CONNECTIONIST MODELS OF CHOICE AND REACTION TIME IN PSYCHOPHYSICS AND WORD RECOGNITION.

YVES LACOUTURE

Department of Psychology

McGill University, Montréal

May 1990

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

A connectionist architecture is developed that can be used for modeling choice probabilities and reaction times in psychophysics and word recognition. The network architecture consists of a feed-forward network and a decoding module. Learning is by mean-variance back-propagation, an extension of the standard back-propagation learning algorithm. The new learning procedure is interpreted as a selective attention mechanism, and leads to a better model of learning in simple identification tasks than the standard back-propagation. Choice probabilities are modeled by the input/output relations of the network, and reaction times are modeled by the time taken for the network, particularly the decoding module, to achieve a stable state. The model is applied to both unidimensional and multidimensional identification tasks in psychophysics and to word recognition.

RÉSUMÉ

Cette thèse propose une architecture connexioniste hybride constituée d'un réseau en couches et d'un module de décodage pour le modelage des choix et des latences dans des tâches d'identification simple et de reconnaissance de mots. De plus, une version modifiée (mean-variance back-propagation) de l'algorithme d'apprentissage par propagation rétroactive de l'erreur est proposée. Les résultats démontrent que ce nouvel algorithme, qui permet l'inclusion d'un mécanisme d'attention sélective, possède une meilleure adéquation que l'approche traditionelle pour modeler l'apprentissage cognitif dans des tâches d'identification. Il est démontré que le modèle proposé peut rendre compte de plusieurs phénomènes comportementaux.

STATEMENT OF ORIGINAL CONTRIBUTIONS

This thesis reports five distinct significant research contributions.

1. While response time is fundamental in the study of cognition, there is limited research demonstrating parallels between human reaction time and processing time in a connectionist network. I suggest a possible implementation in connectionist networks of time-dependent processes. While some researchers have used a goodness of fit indicator such as mean-square error as a predictor for latencies in such networks (Seidenberg & McClelland, 1989), others have proposed dynamic connectionist networks to model reaction time – for instance, Cohen, Dunbar and McClelland (1990) superimposed a cascade mechanism (McClelland, 1979) on each unit of a feed-forward net. I prefer a modular approach whereby distinct modules *map* and *decode* information in real time – specifically, I propose a hybrid architecture consisting of a feed-forward mapping network the output of which is fed into a dynamic decoding module.

2. As a specific application of this approach, a model of absolute identification is proposed. Not only do the results reported here show that a connectionist system can learn simple identification tasks, but they also demonstrate that under reasonable conditions the resulting learning curves and performance indicators match behavioral data.

3. I develop a new learning algorithm that is a modified version of the famous back-propagation (BP) learning algorithm. Back-propagation adaptatively changes the weights of the network to minimize a mean-square error criterion. The modified version that I propose and call *mean-variance back-propagation* (MV-BP) adaptively minimizes a weighted mixture of the mean and variance of the squared error. This new algorithm, which implements a *selective attention mechanism* in the learning process, gives better fits to behavioral data in cognitive learning than does the standard back-propagation algorithm.

4. In an attempt to model larger scale identification tasks a model of word recognition and naming is proposed. Although the scope of this model is limited, it does provide results matching some fundamental behavioral data. Effects of word regularity and word frequency on response time are replicated. A hybrid architecture similar to that used to model absolute identification demonstrates possible links between response time, probability of error and task difficulty in this domain.

5. The work presented here also makes significant technical contributions to the connectionist field through Zip_net, a neural network simulator. It is a fast, easy to use, feed-forward network simulator. This software is available at no charge from the author and is described in Appendix 1.

ACKNOWLEDGEMENTS

Je désire exprimer ma gratitude envers les personnes qui m'ont aidé au cours de mes travaux de doctorat. J'aimerais d'abord remercier les membres de mon comité d'étude; en premier lieu, mon directeur de thèse le Docteur A.A.J. (Tony) Marley, qui a démontré à mon endroit une disponibilité qui dépasse de beaucoup ce qu'il est convenu d'attendre de son directeur. Je le remercie pour sa patience, ses encouragements, sa compréhension et pour avoir su m'inculquer un peu de sa rigueur conceptuelle. Je remercie le Docteur Mark Seidenberg pour ses encouragements pertinents. J'exprime aussi ma gratitude envers le Docteur Jim Ramsay qui, entre autres choses, a fortement contribué à mon adaptation lors de mon arrivé à l'Université McGill. Je désire finalement remercier les nombreux amis qui m'ont aidé de diverses façons, surtout au cours des derniers mois.

J'adresse finalement mes remerciements au Conseil de Recherche en Sciences Naturelles et en Génie du Canada de même qu'à la fondation McConnell de l'Université McGill pour leur assistance financière.

DISCLAIMER

The reader is aware that English is not my first language. I apologize if the grammar and style of the text are sometimes lacking. I also want to again thank my advisor, Tony Marley, for his helpful comments on the style and content of the manuscript.

Yves Lacouture, May 1990.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This thesis studies various identification tasks from the connectionist perspective. Connectionism is an emerging paradigm in cognitive science, with connectionist models being a class of statistical models based on the assumption that cognitive processes can be represented using networks of simple interconnected processing elements (McClelland, Rumelhart & Hinton, 1986). The working principles behind connectionist models are not new, with connectionism forming an extension of frequently used statistical models, the main difference being that connectionist models are generally non-linear and involve a large number of parameters.

The connectionist approach contributes an innovative perspective to the modeling of cognitive processes. First, knowledge representation is *distributed* - i.e. information is represented through a *pattern of activations* distributed across units. Second, in contrast to symbolic information processing, the computational information (program) is not represented differently than the factual information (data), with both being encoded in the weighted links of the network.

This new conceptualization marks an important shift in cognitive science, which has previously been mostly dominated by symbolic information processing approaches. With the symbolic approach, specific chunks of knowledge (data) are

represented by arbitrary symbols independent of the computational information (program). With connectionism, the arbitrary symbolic representation is replaced by distributed patterns of activation while a set of weak processing constraints replaces the inference machine. This is a shift from a conceptualization in which the brain is seen as a finite state automaton to a new view where the processing depends on a large number of analogue devices. This new paradigm has engendered an immense enthusiasm and brought together researchers from various fields including psychology, physics, computer science and matnematics. The work reported here takes a definite psychological perspective. The goal is not so much to develop efficient ways to perform cognitive tasks, but rather to investigate the psychological plausibility of connectionist implementations.

Every connectionist model involves three sets of assumptions concerning *architecture*, *learning* and *stimulus representation*. Independent of the particular application, a careful study of these three aspects is needed to build a good connectionist model. Contrary to common belief, the apparent similarity between connectionist networks and the brain does not guarantee the adequacy of the connectionist approach for modeling behavioral data. For instance, one of the areas discussed in this thesis is simple identification tasks. It will be seen that a psychologically plausible

implementation is obtained only after careful study of the architecture, learning implementation and stimulus representation.

The major methodology used here is simulation "experiments". This is certainly the best way to study a complex self-adaptive artificial system. Each simulation is one out of a large set of possible realizations, which means that a connectionist network is not a general model that represents average characteristics of several realizations, but rather it is a model which attempts to mimic the behavior of a single individual. Thus, the adequacy of the model should be tested by comparing the output of a single simulation to the data of a single individual.

The simulation approach has several limitations, the main one being that it is very costly and time-consuming to evaluate chance effects. When a simulation is run, several pseudo-random processes are involved (such as the starting values of the connecting weights of the network) which affect the network behavior (Kolen & Pollack, 1990). It is always possible that the observed results depend on the specific pseudo-random sequences generated. Also, software bugs are always possible. A solution to these problems is to replicate all simulations several times and estimate average characteristics. This solution is impractical: a single simulation of the type reported in this thesis can easily

take several days to run on a SUN-3 computer (the machine that I used).

1.1. Scope and goals. The motivating factor behind the work reported here was the study of reaction time within a connectionist framework. To do this two exemplar tasks were chosen: a simple one, absolute identification, and a complex one, word recognition and naming. Most of the work reported here concerns absolute identification. The simplicity of this task allows a better understanding of the mechanisms involved, while the reasonable size of the networks needed for the simulations permits detailed study of the network parameters. The word recognition model presented here is intended to demonstrate that complex identification tasks can also be implemented within the proposed approach. Unfortunately, the complexity of the network involved limits the scope of its application – the word recognition simulation reported here represents ten (10) days processing time on a devoted SUN-3/160 computer.

1.2. Structure of the thesis. The text has six chapters. Following this introduction, Chapter 2 provides a review of important connectionist models. The matrix approach which embodies several fundamental characteristics of connectionism is described in some detail. For the same reason the feed-forward network and the back-propagation learning algorithm are described extensively. A study of the classical encoder

problem is provided which should give the reader a better feeling for the working principles of feed-forward networks.

