0.354	3	0.616
37	0.031	0.308	0.234	0.250	0.034	0.375	2	0.484
38	0.110	0.308	0.157	0.257	0.437	0.375	3	0.813
39	0.145	0.308	0.205	0.257	0.212	0.409	3	0.621
40	0.106	0.308	0.263	0.257	0.154	0.430	3	0.584

41	0.065	0.308	0.024	0.257	0.298	0.446	3	0.744
42	0.031	0.308	0.181	0.257	0.126	0.477	3	0.603
43	0.114	0.308	0.128	0.257	0.201	0.494	3	0.695
44	0.124	0.308	0.156	0.257	0.211	0.522	3	0.734
45	0.072	0.308	0.114	0.257	0.229	0.553	3	0.782
46	0.087	0.308	0.058	0.257	0.317	0.583	3	0.900
47	0.058	0.308	0.060	0.257	0.232	0.616	3	0.849
48	0.084	0.308	0.090	0.257	0.301	0.647	3	0.948
49	0.104	0.308	0.080	0.257	0.060	0.686	3	0.746
50	0.048	0.308	0.013	0.257	0.173	0.717	3	0.891
51	0.065	0.308	0.087	0.257	0.262	0.751	3	1.000

Simulation Run 9

1	0.624	0.150	0.000	0.150	0.000	0.150	1	0.774
2	0.720	0.163	0.000	0.150	0.000	0.150	1	0.883
3	0.128	0.173	0.000	0.150	0.000	0.150	1	0.302
4	0.103	0.177	0.000	0.150	0.000	0.150	1	0.280
5	0.222	0.181	0.000	0.150	0.000	0.150	1	0.403
6	0.376	0.186	0.000	0.150	0.000	0.150	1	0.562
7	0.573	0.194	0.000	0.150	0.000	0.150	1	0.767
8	0.540	0.206	0.000	0.150	0.000	0.150	1	0.746
9	0.025	0.219	0.000	0.150	0.000	0.150	1	0.244
10	0.282	0.223	0.000	0.150	0.000	0.150	1	0.505
11	0.207	0.231	0.000	0.150	0.000	0.150	1	0.438
12	0.237	0.237	0.000	0.150	0.000	0.150	1	0.474
13	0.276	0.243	0.277	0.150	0.000	0.150	1	0.520
14	0.262	0.251	0.126	0.150	0.000	0.150	1	0.514
15	0.018	0.260	0.657	0.150	0.000	0.150	2	0.807
16	0.178	0.260	0.335	0.166	0.000	0.150	2	0.501
17	0.246	0.260	0.256	0.177	0.000	0.150	1	0.506
18	0.208	0.268	0.056	0.177	0.000	0.150	1	0.476
19	0.133	0.275	0.345	0.177	0.000	0.150	2	0.522
20	0.287	0.275	0.290	0.187	0.000	0.150	1	0.562
21	0.340	0.282	0.269	0.187	0.000	0.150	1	0.622
22	0.192	0.293	0.518	0.187	0.000	0.150	2	0.705
23	0.042	0.293	0.531	0.204	0.000	0.150	2	0.735
24	0.043	0.293	0.103	0.221	0.000	0.150	1	0.336
25	0.077	0.297	0.288	0.221	0.065	0.150	2	0.510
26	0.188	0.297	0.341	0.232	0.686	0.150	3	0.836
27	0.273	0.297	0.138	0.232	0.074	0.183	1	0.570
28	0.081	0.305	0.178	0.232	0.350	0.183	3	0.533
29	0.131	0.305	0.371	0.232	0.613	0.199	3	0.811
30	0.043	0.305	0.396	0.232	0.596	0.232	3	0.828
31	0.033	0.305	0.310	0.232	0.251	0.267	2	0.542
32	0.152	0.305	0.082	0.243	0.546	0.267	3	0.813
33	0.004	0.305	0.185	0.243	0.258	0.300	3	0.558
34	0.092	0.305	0.299	0.243	0.131	0.315	2	0.542
35	0.097	0.305	0.007	0.254	0.243	0.315	3	0.558
36	0.104	0.305	0.074	0.254	0.203	0.331	3	0.534
37	0.148	0.305	0.281	0.254	0.127	0.347	2	0.535
38	0.032	0.305	0.209	0.265	0.260	0.347	3	0.607
39	0.015	0.305	0.083	0.265	0.030	0.365	3	0.395
40	0.052	0.305	0.076	0.265	0.050	0.372	3	0.422
41	0.118	0.305	0.014	0.265	0.246	0.381	3	0.627
42	0.109	0.305	0.155	0.265	0.322	0.408	3	0.729
43	0.141	0.305	0.088	0.265	0.277	0.437	3	0.714
44	0.011	0.305	0.005	0.265	0.161	0.465	3	0.626
45	0.129	0.305	0.051	0.265	0.114	0.491	3	0.606
46	0.000	0.305	0.030	0.265	0.238	0.508	3	0.746
47	0.055	0.305	0.089	0.265	0.274	0.539	3	0.813
48	0.036	0.305	0.024	0.265	0.013	0.573	3	0.586
49	0.042	0.305	0.072	0.265	0.043	0.589	3	0.632
50	0.108	0.305	0.079	0.265	0.261	0.615	3	0.875
51	0.069	0.305	0.163	0.265	0.117	0.647	3	0.764
52	0.028	0.305	0.072	0.265	0.109	0.677	3	0.787
53	0.001	0.305	0.037	0.265	0.175	0.707	3	0.882
54	0.085	0.305	0.037	0.265	0.064	0.740	3	0.804
55	0.082	0.305	0.113	0.265	0.154	0.772	3	0.926
56	0.081	0.305	0.111	0.265	0.207	0.808	3	1.000