Chapter 3 presents a review of data and non-connectionist models relevant to absolute identification. Basic behavioral phenomena and more traditional modeling approaches are described. The core of the thesis is Chapter 4, where a connectionist implementation of absolute identification is presented. Section 4.1 concerns learning and presents a modified version of the back-propagation learning algorithm that I call *mean-variance back-propagation* – it implements a selective attention mechanism that gives results which are descriptively more adequate than those obtained from standard back-propagation. Section 4.2 discusses a hybrid connectionist architecture that is useful for modeling reaction time – the structure consists of a mapping device (a feed-forward net) and a decoding module. Section 4.3 discusses the implementation of a Gaussian sensory trace model while Section 4.4 integrates the material of the previous three sections and presents the overall model of absolute identification using mean-variance back-propagation, the hybrid connectionist architecture and Gaussian sensory traces. Section 4.5 extends the results to two-dimensional stimuli.

Chapter 5 presents a model for the more complicated identification task of word recognition and naming. A hybrid architecture similar to the one used previously is described.

A general discussion follows in Chapter 6. Finally, Appendix 1 contains a description of the Zip_net neural network simulator that I have developed. The development of the simulator was an integral part of the work on this thesis as no available software had the features that I needed.

CHAPTER 2

A REVIEW OF THE MAJOR CONNECTIONIST MODELS

2.1. Early work

Modern connectionism can be traced to McCulloch and Pitts (1943), who first demonstrated that a network of neuron-like elements can perform elementary computations. This was the beginning of a new conceptualization in which cognition is seen as emerging from computational properties of simple processing elements. Hebb expanded the idea in *The Organization of Behavior* (1949) and suggested how learning might be implemented by cell assemblies. He assumed that knowledge is encoded in the strengths of the interconnections between neurons, and that learning involves changes of these weights. His law governing the weight changes, the now well-known *correlational hypothesis*, can be expressed simply: two neurons will tend to strengthen or develop a common link if their activity levels are correlated. From a neuro-anatomical point of view, Hebb's suggestion has never been proved or disproved in a definite manner since the micro-connections are difficult to study (see McNaughton and Morris, 1987, for a recent neuro-anatomical study of synaptic enhancement and learning). However, the computational plausibility of his suggestion has gained significant support over the years.

Edmonds and Minsky (see Minsky, 1954) implemented the first attempt to test the validity of Hebb's ideas. Unable to make

a formal mathematical demonstration and without computer resources, they built a partly mechanical, partly electronic machine, which, through a feedback process, could exhibit elementary learning. Their machine was never completely operational and although no firm conclusions were extracted from their work it captivated the imagination of researchers: here was a possible implementation of a self-regulating artificial learning device based on the same principle postulated by Hebb to explain neural adaptation.

After several years of work Rosenblatt (1959, 1962) proposed the first implementation of an adaptive network. He developed the *perceptron,* a device that could solve perceptual classification problems efficiently. Made of a set of n visible units connected through weighted links to a single integration unit, the device could exhibit learning through modification of its weighted links. The working principle of the device is simple. Each visible unit has a certain number of input lines organized on an array **R,** defining a perceptual field on which inputs are presented. The input lines of any visible unit cover a subregion or the whole input field and respond to a specific input arrangement on the array acting as a feature predicate. The activity of these visible units defines a vector

$$e = [e_1, e_2, \ldots, e_n]$$

where e_i is the activation of unit i. The integrator unit is usually a simple binary response element: its response is 1 (the unit "fires") if the summation of the input signals is greater than a certain threshold and 0 (the unit is "quiescent") otherwise.

When the system makes a classification error, the weights are changed to lower the probability that the same error will recur. This is implemented by the *perceptron learning algorithm.* Rosenblatt proved that for a stimulus set **F** which is linearly separable into two subsets $\mathbf{F}^-$ and $\mathbf{F}^+$ (i.e. **F** can be divided by a hyperplane) the learning algorithm guarantees that the classification will be learned in a finite number of learning trials (see Hinton & Anderson, 1981, for a discussion).

Initially, the perceptron learning theorem created considerable enthusiasm for "neural network" research. Rosenblatt (1959) even stated in a controversial paper that "... the perceptron establishes, beyond doubt, the feasibility and principle of a non-human system which may embody cognitive functions...". This assertion generated a large controversy which subsided ten years later with the publication of Minsky and Papert's (1969, 1988) *Perceptrons.* This book summarizes research carried out throughout the sixties by the authors, both extending Rosenblatt's work and clearly stating the limits of the perceptron. First, Minsky and Papert demonstrated that it is impossible to use the

perceptron learning algorithm in adaptative *multi-layered networks*. A multi-layer perceptron is a network where the output of one layer of perceptrons is the input to another layer of perceptrons. A single layer perceptron can only perform (binary) linear classification. It appears that most interesting cognitive processes involve n-valued, n>2, non-linear classifications. Minsky and Papert also emphasize the unrealistic psychological aspect of perceptron learning whereby the system learns only from errors; the learning algorithm has no provision for adaptive changes when the net gives the correct classification.

Minsky and Papert's book had a devastating effect on connectionism (see Pollack's, 1989, book review of the 1988 Revised Edition). Most researchers dropped connectionism and turned to the very efficient, and at the time extremely popular, new *information processing approach*, based on the manipulation of symbol structures, and most fully articulated by Newell and Simon (1963). Later work has shown that the research community overreacted; the perceptron can be improved. Almost twenty years after the Minsky and Papert book appeared, Rumelhart and Zipser (1985) proposed "competitive learning", a generalization of the perceptron learning algorithm for multi-layered networks. This work was expanded and reformulated by Rumelhart, Hinton and Williams (1986) and led to the *back-propagation learning algorithm*.

There are four reasons for the reemergence of connectionism in the late seventies. First, the development and availability of very powerful cheap computers allowed simulation, once an exotic and very expensive tool, to become a commonly used technique. Second, the accumulating data on the neuro-anatomical functioning of the brain reinforced the viewpoint that cognition is based on a massively parallel system. Third, with the development of large scale parallel computers a new software approach needed to be imagined. Finally, cognitive scientists were faced with the limitations of the serial rule-based models in domains such as vision and language processing.

While Hebb was developing the idea of cells assemblies, Gabor (1948, 1949) invented holography. This is a process used to encode the interference pattern of two beams of coherent monochromatic light in a translucent sensitive plate. When the two beams are refracted images of two objects, subsequent illumination of the sensitive plate with one of the original images reveals the other one, making the holograph an associative memory device.

Gabor (1968a) proposed in an optimistic paper that holography be used as a model of memory. First, he emphasized that holograms are resistant to local damage: if part of the sensitive plate is destroyed, the system still responds though the recall might be degraded. Second, Gabor pointed out that holograms can encode, on the same plate, different

associations; he claimed that recall could then be done with little interference between associations. Lastly, he remarked that holograms are bi-directional: either element of the associated pair can cue the recall of the other one. Using auto-associations, the hologram can thus act as a content-addressable memory, where partial input cues the reconstruction of whole patterns.

Much later, the concept of holographic memory was extended and generalized to *convolution memory* by Ratcliff & Murdock (1976), Murdock (1982) and Eich (1982). These extensions provided the ground work for better mathematical analysis, and allowed one to study and formalize the characteristics and limits of these memory models. I now summarize briefly the mathematical principles of holographic and convolution memory.

Let f(t) and g(t) be the continuous distributions over time (or space) of two different processes and let c(t) be the convolution of these processes i.e.

$$c(t) = f(t) * g(t)$$

where * represents the convolution operator. Correlation of the resulting convolution with one of the original patterns allows recall of the other since

$$f(t) = g(t) \text{ ' } c(t)$$

and

$$g(t) = f(t) \text{ ' } c(t).$$

where ' is the correlation operator. If the stimuli are discrete vectors of fixed length n, the association (convolution) between two vectors $a= (a_1,..,a_n)$ and $b= (b_1,..,b_n)$ is given by

$$c = a*b = \sum_{i=1}^{n} a_i b_i$$

(for futher details, see the discussion of the matrix model in the following section).

Although at first very appealing, simulation experiments carried out on convolution and holographic memory systems have showed that the recall process induces a high level of noise (see discussion by Hinton & Anderson, 1981). Furthermore, a detailed study of holographic memories by Longuet-Higgins (1968) demonstrated that the amount of information which can be superimposed is very limited and that the resistance to local damage is quite small. Pike (1984) reached similar conclusions about convolution models. He proved very clearly that matrix models are simpler, show better signal-noise ratios and are neuro-anatomically more realistic. Thus, I now turn to these matrix models.

2.2. Matrix models and the Brain-State-in-a-Box (BSB) Model

The matrix models were mainly developed by Anderson, Silverstein, Ritz & Jones (1977), Anderson (1983), Hinton & Anderson (1981) and Kohonen (1977). In its simplest version, a matrix model is the discrete equivalent of the convolution model where the analogue sensitive medium is replaced by a discrete n by n array. I will describe this approach in some detail since it is an important step in the evolution of connectionism and also because I will refer to matrix models later in the thesis.