Simulation Run 10

1	0.271	0.150	0.000	0.150	0.000	0.150	1	0.421
2	0.745	0.156	0.000	0.150	0.000	0.150	1	0.900
3	0.474	0.166	0.000	0.150	0.000	0.150	1	0.640
4	0.544	0.179	0.000	0.150	0.000	0.150	1	0.723
5	0.202	0.191	0.000	0.150	0.000	0.150	1	0.393
6	0.393	0.196	0.000	0.150	0.000	0.150	1	0.589
7	0.031	0.205	0.000	0.150	0.000	0.150	1	0.236
8	0.437	0.208	0.000	0.150	0.000	0.150	1	0.644
9	0.492	0.221	0.000	0.150	0.000	0.150	1	0.713
10	0.312	0.234	0.000	0.150	0.000	0.150	1	0.546
11	0.517	0.242	0.000	0.150	0.000	0.150	1	0.760
12	0.456	0.255	0.000	0.150	0.000	0.150	1	0.711
13	0.134	0.268	0.082	0.150	0.000	0.150	1	0.402
14	0.181	0.273	0.753	0.150	0.000	0.150	2	0.903
15	0.386	0.273	0.721	0.160	0.000	0.150	2	0.881
16	0.276	0.273	0.582	0.170	0.000	0.150	2	0.752
17	0.106	0.273	0.440	0.188	0.000	0.150	2	0.628
18	0.341	0.273	0.308	0.205	0.000	0.150	1	0.614
19	0.106	0.284	0.434	0.205	0.000	0.150	2	0.639
20	0.367	0.284	0.027	0.223	0.000	0.150	1	0.651
21	0.188	0.297	0.477	0.223	0.000	0.150	2	0.700
22	0.288	0.297	0.394	0.241	0.000	0.150	2	0.635