The basic idea behind matrix models is to represent the activity of a network of n simple processing elements by an n-dimensional *state vector V,*

$$V = (v_1, \ldots v_n).^1$$

Each element of the state vector represents the activity of one of the units. The activity level is analogous to a firing rate. A square n by n matrix **A** represents the interconnections of the elements with a_{ik} being the strength of the weighted link from unit i to unit k. This matrix is

[1]On notation: V designates the state vector, v_i the element i of the state vector V and V[t] the state vector at time t.

assumed to be symmetrical ($a_{ik} = a_{ki}$) for mathematical simplicity, although this constraint can be relaxed. All units are assumed connected to all others although, again, this assumption can be weakened; in particular the main diagonal of A, which represents the connections of each unit to itself, can be arbitrarily set to zero.

A system "step" is expressed as

$$V[t+1] = \mathbf{A}V[t]$$

where V[t+1] is the state vector at time t+1 and V[t] represents the state vector at time t. The activation of any specific unit i is given by

$$v_i = \sum_{m=1}^{n} a_{im}\, v_m.$$

The system is assumed asynchronous (this means that the serial updating order of the activity is theoretically irrelevant) and transmission time across units is considered negligible. Making the unit non-linear by, say, the addition of a threshold mechanism such as

$$v_i = \max(0,\ v_i - h_i),$$

where h_i is the threshold of unit i, significantly reduces the total amount of noise in the network (Hinton & Anderson, 1981). The threshold mechanism can easily be implemented with

a *true unit* e_h with fixed activity which is connected to all other units of the network in an inhibitory manner (Hopfield, 1986). Thresholds can be changed through modification of the strength of connections with the true unit.

Anderson et al. (1977) extended the threshold non-linearity by introducing range limits that bound the activity of the units in the range [-1, 1]. The lower bound is the minimum activity, whereas the upper bound is the maximum activity and 0 the spontaneous (resting) level of activity. The range [-1, 1] is a convenient value that simplifies the computations since the correlation of the activities of any two units can then be estimated through simple multiplication.

Matrix models allow easy implementation of Hebb's correlational hypothesis of associative learning – namely, that the strength of the connection between two units changes proportionally to the correlation of their activity, i.e.

$$\Delta a_{ik} = (v_i * v_k)\alpha, \quad i,k = 1, \ldots, n,$$

where Δa_{ik} is the change in connectivity between unit i and k and α is a small nonnegative constant. In matrix notation we have

$$\Delta \mathbf{A} = (\mathbf{V}\mathbf{V}^t)\alpha$$

where V is the state vector, V^t its transpose, and $\Delta\mathbf{A}$ the change in the connectivity matrix. Now, assume that we have a set of n orthogonal non-negative stimulus (or input) vectors $f_1, \ldots, f_n$, each with norm $||f_i|| = 1$. Starting from zero weighted connections ($\mathbf{A} = 0$) let

$$\mathbf{A} = \sum_{j=1}^{n} k_j f_j f_j^t,$$

where k_j is the number of times pattern i has been presented.

Recall is done through post-multiplication of $\mathbf{A}$ with an input f_j, and we readily see that with one of the orthogonal stimuli

$$\mathbf{A}f_j = k_j f_j.$$

This equation is familiar to the reader as the characteristic equation of $\mathbf{A}$. The n independent input vectors are the eigenvectors of $\mathbf{A}$ with associated eigenvalues $k_1, \ldots k_n$. The more frequently an input is presented, the larger is its associated eigenvalue. This gives the model a useful property whereby the system responds more strongly to commonly presented patterns.

With real non-negative input vectors, all elements of $\mathbf{A}$ are non-negative. On the other hand, if the state vector V represents deviation from an average activity level such that f_j is made of positive and negative values, and if the assumption of orthogonality of the input vectors is relaxed,

then the matrix **A** would have an arbitrary positive semi-definite (all eigenvalues greater or equal to zero) structure. Let's redefine the input vectors such that for any f_j we have:

$$f_j = f_j' - \frac{\sum_{k=1}^{h} f_k'}{h}$$

where f_1', ..., f_h' are arbitrary n-dimensional vectors; thus each element of f_j is the deviation of the activity from the mean activity over all the inputs. The quadratic form

$$\Delta\mathbf{A} = f_j f_j^t$$

then represents an increase or a decrease of connectivity. Units with highly correlated activities will tend to build excitatory (positive) links while negatively correlated units will develop inhibitory (negative) connections.

The matrix

$$\mathbf{A} = \sum_{j=1}^{n} k_j f_j f_j^t,$$

is a scalar multiple of the covariance matrix $\mathbf{\Sigma}$ of the inputs, where

$$\mathbf{\Sigma} = \sum_{i=1}^{n} p_i f_i f_i^t - \left(\sum_{i=1}^{n} p_i f_i\right)^t \left(\sum_{i=1}^{n} p_i f_i\right)$$

with p_i the relative frequency of input vector f_i. Since the matrix **A** is positive semi-definite, we know that the eigenvectors of **A** form a basis set of the input space (Morrison, 1967) and thus each input vector f_k can be expressed as a linear combination of the eigenvectors of **A**, i.e.

$$f_k = \sum_{i=1}^{n} c_{ki} e_i$$

where e_i are the eigenvectors of **A**, and c_{ki} are constants corresponding to the contribution of e_i to the input pattern f_k. After presentation and learning of a set of input vectors, the presentation of any partial or deteriorated vector f will lead to a reconstructive process which is described below. Maximum response is achieved when the input vector is an eigenvector of the network. As suggested by Anderson et al. (1977), the eigenvectors are taken to represent the *distinctive features* of the inputs and represent the regularities in the input space. These authors also claim that such regularities are psychologically meaningful.

Such a matrix device can perfectly store a set of n orthogonal patterns where n is the number of (input) units in the network. With arbitrary patterns, the matrix model develops a set of orthogonal archetypes corresponding to the underlying regularities in the stimulus set. We will now

discuss a direct extension of the matrix model, the *Brain-State-in-a-Box* (BSB) model, first proposed by Anderson et al. (1977), which takes advantage of this characteristic and uses feedback to reconstruct a whole feature vector from partial input. I now review Anderson's BSB model.

Suppose we have a network with connection matrix **A** and input vector f. Assume that

$$V(t+1) = V(t)+\mathbf{A}V(t) \qquad (2.1)$$

$$= (\mathbf{I}+\mathbf{A})V(t)$$

where V(t+1) is the state vector at time t+1 and **I** the identity matrix. Following presentation of the vector V, the state vector will monotonically grow with iterative use of Equation 2.1 (Anderson et al., 1977; Golden, 1985). If the input vector is an eigenvector of **A**, the state vector will get longer without changing direction, whereas if the input vector is an arbitrary vector, the state vector will grow and change direction as a result of being "attracted" by the eigenvectors with large associated eigenvalues.

If we bound the activity of the units in a range [-C, +C], C>0, the limits of activation define a box in the n-dimensional space. The state vector grows when provided with an arbitrary input vector. At some point the vector will encounter a "wall" of the box, i.e. a unit has reached its

maximum activation. The vector will continue to grow, following the wall until it reaches another wall and will eventually end up in one of the 2^n corners, where each unit is either at its maximum or minimum level of activity. The corner might be a *stable corner* (see below) in which case the vector will remain there; otherwise, the vector will progress to another corner. If a corner is stable, the contra-lateral corner is also.

What defines stable corners? The matrix **A** has n eigenvectors with corresponding non-null eigenvalues. For each one there is at least one pair of contra-lateral stable corners (Golden, 1986). Stable corners are closely related to the distinctive features of the input vectors, as well as the dimensionality of the space (Proulx, 1986). Let's consider the Hopfield-Tank energy function (Hopfield & Tank, 1985):

$$s = -\frac{1}{2} \sum_i \sum_k w_{ik} f_i f_k$$

where w_{ik} is the weighted link between units i and k and f_i the activation of unit i. The feed-back process which leads the state vector to grow corresponds to a gradient descent on the surface of this energy function (Golden, 1986), with stable corners corresponding to some minimum energy level.

Such networks act as input classifiers and the input space is divided into attraction regions with determined boundaries. The network is insensitive to within region differences but

discriminates perfectly between two stimuli placed on different sides of a boundary.

Because of its auto-associative properties, the Brain-State-in-a-Box model is a useful model of classification learning in tasks such as letter perception (Anderson, 1983), word recognition (Golden, 1985), image recognition (Kohonen, 1984) and semantic processing (Kawamoto & Anderson, 1984). On the other hand the correlational learning algorithm has limited power – for instance, the BSB model cannot solve problems such as "exclusive or" or "parity" (Rumelhart et al., 1986). Nonetheless it has the advantage of doing "real time" reconstruction of partial inputs, which is a characteristic shared by few connectionist architectures.

To extend the application of connectionist networks to more complex tasks, such as "exclusive or" classifications, Rumelhart, Hinton and Williams (1986) introduced the feed-forward network and the back-propagation learning algorithm. I now describe their work which overcomes the limits of the previously described architecture.

2.3. Feed-forward networks and the back-propagation learning algorithm

After the publication of the book "PDP processing: Explorations in the Microstructure of Cognition" by McClelland and Rumelhart in 1986, the connectionist field literally exploded. The book summarized most of the

connectionist work under development at that time and introduced a new learning algorithm for layered networks: the back-propagation learning algorithm. This is a generalized non-linear version of the delta-rule (Widrow & Hoff, 1960) and of the perceptron learning algorithm (Minsky & Papert, 1968, 1988). It resolves the following *credit assignment problem* encountered with layered networks: when a network made of layered perceptrons makes a classification error, the difficulty is to determine which perceptrons from the previous layers are responsible for the mistake and thus should be modified. Back-propagation resolved this problem and proved to be so powerful that it suddenly opened up a vast array of new applications.

A schematic view of a feed-forward network is presented in Figure 2.1 In this network each unit in a layer connects to each unit in the next layer but there are no connections within a layer or "backward" from a "higher" to a "lower" layer. There are three types of units: *input, hidden,* and *output*. *Input units* are those for which the activity is determined by outside input; *output units* are units for which the activity is taken as response; the remaining units are the *hidden units*. Activation spreads forward through the layers and the network which can have either no or several layers of hidden units.

The weighted links (or weights) of the network are positive (excitatory) or negative (inhibitory) real numbers. The activity (or output) o_i of any unit i is given by

$$o_i = f(net_i)$$

where f is a nonlinear (usually differentiable) function, and net_i is the net input to unit i defined as

$$net_i = \sum_m w_{im} o_m$$

where w_{im} is a weighted unidirectional link from unit m to unit i and o_m output from unit m. The function f is a *squashing function* which bounds the activity of the unit in a specific range. The most commonly used squashing function is the *logistic function*

$$f(x) = \frac{1}{1+e^{-x}} \qquad (2.1)$$

proposed by Rumelhart, Hinton and Williams (1986). A plot of this function is shown in Figure 2.2. It is a simple semi-linear function with activity in the interval [0,1]. For positive net input the output of the unit is greater than 0.5, and for negative net input the output is smaller than 0.5. Alternative squashing functions that have been used include the sine and Gaussian functions (LeCun, 1989). To reduce the noise in the network and increase the resolving power, thresholds are generally added to the units. As

mentioned earlier, the threshold mechanism can easily be implemented with a *true unit* with fixed activity which is connected to all other units of the network in an inhibitory manner as proposed by Hopfield (1986); thresholds can be changed through modification of the strength of connections with the true unit.

The feed-forward network is essentially a mapping device. Its purpose is to map each stimulus S_i, i=1,..,n, to the associated desired response R_i, i=1,..,n. This architecture is in essence behavioristic and has no provision for dynamic processes. This means that for a network with specific weighted links a given stimulus will always produce the same response with the same latency (which is the time needed for the activation to spread through the layers of the network). Notice that with the feed-forward network, the unit activations must be updated in specific order one layer at a time.

The standard method used to adjust the weighted links in a feed-forward network is the *back-propagation learning algorithm* (Rumelhart et al., 1986). This is a least mean-square error fitting method. The goal is to iteratively minimize an error criterion such that presentation of stimuli S_i will lead to response R_i for each i. Because this statistical fitting process is based on recurrent presentations of a finite set of exemplars it is often referred to as a learning process.

Let the current error associated with S_p/R_p, the stimulus/response pair p, be

$$E_p = \frac{1}{2} \sum_{j=1}^{g} (T_{pj} - O_{pj})^2$$

where O_{pj} is the activity of the output unit j given pattern p and T_{pj} is the target or desired output for that unit, and g is the number of output units. The overall mean-square error $\underline{E}$ computed across patterns is then

$$\underline{E} = \frac{1}{n} \sum_{p=1}^{n} E_p$$

where n is th number of patterns. Now consider $\underline{E}$ as a function of the network weights: $\underline{E}= g(w_{12}, w_{13}, w_{14}...)$ where w_{ij} is a connection from unit i to unit j in the feed-forward net. To minimize $\underline{E}$, we need to solve, for all w_{ij}, the following differential equation

$$\frac{\partial \underline{E}}{\partial w_{ij}} = 0.$$

The non-linearity of this function (due to the nonlinear squashing function) makes resolving this equation non-trivial. A more practical approach is to attempt to iteratively decrease $\underline{E}$ by following the steepest gradient on the error surface. The network is initially set up with small random weights and then, for any weight w_{ij} at any learning trial, the change Δw_{ij} in w_{ij} is set proportional to

$$\frac{\partial E}{\partial w_{ij}},$$

i.e.

$$\Delta\ w_{ij} = -\alpha \frac{\partial E_p}{\partial w_{ij}}$$

where α is a small constant which controls the learning rate (the step size on the error surface). These small steps made on the error surface are always toward lower levels of error and iteratively develop a weight structure associated with minimum level of error (Rumelhart et al., 1986).

To evaluate this derivative the squashing function f must be a differentiable function. Then using the chain rule, we obtain a general recursive formulation which allows us to assign a relative error score δ_{kj} to each unit j in any layer k of the network (for a detail description see Rumelhart et al., 1986). After differentiation and some algebra this yields for any unit j in layer k given pattern p:

$$\delta_{kj} = (T_{kj} - O_{kj})\ f'(net_{kj})$$

if j is an output unit and

$$\delta_{kj} = f'(net_{kj}) \sum_{r} \delta_{(k+1)r}\ w_{(k+1)r}.$$

otherwise. The value δ_{kj} is the contribution of unit j in layer k to the mean-squares error (E).

In these equations f' is the derivative of the squashing function f which for Equation 2.1 is given by

$$f'(x) = f(x)(1-f(x)).$$

For a specific weight w_{ij} connecting unit i in layer k-1 with unit j in layer k the weight update after presenting stimulus/response pair p is

$$\Delta w_{ij} = -\alpha \delta_{kj} f(net_{(k-1)j}). \qquad (2.2)$$

As can be seen, the error score is propagated backward in the network. Once the error criterion δ_{kj} is computed for all units the weight update for w_{ij} is done following Equation 2.2 where α is a small constant controlling learning rate.

Feed-forward nets with the back-propagation learning algorithm are extremely powerful mapping devices. The non-linearity makes it difficult to establish what can and can't be learned given a particular net. At most, some researchers (Volper & Hampson, 1987; Hornik, Stinchcombe, & White, 1988; Baum & Haussler, 1989) have established upper bounds on the number of patterns that can be learned by a feed-forward network of particular size given a specific task.

Feed-forward nets with the back-propagation learning algorithm are a special case of multivariate non-linear regression with the hidden units interpreted as latent variables (LeCun, 1988). Rumelhart et al. (1986) refer to hidden units as the "internal representation". Since, generally, the number of hidden units is smaller than the number of input units, the network does some dimensionality reduction on the input space. We have direct access to this representation and can therefore try to make it meaningful. For instance, Sanger (1989) proposed a statistical approach that he called *contribution analysis* which is similar to cluster analysis; while Rosenberg (1987) developed an analogous technique that he used to interpret the hidden unit representation built by NetTalk (Sejnowski & Rosenberg, 1986), a network that maps English orthography to phonemic. Interestingly enough, while the goal of several statistical approaches such as multidimensional scaling is to infer the psychological unobservable representation, feed-forward networks provide direct access to the internal representations developed by the model.

A unit of the feed-forward net performs computations similar to that involved in logistic regression (Conover, 1973). In both cases the model yields an output value (probability of group membership) between 0 and 1 depending on a set of inputs (covariates) and weights (coefficients). The main difference is that logistic regression generally involves the

maximum likelihood rather than the least mean-square error method. No maximum likelihood estimation method has been developed for layered networks, probably because the recursive formulation involved makes the mathematics fairly complex.

If the non-linearity is removed, the unit reduces to a simple linear regression operator. If we take a whole *linear* feed-forward network and use the back-propagation algorithm it reduces to principal components analysis when the response vectors are mirror images of the stimulus vectors and to canonical correlation for arbitrary response vectors (LeCun, 1988; Hornik et al., 1988). In both cases the hidden units can be directly interpreted as latent variables.

2.4. An application of feed-forward networks to the encoder problem

This section is devoted to the study of the *encoder problem*, a task proposed by Ackley, Hinton and Sejnowski (1985) to evaluate the learning performance of connectionist networks. This fairly complex task is often used as a benchmark to test the power and the efficiency of learning algorithms. For any specific input/target pair both vectors are identical. In most implementations the number of hidden units is small and the task involves coding a set of binary vectors through a small set of real-valued activation levels. The encoder problem is similar to an identification task which involves n

simple stimulus/response pairs. The results reported here will be used later as a comparison basis when a more elaborate model of identification is presented.

I implement the encoder problem in a three layer back-propagation feed-forward network. Four independent variables are of interest: the set size, the number of hidden units, the number of learning trials and the learning rate. One dependent variable will be considered: the mean-square error (MSE) computed at the output of the network.

The n stimuli (respectively, n responses) are represented as n binary-valued, mutually orthogonal input (respectively, target) vectors, i.e.:

INPUT	TARGET
[1000...0]	[1000...0]
[0100...0]	[0100...0]
[0010...0]	[0010...0]
[0001...0]	[0001...0]
•	•
•	•
•	•
[0000...1]	[0000...1].

The number of input (respectively, output) units used to implement the task is n, the number of stimulus/response pairs.