23	0.150	0.297	0.422	0.259	0.000	0.150	2	0.681
24	0.175	0.297	0.341	0.277	0.000	0.150	2	0.618
25	0.042	0.297	0.208	0.291	0.145	0.150	2	0.499
26	0.008	0.297	0.051	0.298	0.644	0.150	3	0.794
27	0.034	0.297	0.405	0.298	0.045	0.180	2	0.704
28	0.115	0.297	0.369	0.316	0.352	0.180	2	0.685
29	0.035	0.297	0.140	0.334	0.117	0.180	2	0.475
30	0.185	0.297	0.124	0.342	0.650	0.180	3	0.830
31	0.056	0.297	0.165	0.342	0.126	0.211	2	0.506
32	0.106	0.297	0.269	0.352	0.412	0.211	3	0.623
33	0.008	0.297	0.250	0.352	0.245	0.234	2	0.602
34	0.039	0.297	0.208	0.362	0.428	0.234	3	0.662
35	0.147	0.297	0.135	0.362	0.289	0.263	3	0.552
36	0.084	0.297	0.102	0.362	0.231	0.280	3	0.511
37	0.139	0.297	0.060	0.362	0.186	0.297	3	0.482
38	0.161	0.297	0.132	0.362	0.035	0.309	2	0.494
39	0.106	0.297	0.080	0.370	0.217	0.309	3	0.526
40	0.044	0.297	0.190	0.370	0.166	0.325	2	0.560
41	0.053	0.297	0.021	0.378	0.022	0.325	2	0.399
42	0.011	0.297	0.205	0.384	0.032	0.325	2	0.589
43	0.093	0.297	0.032	0.393	0.228	0.325	3	0.553
44	0.136	0.297	0.017	0.393	0.114	0.342	3	0.456
45	0.033	0.297	0.050	0.393	0.090	0.355	3	0.445
46	0.044	0.297	0.174	0.393	0.239	0.365	3	0.605
47	0.112	0.297	0.059	0.393	0.325	0.382	3	0.707
48	0.069	0.297	0.110	0.393	0.204	0.412	3	0.616
49	0.062	0.297	0.143	0.393	0.295	0.433	3	0.728
50	0.076	0.297	0.020	0.393	0.268	0.462	3	0.730
51	0.069	0.297	0.113	0.393	0.056	0.492	3	0.548
52	0.044	0.297	0.108	0.393	0.264	0.508	3	0.772
53	0.062	0.297	0.079	0.393	0.021	0.540	3	0.561
54	0.033	0.297	0.098	0.393	0.146	0.556	3	0.702
55	0.043	0.297	0.107	0.393	0.073	0.585	3	0.657
56	0.034	0.297	0.108	0.393	0.057	0.613	3	0.670
57	0.000	0.297	0.085	0.393	0.146	0.641	3	0.787
58	0.064	0.297	0.058	0.393	0.171	0.670	3	0.840
59	0.022	0.297	0.076	0.393	0.159	0.700	3	0.859
60	0.015	0.297	0.006	0.393	0.081	0.732	3	0.812
61	0.035	0.297	0.031	0.393	0.067	0.765	3	0.832
62	0.028	0.297	0.062	0.393	0.132	0.797	3	0.928
63	0.021	0.297	0.005	0.393	0.116	0.834	3	0.950
64	0.034	0.297	0.093	0.393	0.033	0.873	3	0.906
65	0.049	0.297	0.091	0.393	0.129	0.907	3	1.000

Simulation Run 11

1	0.374	0.150	0.000	0.150	0.000	0.150	1	0.524
2	0.719	0.158	0.000	0.150	0.000	0.150	1	0.878
3	0.152	0.169	0.000	0.150	0.000	0.150	1	0.321
4	0.225	0.173	0.000	0.150	0.000	0.150	1	0.398
5	0.277	0.178	0.000	0.150	0.000	0.150	1	0.455
6	0.440	0.184	0.000	0.150	0.000	0.150	1	0.624
7	0.001	0.196	0.000	0.150	0.000	0.150	1	0.198
8	0.500	0.196	0.000	0.150	0.000	0.150	1	0.696
9	0.236	0.209	0.000	0.150	0.000	0.150	1	0.445
10	0.209	0.215	0.000	0.150	0.000	0.150	1	0.424
11	0.207	0.221	0.000	0.150	0.000	0.150	1	0.427
12	0.410	0.227	0.000	0.150	0.000	0.150	1	0.636
13	0.028	0.239	0.438	0.150	0.000	0.150	2	0.588
14	0.340	0.239	0.709	0.159	0.000	0.150	2	0.868
15	0.306	0.239	0.467	0.169	0.000	0.150	2	0.636
16	0.291	0.239	0.658	0.187	0.000	0.150	2	0.845
17	0.088	0.239	0.659	0.197	0.000	0.150	2	0.856
18	0.150	0.239	0.009	0.207	0.000	0.150	1	0.389
19	0.183	0.244	0.571	0.207	0.000	0.150	2	0.778
20	0.276	0.244	0.458	0.223	0.000	0.150	2	0.681
21	0.095	0.244	0.101	0.242	0.000	0.150	2	0.342
22	0.163	0.244	0.198	0.246	0.000	0.150	2	0.444
23	0.198	0.244	0.340	0.253	0.000	0.150	2	0.593
24	0.145	0.244	0.416	0.262	0.000	0.150	2	0.679
25	0.123	0.244	0.075	0.280	0.684	0.150	3	0.834
26	0.237	0.244	0.064	0.280	0.025	0.182	1	0.482
27	0.017	0.251	0.416	0.280	0.592	0.182	3	0.773
28	0.013	0.251	0.355	0.280	0.259	0.213	2	0.635
29	0.198	0.251	0.341	0.298	0.342	0.213	2	0.640
30	0.160	0.251	0.062	0.316	0.533	0.213	3	0.745
31	0.139	0.251	0.307	0.316	0.257	0.243	2	0.623
32	0.116	0.251	0.318	0.334	0.092	0.243	2	0.652
33	0.202	0.251	0.277	0.352	0.142	0.243	2	0.630
34	0.150	0.251	0.246	0.370	0.234	0.243	2	0.616
35	0.032	0.251	0.194	0.384	0.505	0.243	3	0.749
36	0.024	0.251	0.294	0.384	0.292	0.275	2	0.678
37	0.104	0.251	0.226	0.402	0.114	0.275	2	0.628
38	0.097	0.251	0.213	0.419	0.137	0.275	2	0.632
39	0.147	0.251	0.226	0.437	0.125	0.275	2	0.663
40	0.038	0.251	0.040	0.456	0.356	0.275	3	0.631
41	0.124	0.251	0.218	0.456	0.041	0.300	2	0.674
42	0.073	0.251	0.208	0.474	0.051	0.300	2	0.682
43	0.070	0.251	0.165	0.492	0.268	0.300	2	0.657
44	0.095	0.251	0.069	0.511	0.031	0.300	2	0.580
45	0.001	0.251	0.138	0.520	0.166	0.300	2	0.658
46	0.071	0.251	0.023	0.539	0.190	0.300	2	0.562
47	0.042	0.251	0.179	0.548	0.043	0.300	2	0.727
48	0.074	0.251	0.002	0.564	0.254	0.300	2	0.566
49	0.072	0.251	0.011	0.573	0.217	0.300	2	0.584
50	0.070	0.251	0.012	0.582	0.236	0.300	2	0.594
51	0.063	0.251	0.053	0.592	0.167	0.300	2	0.645
52	0.094	0.251	0.152	0.610	0.144	0.300	2	0.762