The simulation results presented here are based on a single realization of each simulation. For each simulation the initial weights were assigned small random values in the interval [-0.5, 0.5]. Although the initial weights are small, any specific starting set of weights affects the subsequent

learning process (see Kolen & Pollack, 1990, for a discussion); unfortunately, as mentioned earlier, the long duration of each simulation prevented me from rerunning each simulation several times with different small initial random weights.

To assess the effect of the three independent variables, I ran simulations of networks implementing the encoder problem for set sizes from 4 to 64 with 1 to 4 hidden units. Standard back-propagation was used. All simulations were performed for a total of 10 000 *epochs*: an *epoch* consists of one presentation of each element of the stimulus set. The (asymptotic) MSE computed at the end of the learning process provides a direct basis for comparison between results for different set sizes and various numbers of hidden units.

Effect of learning rate. The learning rate is a constant that controls the step size during the gradient descent. Two preliminary series of simulations were performed with sets of 8 and 16 stimuli and with two hidden units. Various learning rates α were used. Figures 2.3 and 2.4 present the change in MSE as a function of the epoch number for α = 0.05, 0.45 and 0.95[2]. As can be seen, changing the learning rate simply

---

[2]The epochs are numbered using scientific notation where $_{ej}^{i}$ means $i \cdot 10^j$, e.g. $_{e2}^{1}$ means $10^2=100$, $_{e3}^{3}$ means $3 \cdot 10^3=3000$, etc...

changes the speed at which the performance improves, with a ceiling for $\alpha \geq 0.45$. One problem reported by Rumelhart et al, (1986) for large step sizes (i.e. large α) is an oscillating behavior of the weight structure, preventing the network from reaching a lower level of mean-square error; thus to avoid this problem one normally uses small learning rates. For this reason, and because the results reported here suggest that speed of learning does not increase for $\alpha > 0.45$, the learning rate used in most simulations with the feed-forward network reported in this thesis is $\alpha = 0.45$.

Results on learning. Figures 2.5 to 2.8 display on a log-log plot the change of the MSE through learning for networks with from 1 to 4 hidden units and various set sizes. Except for a plateau at the beginning the curves present relatively straight lines until they asymptote. Such straight lines curves on log-log plots are characteristic of the *power law of learning* (Newell & Rosenbloom, 1981). Although not perfect, these results suggest that back-propagation is a plausible model of this aspect of (cognitive) learning. In a subsequent section of this thesis I propose a modified version of the back-propagation learning algorithm that demonstrates a better fit to the behavioral data. The number of trials involved here might appear large compared to the numbers of trials in behavioral experiments; however, the learning trials in a connectionist implementation should be seen as "neural" updates and several such updates might

correspond to a single behavioral trial. When a human being responds to a stimulus and gets feedback, it is conceivable that several neural updates occur while the stimulus, response and feedback are present in the attention span, while in the connectionist implementation a single neural update is performed after each presentation.

Set size effect. As can be seen in Figure 2.9, for a fixed number of hidden units the MSE increases with the set size. These curves are similar to the behavioral curves observed for identification tasks (see Luce, 1986), although the behavioral data is based on performance indicators such as reaction time while the curves reported here involve mean-square error; however, as I discuss in a later section, mean-square error and reaction time are often correlated. Given a fixed number of hidden units the mean-square error increases and asymptotes as n increases, the main difference being that the behavioral data asymptote somewhere over n=10 whereas in the simulation reported here it asymptotes nearer n=30. I will demonstrate later that choosing the stimulus representation carefully leads to a better fit between the simulation and behavioral data.

Effect of the number of hidden units. As mentioned earlier the number of hidden units can be interpreted as the "dimensionality" of the internal representation built by the network. With higher dimensionality more complex mappings can be learned. The results of Figure 2.9 are replotted in Figure

2.10 to explicitly show that the MSE decreases as the number of hidden units increase.

Conclusion. This study of the encoder problem demonstrates that the mean-square error decreases with learning trials, decreases as the number of hidden units is increased, and increases as the set size is increased.

CHAPTER 3

ABSOLUTE IDENTIFICATION: AN OVERVIEW OF THE PARADIGM, DATA AND MODELS

Absolute identification is a task that involves mapping each element of a set of n simple stimuli to a corresponding element of a set of n simple responses. Usually, the stimulus and response dimensions are each unidimensional and one specific response (out of the n possible) is correct for each stimulus. The task is generally structured with repetitive trials, where on each trial one of the stimuli is randomly presented. The subject tries to give, as fast as possible, the corresponding correct response. As described in detail below, performance is measured through two dependent variables: probability correct (PC) which is the probability that a stimulus leads to a correct response and latency or reaction time (RT). Such performance depends on three main independent variables: the set size (the number of stimulus/response pairs), the number of trials performed and (in some cases) the sensorial range along which the stimuli are spread.

3.1. Behavioral phenomena

Here I describe five well documented basic behavioral phenomena linking performance with the independent variables. First, the subject's reaction time depends on the set size (Merkel, 1885; Hick, 1952; Laming, 1966); there is one major exception to this statement (see later). As can be seen in

Figure 3.1, reporting Merkel's (1885) data, the reaction time increases as n increases and (might) asymptote around n=10.

Many variants of absolute identification involving different perceptual and response modalities are described in the literature. Stimuli can be lights, digits on a screen or pure tones. Responses can be motor or verbal. The magnitude of the functional relation linking set size and performance depends on the stimulus and response modalities and the extent to which the responses are a natural mapping of the stimuli. This latter constraint is called *mental compatibility* by Luce (1986) – for example, a digit-voice mapping is more compatible than a digit-key mapping. A survey of the literature by Teichner & Krebs (1974) showed that compatibility plays an important role in the phenomenon; Figure 3.2, adapted[1] from Teichner & Krebs (1974), demonstrates that the standard relation between set size and latency does not hold for compatible stimuli when both the stimuli and responses are complex (e.g. visual and spoken digits).

Also, Theios (1975) performed an experiment designed to establish further the effect of mental compatibility. Figure 3.3, adapted from Theios (1975), presents the relation

[1]Letters are used on the plots instead of symbols because of limitations of the graphics software.

between latencies and set size with compatible (naming) and incompatible (button) responses for complex stimuli (digits). As can be seen, the curve is flatter for the compatible and complex stimuli/responses.

The second main phenomenon links the subject's performance to the number of trials performed and is known as the *power law of practice* (Newell & Rosenbloom, 1981). Drawn on a log-log plot, the learning curve is usually linear, possibly asymptoting for a large number of trials. Figure 3.4, adapted from Kolers (1975), shows such a log-log plot for a simple reading task. The power law of practice is a ubiquitous learning phenomenon observed in most (if not all) activities where practice plays a role – for instance in perceptual/motor skills (Snoddy, 1926; Crossman, 1958); perception (Kolers, 1975; Neisser, Novick & Lazar, 1963), discrimination tasks (Seibel, 1963), memory (Ratcliff, 1978, Anderson, 1982), routine cognitive skills (Moran, 1980), problem solving (Neves & Anderson, 1981) and automatization (Logan, 1988).

The basic form of power law of practice is

$$Perf = bN^{-c}$$

where Perf is a performance indicator such as mean reaction time, b the "amount" to be learned which is the difference between performance at the beginning of learning and perfect

performance, N the number of trials performed and c a positive constant specific to the process under study. However, for real data, the minimum of the curve is likely not to be 0, and thus an asymptote will be observed on the log-log plot. To incorporate this effect and to take into account other effects of practice anterior to the experiment, the power law can be generalized to

$$Perf = A + b(N-E)^{-c},$$

where A is an absolute minimum greater than (or equal to) 0, and E is some level of practice already acquired. For the purpose of the work reported here simple curve fitting using log-log plots will be used, and thus we can expect the curves to reach an asymptote for a large number of learning trials.

The third main phenomenon links set size and the amount of information transmitted. As the set size increases, the amount of information transmitted (the quantity T in Shannon's (1948) theory of information) increases and asymptotes with a possible decrease for large stimulus set sizes (Pollack, 1953; Garner, 1953; Luce 1986). Figure 3.5 presents the Pollack (1953) and Garner (1953) data which show the functional relation between set size and information transmitted for absolute identification of pure tones of equal intensity equally spaced over a frequency range (Pollack data) and over a fixed intensity range for a fixed

(1000 Hz.) frequency (Garner data). The amount of information transmitted is generally evaluated from the stimulus/response matrix. This array has a diagonal configuration (something like a Toeplitz structure; Jenkin & Watts, 1968) with larger values on the main diagonal smoothly decreasing away from the main diagonal. As the set size increases, the performance deteriorates and the "width" of the diagonal band gets larger.

While Pollack concluded that the total *frequency range* along which the stimuli are spread has negligible effect on the amount of information transmitted, Garner demonstrated that the *intensity range* plays an important role. For tasks such as absolute identification of intensity of pure tones, the amount of information transmitted also increases as the separation (larger perceptual range) of the stimuli increases (Garner, 1953, Green & Swets, 1966), but asymptotes as the range gets larger. For visual stimuli, such as lights at different spatial locations or digits, which form most of the behavioral data, range is not one of the main independent variables, providing that the stimuli are not clustered together (Teichner & Krebs, 1974).