53	0.078	0.251	0.038	0.626	0.028	0.300	2	0.664
54	0.004	0.251	0.095	0.645	0.200	0.300	2	0.740
55	0.060	0.251	0.100	0.663	0.015	0.300	2	0.763
56	0.013	0.251	0.057	0.679	0.059	0.300	2	0.736
57	0.014	0.251	0.029	0.696	0.206	0.300	2	0.725
58	0.071	0.251	0.085	0.712	0.064	0.300	2	0.797
59	0.069	0.251	0.114	0.729	0.168	0.300	2	0.843
60	0.005	0.251	0.001	0.739	0.165	0.300	2	0.740
61	0.064	0.251	0.063	0.756	0.001	0.300	2	0.819
62	0.041	0.251	0.078	0.773	0.090	0.300	2	0.851
63	0.013	0.251	0.065	0.782	0.127	0.300	2	0.848
64	0.056	0.251	0.025	0.792	0.114	0.300	2	0.817
65	0.025	0.251	0.005	0.808	0.112	0.300	2	0.813
66	0.034	0.251	0.071	0.825	0.132	0.300	2	0.895
67	0.033	0.251	0.088	0.835	0.018	0.300	2	0.923
68	0.010	0.251	0.009	0.845	0.113	0.300	2	0.854
69	0.012	0.251	0.022	0.855	0.063	0.300	2	0.876
70	0.034	0.251	0.031	0.865	0.051	0.300	2	0.896
71	0.038	0.251	0.043	0.875	0.100	0.300	2	0.917
72	0.012	0.251	0.035	0.885	0.096	0.300	2	0.919
73	0.000	0.251	0.022	0.895	0.058	0.300	2	0.917
74	0.020	0.251	0.022	0.905	0.006	0.300	2	0.927
75	0.015	0.251	0.049	0.915	0.052	0.300	2	0.965
76	0.023	0.251	0.016	0.926	0.097	0.300	2	0.941
77	0.008	0.251	0.033	0.936	0.067	0.300	2	0.969
78	0.022	0.251	0.039	0.946	0.087	0.300	2	0.985
79	0.021	0.251	0.050	0.957	0.010	0.300	2	1.000