The *end anchor effect* (Marley & Cook 1984), also called the *serial order effect* (Vickers, 1979) is another phenomenon observed given a fixed set size and unidimensional stimuli. Performance (as measured by d') is better for stimuli/responses at the ends of the range.

Finally, there are additional data concerning the effect of sequential dependencies in the presentation of the stimuli (Laming, 1968; Ward and Lockhead, 1970; Luce, 1986). Such sequential effects are not discussed in this thesis.

3.2. Non-connectionist models of absolute identification

Two main (non-connectionist) processing architectures have been proposed for modeling choice and reaction time in absolute identification: *serial* and *parallel* (Vickers, 1979; Townsend & Ashby, 1983; Luce, 1986).

The serial architecture. Donders (see 1969 paper), was in 1868, the first known experimentalist to study reaction time in identification tasks. Donders classified choice experiments into three types a, b and c. In each case the subject is instructed to respond as quickly as possible to some stimulus. An a-reaction experiment involves the presentation of one stimulus for which there is only one correct response. A b-reaction experiment involves the presentation of one out of n possible stimuli, each stimulus being associated with one specific correct response. In a c-reaction experiment, a set of n stimuli is used, but the subject is instructed to make a particular response to only one of the stimuli, not responding at all to the other stimuli when they are presented. Donders studied all three experimental paradigms with various stimulus set sizes. He

found that for a given set size the mean reaction times associated with the three types of experiments have the following relation:

b-reaction > c-reaction > a-reaction.

Donders proposed that the difference between the latency in the c-reaction and a-reaction, i.e. RT_c-RT_a, is the time needed to identify which stimulus is presented; he believed that this identification process depended on several non-overlapping processes and that the reaction time difference is a monotonic function of the set size n.

Around the same time Merkel (1885) attempted to characterize the functional link between set size and latency in identification tasks. Using an experimental design where subjects had to identify a stimulus (letter or digit) presented on a screen by pressing a button, he established that the reaction time is proportional to the logarithm of the number of stimuli (n). In attempting to explain this log relation, Hick (1952), Crossman (1955), Welford (1960, 1968), and others, postulated that the identification process is based on a *serial elimination* process. The idea is that identification involves a series of sub-decisions, each one reducing the number of possible response alternatives. In its simplest form, the model postulates that the probability space is cut in half at each decision point until the

response is found. This approach predicts that reaction time is proportional to the logarithm (to base 2) of n. According to Hick, this elimination process is based on a feature decomposition approach with the sub-decisions being taken on the basis of the presence or absence of particular features.

Serial elimination models have three main limitations. First, the assumption of equal time for each sub-test can hardly be defended since finer sub-decisions should be more difficult to achieve and need longer processing time. Second, it is very difficult to establish the nature of the subdivisions used; if features are involved as suggested by Hick, what is their nature and how are they extracted from the stimuli? Finally, experimental data demonstrate the role of learning in identification tasks (see the review article by Teichner & Krebs, 1974). With large amounts of learning the relation between response time and set size becomes flatter and might tend toward a zero slope (Welford, 1968); the serial elimination model cannot account for this learning effect since, despite practice, the number of decisions involved remains fixed, being larger for larger set sizes.

The parallel architecture. Parallel architectures have been proposed by Christie & Luce (1956), Rapoport (1959), Laming (1966), Vickers (1972, 1979), and others, and are based on the idea that identification is performed through a set of concurrent parallel (independent) processes. Vickers (1979) proposes to classify parallel models of absolute

identification into two groups: 1) *parallel elimination models* also known as *parallel exhaustive search models* (Luce, 1986), and, 2) *parallel eventuation models*. Both groups of models assume that a set of n exemplars V_i, i=1,...,n, one for each of the n possible stimuli, S_i, i=1,...,n is stored in memory. After presentation of a specific stimulus S_i, simultaneous comparisons of the input with each exemplar are carried out. The *parallel elimination model* postulates that the response is supplied after all comparisons processes are finished, whereas the *parallel eventuation model* proposes that there is a race between several competing processes and the response is given when the first parallel process terminates. The processing time associated with each stimulus is assumed to follow a probability (generally Gaussian) distribution; if the same variance is assumed for each stimulus, then the parallel elimination model predicts a mean reaction time proportional to $\log_2(n)$ (Vickers, 1979).

The main difficulty with the parallel elimination model is, like the serial model, its inability to explain learning phenomena. Vickers (1972, 1979) demonstrated, with simulation experiments, that parallel elimination cannot explain the change in slope, linking set size and reaction time, observed as learning occurs. Neither making all processes faster nor changing the overall response criteria can adequately match the behavioral phenomenon. This is why Vickers proposed the parallel eventuation model where several processes are racing

to respond. In this conceptualization, all competitive processes have adaptive capabilities, which allows the model to replicate the observed learning curves. Additionally, Vickers proposed a version of the eventuation model with limited resources (a fixed number of parallel processes) which could also predict serial order (end anchor) effects. Because it explains set size, learning and anchor effects this is currently the best non-connectionist model in this domain.

CHAPTER 4

A CONNECTIONIST MODEL OF ABSOLUTE IDENTIFICATION

4.1. Learning assumptions

In the first part of this section I will demonstrate that back-propagation, although very powerful, might not be suited to modeling cognitive learning in identification tasks. The problem concerns the way the network resources are allocated throughout learning. After illustrating this difficulty, I propose a modified version of the back-propagation learning algorithm that is better able to model cognitive learning in identification tasks.

Back-propagation minimizes the mean-square error computed at the output of a feed-forward net. To do this the weighted links of the network are modified following the direction of the steepest gradient computed on the error surface defined in the weighted links space (see Section 2.3). When the resources (i.e. the number of hidden units) are large relative to the number of stimulus/response pairs, the network learns the task readily and performs very well on the whole stimulus set. This was shown in Chapter 2 where I looked at the encoder problem.

In those simulations the mean-square error decreased monotonically as a function of the number of learning trials until it asymptoted, with the slope of the learning curve decreasing as the set size increased (for a fixed number of

hidden units). These characteristics can erroneously lead one to conclude that the learning process and the performance of the network degrade smoothly as a function of the set size. As I now show, this is not necessarily the case.

Figure 4.1 and 4.2 present the overall mean-square error (computed over the stimulus set), and the squared error for each individual stimulus, for the 16-stimulus encoder problem with 2 hidden units; the graphs report the error scores computed over 10 000 epochs where each epoch consists of one presentation of each of the 16 stimuli[2]. The graph of mean-square error shows the expected decrease over epochs. However, inspection of the graph for the individual square error associated with each stimulus shows a very different picture. Obviously, while the network does very well on some stimuli it performs very poorly on others. This is especially surprising since all the stimuli are equally "spaced" in the multidimensional representation (by binary, orthogonal vectors).

I ran additional simulations and did multiple replications varying the set size from 4 to 64 and the number of hidden units from 1 to 4. In all cases the starting weight values were small random numbers. These simulations show that, for

---

[2]On the figures the epoch scale is divided by 50 (Epoch/50), e.g. 30 on the plotted scale corresponds to epoch 1500.

the encoder problem, the number of stimuli learned correctly is roughly equal to 2^h where h is the number of hidden units. As mentioned above, the back-propagation learning algorithm has the "ability" to ignore a subset of the stimuli – Rumelhart, Hinton & Williams (1986) called this the *asymmetric learning ability* of back-propagation. According to these authors back-propagation gains power from its ability to attain at first better performance on a subset of the stimuli and later extend it to the whole stimulus set; this is how, by learning one stimulus/response pair at a time, the feed-forward network can learn the exclusive-or problem. This is surely a desirable characteristic if resources are large and if the goal is to find a solution which is as close as possible to the absolute minimum error, but it has the above (perhaps undesirable) side effect when resources are limited.

Also, when the goal is to model cognitive processes, finding the absolute minimum might not be adequate. Humans are not artificial systems which do perfectly, and the asymmetric learning strategy is certainly psychologically false – at least in simple absolute identification tasks with small set sizes. Behavioral data show that human beings allocate their resources over the stimulus set in these experiments in such a way that performance is to a significant extent the same over all the stimuli; one exception to this generalization being the end-anchor effect (Vickers, 1979; Marley & Cook, 1984), but even here performance is not of the kind shown

above for back-propagation. Finally, Logan (1988) reports some data on cognitive learning showing that the variance of the performance indicators computed across the stimulus set decreases with learning. Logan in his model of automatization postulates that both variance and mean of the performance indicator (latency) decrease following the same power curve. Obviously, in Figure 4.2 variance does not decrease with learning.

Hoskins (1989) proposed a modified version of the back-propagation learning algorithm that he called *focused back-propagation*. In this algorithm, the presentation probability of a given stimulus is made proportional to its associated square error. Hoskins presentation method is not compatible with actual experiments where stimuli are usually presented equally often. Hoskins introduced focused back-propagation to speed up the learning process and one desirable (for us) side effect of Hoskins approach is to reduce discrepancies in the performance across the stimuli. On the other hand, focused back-propagation carries the locus of control for the learning process outside the system and requires extraneous control. The idea of giving more attention to stimuli for which the performance is not very good is certainly interesting but it should be implementable independently of the stimulus presentation probabilities. A way to achieve this would be to use an internal mechanism that ignores already learned stimuli when they are presented – this is

essentially implemented in the technique that I propose below.

In another approach, Lisker (1989) proposed a learning algorithm that maximizes, following gradient ascent, the amount of information transmitted by the network. This agrees with behavioral data suggesting that human beings tend, within the limits of their abilities, to maximize the amount of information transmitted as they learn (Pollack, 1953). Although appealing, Lisker's algorithm is both formally and computationally very complex which greatly limits its potential application.

I propose a modification to back-propagation which allocates the resources of the network in such a way that it tends to perform equally well on each member of the stimulus set. This new algorithm implements a form of *selective attention* — when a stimulus associated with a relatively large (respectively, small) square-error is presented the adaptive modification of the network is made larger (respectively, smaller). The network thus "pays attention" to the (historical) characteristics of the presented stimulus and selectively changes its learning rate. Mean-variance back-propagation devotes more "attention" to a stimulus associated with larger square-error without requiring a change in presentation probability (as does focused back-propagation).

With standard back-propagation the network tends first to learn a subset of the stimuli. The difference in squared error observed between the stimuli already learned and the others is large, giving a large variance for the squared error. The idea underlying my revised back-propagation is simply to attempt to keep this variance small through adaptive changes of the weighted links while, at the same time, keeping the overall mean-square error small. I first derive the necessary formula for the variance term, then combine those results with standard results for the mean-square error term.

Let

$$E_p = \frac{1}{2} \sum_{i=1}^{k} (T_{pi}-O_{pi})^2 \tag{4.1}$$

be the square-error for pattern p, where T_{pi} and O_{pi} are, respectively, the target and observed output for unit i given stimulus p, and where k is the number of output units.

Let

$$E = \frac{1}{n} \sum_{r=1}^{n} E_r \tag{4.2}$$

be the overall mean-square error computed over the set of patterns,

and let

$$\begin{aligned} Var_E &= \frac{1}{n}\sum_{q=1}^{n}\{E_q - \bar{E}\}^2 \\ &= \frac{1}{n}\sum_{q=1}^{n}\{E_q - \frac{1}{n}\sum_{r=1}^{n} E_r\}^2 \end{aligned} \tag{4.3}$$

be the variance of the squared error computed over the set of patterns. Assume for now that we want, through learning, to minimize this variance. The output vector depends on the input vector fed to the network and on the weighted links of the net. Consider Var_E as a function of the weights of the network:

$$Var_E = f(w_{11}, w_{12}, w_{13}, \ldots, w_{nn})$$

where w_{ij} denotes the weighted link between unit i in layer k and unit j in layer k-1. To minimize Var_E we will iteratively modify the weights by a small amount following the steepest gradient on this variance surface. The total derivative of the variance with respect to weight w_{ij} is:

$$\frac{\partial Var_E}{\partial w_{ij}} = \sum_{p=1}^{n} \frac{\partial Var_E}{\partial E_p} \frac{\partial E_p}{\partial w_{ij}}.$$

The right-most element of this equation is the partial derivative of the square error with respect to weight w_{ij}, given pattern p, which is the standard back-propagation error gradient.

The first term inside the summation on the right hand side is (using Equations 4.1, 4.2 and 4.3)

$$\frac{\partial Var_E}{\partial E_p} = \frac{1}{n} \left\{ \sum_{\substack{q=1 \\ q \neq p}}^{n} 2 \left\{ E_q - \frac{1}{n} \sum_{r=1}^{n} E_r \right\} \cdot \left(\frac{-1}{n} \right) \right.$$

$$\left. + \frac{2}{n} \left\{ E_p - \frac{1}{n} \sum_{r=1}^{n} E_r \right\} \left(1 - \frac{1}{n} \right) \right\}$$

$$= \frac{2}{n} (E_p - \bar{E}) - \frac{2}{n^2} \sum_{q=1}^{n} \left\{ E_q - \bar{E} \right\}$$

$$= \frac{2}{n} (E_p - \bar{E})$$

Thus, to adaptively reduce the variance after presentation of the stimulus/response pair p, change the weight w_{ij} by the amount

$$\Delta w_{ij} = - \lambda \frac{\partial Var_E}{\partial E_p} \frac{\partial E_p}{\partial w_{ij}}$$

$$= - \lambda \frac{2}{n} (E_p - \bar{E}) \frac{\partial E_p}{\partial w_{ij}}$$

where λ is a non-negative constant controlling the learning rate. Clearly minimizing the variance has a trivial solution: make the same response for all stimuli. However, the overall mean-square error would then be large. Thus, we will consider minimizing a weighted mixture of the mean-square error and the variance of the error, i.e. minimizing

$$\alpha \underline{E} + \lambda Var_E,$$

where $\alpha,\lambda > 0$. Again, using the steepest gradient we obtain for the change of weight w_{ij} after presentation of a stimulus/response pair p

$$\Delta w_{ij} = - \left(\lambda \frac{\partial Var_E}{\partial w_{ij}} + \alpha \frac{\partial E_p}{\partial w_{ij}}\right).$$

$$= - \left(\lambda \frac{\partial Var_E}{\partial E_p} \frac{\partial E_p}{\partial w_{ij}} + \alpha \frac{\partial E_p}{\partial w_{ij}}\right),$$

$$= - \frac{\partial E_p}{\partial w_{ij}} \left(\alpha + \lambda \frac{\partial Var_E}{\partial E_p}\right).$$

$$= - \frac{\partial E_p}{\partial w_{ij}} \left(\alpha + \lambda \frac{2}{n} (E_p - \underline{E})\right).$$

i.e.

$$\Delta w_{ij} = - \frac{\partial E_p}{\partial w_{ij}} (\alpha + \gamma (E_p - \underline{E})).$$

where

$$\gamma = \frac{2 \lambda}{n}.$$

This equation can be seen as an implementation of a gradient descent with variable step size. Since the activity of every output unit is bound in the range [0, 1], both E_p and $\underline{E}$ are

also bounded in [0, 1], thus the value of $(E_p - \underline{E})$ is in the range [-1, 1] and for $\alpha \geq \gamma$

$$(\alpha + \gamma (E_p - \underline{E})) \geq 0$$

Under that condition, steps are made larger (in absolute value) for patterns associated with square-error larger than $\underline{E}$ and smaller (in absolute value) for patterns associated with square-error smaller than $\underline{E}$. If γ is assigned a value larger than α, an oscillating behavior can be observed preventing the network from reaching lower levels of mean-square error. Most simulations reported in this thesis use the value α=0.45 and λ=0.20 for which $\gamma < \alpha$ for all $n \geq 1$.

As mentioned above, this *mean-variance back-propagation learning algorithm* can be seen as implementing a *selective attention* mechanism that allows the amount of weight change to depend on the relative performance achieved on a specific stimulus. Notice that for stimuli associated with small square-error the step sizes on the error surface are made smaller (in absolute value) leading to finer adjustment of the weight structure. This procedure is similar to other proposals such as *simultaneous annealing* (Kirkpatrick, Gellat & Vecchi, 1983). The simulation results presented next show that mean-variance back-propagation exhibits the general

speed-up of learning that is characteristic of focused back-propagation (Hoskins, 1989).

Simulation results. As a first approximation to a model of absolute identification, I tested the above mean-variance back-propagation algorithm on several versions of the encoder problem varying the number of stimuli and the number of hidden units. Figures 4.3, 4.4 and 4.5, respectively, present the obtained mean-square errors as a function of epoch for the 4, 8 and 16-stimulus sets, respectively, with 2 hidden units. The solid lines represents the MSE observed with standard back-propagation learning algorithm while the dashed line represents the results with mean-variance back-propagation. The learning parameters used where λ=.20 and α=0.45. The simulations were run for a total of 2000 epochs.

The simulation results show that overall mean-variance back-propagation (MV-BP) initially gives faster learning, but when the stimulus set is large relative to the number of hidden units the MV-BP asymptotes faster and has a larger level of mean-square error. This can be explained by the absence of the asymmetric learning characteristic of the BP. A network using MV-BP would certainly have (like a human being) difficulties in learning exclusive-or contingencies since it will not be able to first attain better performance on a subset of the stimuli. Figures 4.6, 4.7 and 4.8, respectively, present plots of the variance of the error computed over the stimulus set through learning for stimulus

sets size 4, 8 and 16. As expected, the variance observed for the MV-BP (dashed lines) stays small. Two additional comparisons were made: Tables 4.1 shows the amount of information transmitted by the network and Table 4.2 shows the average probability of error. This latter statistic is computed over the final 500 learning epochs for set sizes 4,8 and 16, respectively. Figure 4.9 and 4.10 plot these results. As can be seen, the amount of information transmitted is greater and the probability of error is smaller when MV-BP is used.

Conclusion. When applied to a network with a large amount of resources, i.e. a large number of hidden units, the mean-variance back-propagation learning algorithm yields learning curves similar to those observed with the standard back-propagation learning algorithm but with faster learning. When the new learning algorithm is used on a network with limited resources learning is still faster but performance asymptotes at a higher level of mean-square error. The proposed MV-BP learning algorithm might not find the best solution (in terms of minimizing mean-square error), but it is probably more adequate for modeling cognitive learning since it allocates the resources in such a way that performance tends to be similar on all stimuli.