APPENDIX B

SIMULATIONS FOR THE COMPONENTS NEURAL NETWORK

COMPUTER SIMULATION RUN 1

1	1	0.035	41	1	0.080	
2	2	0.033	42	1	0.092	
3	3	0.027	43	1	0.093	
4	2	0.024	44	1	0.087	
5	3	0.037				
6	3	0.037	45	1	0.103	discovered
7	2	0.039	46	2	0.102	discovered
8	3	0.027	47	1	0.118	
9	1	0.041	48	2	0.109	
10	3	0.033	49	1	0.127	
11	2	0.037	50	1	0.121	
12	2	0.038	51	1	0.125	
13	3	0.027	52	1	0.140	
14	1	0.044	53	1	0.136	
15	1	0.047	54	1	0.148	
16	2	0.042	55	1	0.143	
17	2	0.040	56	1	0.157	
18	1	0.043	57	1	0.172	
19	3	0.042	58	2	0.181	
20	3	0.043	59	2	0.187	
21	2	0.045	60	3	0.192	discovered
22	1	0.053	61	3	0.210	
23	2	0.050	62	3	0.229	
24	1	0.059	63	2	0.252	
25	1	0.054	64	2	0.271	
26	1	0.061	65	3	0.297	
27	1	0.055	66	3	0.325	
28	1	0.059	67	2	0.341	
29	1	0.049	68	3	0.375	
30	1	0.054	69	3	0.412	
31	1	0.058	70	3	0.433	
32	1	0.063	71	3	0.487	
33	1	0.072	72	3	0.529	
34	1	0.074	73	3	0.569	
35	2	0.062	74	3	0.641	
36	1	0.078	75	3	0.735	
37	1	0.084	76	3	0.818	
38	1	0.070	77	3	0.949	
39	1	0.070	78	3	1.000	
40	1	0.079				

COMPUTER SIMULATION RUN 2

1	1	0.034	5	1	0.040
2	1	0.040	6	3	0.037
3	1	0.031	7	3	0.037
4	3	0.031	8	3	0.040

9	1	0.040		45	1	0.088	
10	2	0.038					
11	1	0.041		46	1	0.102	discovered
12	3	0.035		47	2	0.102	discovered
13	3	0.042		48	1	0.117	
14	3	0.041		49	1	0.119	
15	1	0.028		50	2	0.123	
16	1	0.036		51	2	0.131	
17	2	0.046		52	1	0.133	
18	1	0.035		53	2	0.140	
19	2	0.043		54	3	0.139	discovered
20	1	0.041		55	2	0.159	
21	1	0.045		56	1	0.160	
22	2	0.050		57	3	0.166	
23	1	0.046		58	3	0.175	
24	3	0.043		59	1	0.190	
25	2	0.052		60	2	0.198	
26	1	0.051		61	2	0.223	
27	3	0.052		62	3	0.231	
28	1	0.048		63	3	0.248	
29	1	0.054		64	3	0.264	
30	3	0.055		65	2	0.290	
31	1	0.051		66	3	0.318	
32	1	0.067		67	2	0.336	
33	2	0.052		68	3	0.366	
34	2	0.063		69	3	0.405	
35	2	0.066		70	3	0.447	
36	1	0.067		71	3	0.482	
37	1	0.074		72	3	0.520	
38	1	0.078		73	3	0.565	
39	1	0.084		74	3	0.625	
40	1	0.083		75	3	0.726	
41	1	0.083		76	3	0.812	
42	1	0.077		77	3	0.911	
43	1	0.093		78	3	1.000	
44	2	0.088					

COMPUTER SIMULATION RUN 3

1	1	0.026		24	2	0.049	
2	2	0.030		25	1	0.045	
3	1	0.029		26	1	0.059	
4	2	0.032		27	1	0.054	
5	2	0.040		28	1	0.048	
6	1	0.036		29	2	0.056	
7	2	0.034		30	1	0.068	
8	1	0.034		31	1	0.064	
9	3	0.032		32	2	0.064	
10	2	0.040		33	1	0.074	
11	2	0.039		34	1	0.076	
12	3	0.041		35	1	0.076	
13	3	0.034		36	1	0.071	
14	2	0.042		37	1	0.076	
15	2	0.042		38	2	0.072	
16	3	0.029		39	1	0.084	
17	1	0.043		40	1	0.085	
18	1	0.038		41	1	0.089	
19	3	0.040		42	1	0.078	
20	2	0.049		43	1	0.098	
21	2	0.044					
22	1	0.054		44	1	0.102	discovered
23	2	0.050		45	1	0.097	

46	2	0.103	discovered
47	1	0.114	
48	1	0.105	
49	1	0.125	
50	1	0.131	
51	2	0.131	
52	1	0.139	
53	1	0.149	
54	1	0.149	
55	1	0.155	
56	1	0.160	
57	1	0.163	
58	2	0.175	
59	1	0.191	
60	3	0.190	discovered
61	3	0.211	
62	2	0.234	
63	2	0.240	
64	2	0.266	
65	3	0.288	
66	3	0.318	
67	3	0.341	
68	3	0.362	
69	3	0.411	
70	3	0.443	
71	3	0.465	
72	3	0.507	
73	3	0.561	
74	3	0.635	
75	3	0.699	
76	3	0.789	
77	3	0.915	
78	3	1.000	

COMPUTER SIMULATION RUN 4

1	1	0.019	
2	3	0.039	
3	2	0.037	
4	3	0.040	
5	2	0.038	
6	3	0.025	
7	3	0.034	
8	1	0.039	
9	2	0.037	
10	1	0.039	
11	3	0.036	
12	1	0.043	
13	2	0.043	
14	2	0.028	
15	1	0.037	
16	1	0.032	
17	3	0.028	
18	3	0.045	
19	3	0.038	
20	1	0.039	
21	3	0.043	
22	1	0.055	
23	3	0.044	
24	3	0.044	
25	1	0.047	
26	2	0.051	
27	1	0.048	
28	1	0.062	
29	1	0.067	
30	1	0.054	
31	1	0.065	
32	1	0.059	
33	2	0.066	
34	2	0.057	
35	3	0.063	
36	2	0.067	
37	1	0.085	
38	1	0.075	
39	1	0.083	
40	1	0.078	
41	1	0.089	
42	2	0.090	
43	3	0.084	
44	2	0.097	
45	1	0.100	discovered
46	1	0.105	
47	1	0.111	
48	1	0.123	
49	2	0.111	discovered
50	1	0.118	
51	1	0.136	
52	1	0.142	
53	1	0.145	
54	2	0.142	
55	1	0.155	
56	1	0.166	
57	2	0.166	
58	2	0.175	
59	1	0.190	
60	2	0.203	
61	2	0.220	
62	2	0.226	
63	2	0.231	
64	2	0.260	
65	2	0.281	
66	2	0.316	
67	3	0.332	discovered
68	3	0.354	
69	3	0.400	
70	3	0.416	
71	3	0.455	
72	3	0.500	
73	3	0.536	
74	3	0.602	
75	3	0.666	
76	3	0.753	
77	3	0.848	
78	3	0.998	

COMPUTER SIMULATION RUN 5

1	3	0.030	41	1	0.086	
2	1	0.030	42	1	0.095	
3	1	0.039	43	1	0.097	
4	1	0.032				
5	3	0.038	44	1	0.101	discovered
6	1	0.033	45	1	0.096	not used
7	2	0.031	46	2	0.105	discovered
8	1	0.040	47	1	0.116	
9	1	0.041	48	2	0.112	
10	1	0.038	49	1	0.117	
11	2	0.032	50	1	0.127	
12	2	0.040	51	1	0.123	
13	2	0.039	52	1	0.143	
14	2	0.040	53	1	0.142	
15	2	0.039	54	2	0.144	
16	1	0.046	55	1	0.158	
17	2	0.039	56	1	0.163	
18	2	0.037	57	2	0.164	
19	3	0.038	58	2	0.174	
20	1	0.042	59	2	0.193	
21	3	0.047	60	1	0.203	
22	3	0.037	61	2	0.216	
23	1	0.055	62	3	0.227	discovered
24	3	0.043	63	2	0.244	
25	1	0.059	64	2	0.273	
26	1	0.041	65	2	0.289	
27	3	0.052	66	2	0.317	
28	1	0.040	67	2	0.335	
29	2	0.059	68	2	0.350	
30	1	0.065	69	2	0.389	
31	2	0.058	70	3	0.417	
32	3	0.059	71	3	0.461	
33	1	0.070	72	3	0.511	
34	1	0.064	73	3	0.544	
35	1	0.070	74	3	0.609	
36	1	0.061	75	3	0.669	
37	2	0.071	76	3	0.737	
38	1	0.074	77	3	0.829	
39	1	0.085	78	3	0.981	
40	1	0.091				

COMPUTER SIMULATION RUN 6

1	2	0.029	17	2	0.032
2	1	0.040	18	3	0.037
3	3	0.040	19	2	0.043
4	1	0.031	20	2	0.039
5	1	0.039	21	2	0.048
6	1	0.041	22	1	0.042
7	2	0.038	23	1	0.045
8	2	0.029	24	1	0.055
9	2	0.038	25	2	0.036
10	1	0.035	26	1	0.051
11	1	0.029	27	1	0.057
12	2	0.041	28	1	0.064
13	1	0.032	29	1	0.061
14	1	0.044	30	1	0.047
15	2	0.045	31	2	0.062
16	1	0.048	32	2	0.058

33	1	0.068	
34	1	0.068	
35	1	0.068	
36	2	0.073	
37	1	0.061	
38	1	0.074	
39	1	0.084	
40	1	0.080	
41	1	0.078	
42	1	0.089	
43	1	0.098	
44	1	0.092	
45	3	0.090	
46	2	0.104	discovered
47	2	0.096	not used
48	1	0.101	discovered
49	1	0.122	
50	2	0.121	
51	1	0.129	
52	1	0.127	
53	1	0.134	
54	1	0.152	
55	1	0.149	
56	1	0.161	
57	2	0.164	
58	2	0.179	
59	1	0.182	
60	1	0.201	
61	2	0.207	
62	2	0.224	
63	2	0.240	
64	2	0.261	
65	2	0.278	
66	2	0.311	
67	2	0.338	
68	3	0.351	discovered
69	3	0.385	
70	3	0.411	
71	3	0.460	
72	3	0.490	
73	3	0.545	
74	3	0.594	
75	3	0.665	
76	3	0.758	
77	3	0.859	
78	3	0.995	

COMPUTER SIMULATION RUN 7

1	3	0.019	
2	1	0.039	
3	1	0.038	
4	2	0.030	
5	2	0.030	
6	2	0.039	
7	1	0.039	
8	3	0.040	
9	3	0.038	
10	2	0.030	
11	1	0.036	
12	3	0.036	
13	2	0.036	
14	2	0.039	
15	1	0.041	
16	1	0.047	
17	1	0.031	
18	3	0.040	
19	1	0.041	
20	1	0.032	
21	3	0.046	
22	1	0.033	
23	1	0.048	
24	2	0.052	
25	3	0.045	
26	1	0.054	
27	1	0.060	
28	1	0.050	
29	1	0.049	
30	1	0.061	
31	1	0.064	
32	1	0.069	
33	1	0.067	
34	1	0.070	
35	1	0.069	
36	2	0.063	
37	1	0.066	
38	1	0.073	
39	2	0.070	
40	3	0.076	
41	1	0.088	
42	1	0.080	
43	1	0.093	
44	1	0.095	
45	1	0.107	discovered
46	1	0.095	
47	2	0.105	discovered
48	2	0.113	
49	1	0.118	
50	1	0.127	
51	1	0.123	
52	2	0.137	
53	2	0.140	
54	2	0.136	
55	1	0.162	
56	1	0.169	
57	2	0.163	
58	2	0.174	
59	2	0.196	
60	2	0.202	
61	2	0.205	
62	2	0.220	
63	3	0.240	discovered
64	3	0.272	
65	2	0.299	
66	2	0.319	
67	3	0.340	
68	3	0.374	
69	3	0.392	

70	3	0.436
71	3	0.486
72	3	0.529
73	3	0.569
74	3	0.632
75	3	0.709
76	3	0.794
77	3	0.884
78	3	1.000

COMPUTER SIMULATION RUN 8

1	1	0.032	
2	1	0.029	
3	3	0.035	
4	1	0.033	
5	3	0.039	
6	3	0.040	
7	2	0.036	
8	2	0.037	
9	1	0.028	
10	2	0.041	
11	3	0.038	
12	3	0.039	
13	3	0.041	
14	2	0.044	
15	1	0.040	
16	2	0.044	
17	1	0.042	
18	1	0.047	
19	1	0.046	
20	3	0.045	
21	2	0.045	
22	2	0.034	
23	2	0.050	
24	1	0.035	
25	1	0.036	
26	1	0.055	
27	1	0.052	
28	2	0.046	
29	1	0.058	
30	1	0.062	
31	1	0.067	
32	1	0.073	
33	1	0.065	
34	1	0.074	
35	2	0.066	
36	1	0.078	
37	1	0.071	
38	2	0.069	
39	1	0.083	
40	1	0.074	
41	2	0.080	
42	1	0.091	
43	1	0.102	discovered
44	2	0.095	no strategy use
45	2	0.099	no strategy use
46	1	0.102	
47	1	0.111	
48	1	0.124	
49	1	0.116	
50	2	0.124	discovered
51	2	0.128	
52	1	0.133	
53	1	0.141	
54	1	0.147	
55	1	0.161	
56	2	0.166	
57	2	0.155	
58	1	0.185	
59	2	0.186	
60	1	0.198	
61	2	0.203	
62	2	0.226	
63	2	0.242	
64	2	0.265	
65	2	0.289	
66	2	0.299	
67	2	0.325	
68	2	0.358	
69	2	0.392	
70	3	0.412	discovered
71	3	0.446	
72	3	0.497	
73	3	0.538	
74	3	0.597	
75	3	0.651	
76	3	0.734	
77	3	0.826	
78	3	0.940	

COMPUTER SIMULATION RUN 9

1	2	0.030
2	2	0.038
3	3	0.038
4	1	0.040
5	1	0.039
6	2	0.039
7	3	0.031
8	3	0.033
9	3	0.040
10	1	0.038
11	1	0.030
12	2	0.029
13	1	0.041
14	2	0.043
15	2	0.034
16	2	0.041
17	3	0.041
18	1	0.044
19	1	0.048
20	1	0.048
21	1	0.046
22	1	0.053

23	1	0.054	
24	2	0.050	
25	3	0.049	
26	1	0.048	
27	1	0.057	
28	2	0.056	
29	1	0.060	
30	1	0.052	
31	1	0.071	
32	1	0.055	
33	1	0.071	
34	1	0.056	
35	1	0.069	
36	2	0.065	
37	1	0.085	
38	1	0.075	
39	1	0.082	
40	1	0.082	
41	1	0.089	
42	1	0.083	
43	2	0.084	
44	1	0.102	discovered
45	1	0.099	not used
46	1	0.109	
47	1	0.098	not used
48	1	0.114	
49	1	0.117	
50	1	0.121	
51	1	0.118	
52	2	0.135	discovered
53	1	0.148	
54	2	0.149	
55	1	0.140	
56	1	0.166	
57	1	0.162	
58	1	0.168	
59	3	0.180	discovered
60	1	0.191	
61	3	0.204	
62	2	0.227	
63	3	0.241	
64	3	0.252	
65	2	0.286	
66	2	0.310	
67	3	0.331	
68	3	0.360	
69	3	0.412	
70	3	0.419	
71	3	0.457	
72	3	0.513	
73	3	0.569	
74	3	0.609	
75	3	0.693	
76	3	0.775	
77	3	0.882	
78	3	1.000	

COMPUTER SIMULATION RUN 10

1	2	0.039	
2	2	0.038	
3	3	0.040	
4	1	0.040	
5	1	0.026	
6	3	0.038	
7	3	0.028	
8	3	0.029	
9	1	0.029	
10	2	0.035	
11	1	0.039	
12	1	0.035	
13	2	0.034	
14	1	0.033	
15	2	0.040	
16	1	0.038	
17	1	0.045	
18	2	0.038	
19	1	0.040	
20	2	0.032	
21	1	0.054	
22	1	0.044	
23	2	0.043	
24	1	0.056	
25	2	0.046	
26	1	0.049	
27	1	0.055	
28	1	0.047	
29	1	0.057	
30	1	0.052	
31	1	0.057	
32	1	0.064	
33	2	0.066	
34	1	0.065	
35	2	0.061	
36	2	0.064	
37	2	0.070	
38	1	0.070	
39	1	0.068	
40	1	0.088	
41	1	0.084	
42	1	0.091	
43	1	0.098	
44	1	0.085	
45	2	0.092	
46	1	0.106	discovered
47	1	0.097	not used
48	1	0.122	
49	1	0.115	
50	2	0.117	discovered
51	1	0.137	
52	1	0.122	
53	1	0.139	
54	1	0.149	
55	1	0.142	
56	1	0.152	
57	3	0.164	discovered
58	1	0.175	
59	2	0.194	

60	2	0.200
61	2	0.212
62	3	0.231
63	2	0.256
64	2	0.268
65	3	0.280
66	3	0.315
67	3	0.330
68	3	0.360
69	3	0.419
70	3	0.449
71	3	0.471
72	3	0.517
73	3	0.574
74	3	0.638
75	3	0.720
76	3	0.790
77	3	0.921
78	3	1.000

GRADUATE SCHOOL
UNIVERSITY OF ALABAMA AT BIRMINGHAM
DISSERTATION APPROVAL FORM

Name of Candidate VIVEK ANUMOLU

Major Subject Computer and Information Sciences

Title of Dissertation "A Hybrid Neural Network Methodology for Studying the Development of External Memory Strategies in Problem-Solving."

Dissertation Committee:

Kevin D. Reilly, Chairman

Warren D. Jones

Director of Graduate Program Warren D. Jones

Dean, UAB Graduate School W. A. Bibb

Date