4.2. Architectural assumptions

The major behavioral data gathered on absolute identification are the response probabilities and the latencies. Building a connectionist model of absolute identification requires that one either choose predictors for latencies (e.g. mean-square error) or incorporate real-time processing into the architecture. Seidenberg & McClelland (1989), in their word recognition model, picked the first solution and used the mean-square error provided by a feed-forward network as a predictor of latencies. This approach was motivated by the assumption that the network is a partial implementation of a larger system, with the feed-forward network being a mapping device and with a subsequent decoding module (which they did not present) assumed to compute the response and thus induce latencies. In particular, Seidenberg & McClelland assumed that the larger the error for a given pattern, the longer (on average) will be the decoding time, and hence the response time, for that pattern. On the other hand, Cohen, Dunbar & McClelland (1990) superimposed a cascade structure (McClelland, 1979) over a feed-forward net in order to implement real-time processing; they demonstrated that their implementation could be used to accurately model latencies in Stroop effect paradigms.

To model latency data I use a hybrid architecture made of a mapping module (a feed-forward net) and a decoding (or decision) module (a feedback network). This approach is

different from the one developed by Cohen, Dunbar & McClelland since I implement the mapping and decoding processes in different structures.

Figure 4.11 gives a schematic view of the proposed architecture. The output of the mapping module is fed into the decision module which eventually produces the response. The identification task is here conceived as a dual process. First, a feed-forward network (the mapping device) computes, in a time fixed for all stimuli, a multi-dimensional real-valued vector. The decoding module takes this output from the feed-forward network and make decisions on the presence or absence of the relevant "features". The network latency is provided in network steps by the decoding module. This structure is appealing since mapping and decision processes are implemented independently, and therefore to implement other tasks such as categorization, one simply uses a different decoding module.

Various decoding modules were tried in attempting to model reaction times in absolute identification. Three devices, each made of a single layer of units with recurrent connections, will be discussed here. In increasing order of complexity they are: 1) a network of simple integrators with thresholds similar to the cascade units proposed by McClelland (1979), 2) Koch-Ullman's winner-take-all network (Koch & Ullman, 1985) and 3) the Brain-State-in-a-Box matrix model (Anderson et al., 1977), previously discussed.

4.2.1. A network of simple integrators with thresholds

A network of simple integrators with thresholds consists of a set of simple linear units each associated with a threshold detector. Thus, the response o_i of unit i is given by

$$o_i = \begin{cases} 0 \text{ if } v_i < \theta \\ 1 \text{ if } v_i \geq \theta \end{cases}$$

where θ is a threshold (or bias) common to all units and v_i is the net input to the unit. Each of the units is connected in a one-to-one fashion with one of the output units of the feed-forward network. The state vector V, which depends on feedback connections linking each unit with itself, is given by

$$V(t+1) = \mathbf{A}V(t)$$

where $\mathbf{A}=\mathbf{I}g$ is the connection matrix, i.e.

$$\mathbf{A} = \begin{bmatrix} g & 0 & 0 & \ldots & 0 \\ 0 & g & 0 & \ldots & 0 \\ 0 & 0 & g & \ldots & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & 0 & 0 & g \end{bmatrix}$$

with g controlling the gain of the integrators - larger g means faster responses. In all simulations involving integrators with threshold reported in this thesis and in order to obtain a reasonable range of values for decoding time, g is (arbitrarily) set to 1.05. The first unit to reach

its threshold is taken as the response. In this race, since there are no within-layer connections, each of the n competitive processes are independent. Let $O=[O_1, \ldots, O_n]$ be the output vector from the feed-forward network. If $V[0]=O$, then the unit i, for which O_j is the maximum of O, will be the eventual winner of the race and the first to respond. The response time RT is a function of this maximum value of O. This device takes decision on independent dimensions and is adequate to decode binary vectors where a non-zero value of element j of the vector represents stimulus j, and hence response j.

Finding the maximum of the output vector might appear trivial. Why not just select the "instantaneous" maximum – i.e. the unit at time zero that achieves the maximum? The answer is simple: there is no trivial way of doing this in a connectionist network without postulating some external "meta" system. Using integrator units is certainly one of the easiest solutions. Vickers (1979) claims that this implementation, which he calls parallel eventuation (see earlier), needs to be adaptive in order to fit behavioral data. This is not the case here since adaptation is already implemented in the feed-forward network. Moreover, the results will show that this decoding device can generate latencies matching behavioral data.

4.2.2. The winner-take-all network

The winner-take-all (WTA) network is a decoding device which involves feedback and within-layer inhibitory connections. Let O be the input vector fed to the winner-take-all net. Let V[t] be the state vector at time t in the decoding WTA net and let **A** be the connection matrix. Each unit is a simple non-linear unit such that

$$V(t+1)=\text{trunc}(\mathbf{A}V(t))$$

where trunc is a non-linear squashing function such as

$$\text{trunc}(x) = \frac{1}{1+e^{-x}}$$

which limits the activity of each unit to a bounded range.

The matrix **A** has the following structure:

$$\mathbf{A} = \begin{bmatrix} n\delta & -\delta & -\delta & \dots & -\delta \\ -\delta & n\delta & -\delta & \dots & -\delta \\ -\delta & -\delta & n\delta & \dots & -\delta \\ . & . & . & . & . \\ . & . & . & . & . \\ -\delta & -\delta & -\delta & -\delta & n\delta \end{bmatrix}$$

where the entry a_{ij} is the within-layer connection between unit i and j, while δ is a small constant which controls the gain and thus response speed of the network and n is the number of units in the network.

Starting with the state vector V[0]=0, the activity in the network will evolve until very unit, except one, is at its minimum level of activity (Koch & Ulmann, 1985); the time needed to complete this process is taken as the latency. In most implementations other constraints are built-in so that $||V|| = 1$ and $0 \leq v_i \leq 1$ for all i; this guaranties that the winning unit will be at its maximum level of activity when all other units will be at their minimum. The winning unit is the unit j for which O_j is maximum at time 0. Because of the within-layer inhibition, the latency will depend on the differences in activity across units. If the activity levels are similar, there will be much competition and the latency will be longer. One extreme case is observed if the activity v_i is the same for all units; then the state vector will remain constant and the response time will be infinite. The winner-take-all network is adequate to decode simple binary vectors where a non-zero value of element j of the vector represents stimulus j, and hence response j.

4.2.3. The Brain-State-in-a-Box network

The Brain-State-in-a-Box (BSB) matrix model was discussed in the review section. It is a non-linear matrix network for which the state vector V, at time t+1, is given by:

$$V(t+1) = \text{trunc}(\mathbf{A}V(t))$$

where **A** is the connection matrix and trunc is a squashing function that limits the activation of each unit to a bounded interval. The matrix **A** has a symmetric structure; here, **A** is a scalar multiple of the covariance matrix of the stimuli (see Section 2.3) and has the following structure:

$$\mathbf{A} = \alpha \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{nn} \\ c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\ c_{31} & c_{32} & c_{33} & \cdots & c_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nn} \end{bmatrix}$$

where c_{ij} is the weighted link between units i and j. The time needed to reach a corner of the box (each unit either at its maximum or minimum firing rate) is taken as the response time. This model has the advantage that it can be applied to any response vector regardless of the representation and can be used to decode complex feature vectors. These attributes are important for applications such as word recognition. The BSB might not be adequate to model latencies in simple psychophysics where a single element of the state vector represents a specific stimulus: first, the network might end-up in a corner that does not correspond to a stimulus (e.g. 01100..); second, since the representation vectors are orthogonal (or close to orthogonal) and since the weighted connections of **A** develop following a correlational rule, all (or at least most) of the entries off the main diagonal of **A** would be zeros.

In summary, all three decision modules described here do parallel computation. The simplest one, made of integrator units with threshold detectors, is an implementation of a simple parallel eventuation model (Vickers, 1979). The process ends when one of the units reaches threshold. This system can also be interpreted as a linear distance classifier (Ashby & Gott, 1988) which performs decisions on independent (unidimensional) features. The winner-take-all net does parallel elimination (Vickers, 1979). The response is supplied once a single unit has reached its maximum level of activity with the decision provided depending on the interaction of several parallel processes. It is an integral dimension classifier (see Ashby & Gott, 1988) using a multidimensional classification rule; by integral classifier these authors mean a device that applies classification rules to the joint values of several dimensions (or features) in opposition to a device that applies classification rules independently on each dimensions. The brain-state-in-a-box performs sophisticated parallel elimination based on integral dimensions. Because the connection matrix can have any (symmetric) structure, the network can implement complex decision criteria. This characteristic is important when the response vector needed is a complex pattern of binary features such as in word recognition (see later).

In this connectionist implementation of absolute identification, the adequacy of the *network of simple*

integrators and the *winner-take-all-net* to model latencies will be verified. The brain-state-in-a-box will be used later to model word recognition and naming. As mentioned previously, the connection matrix **A** of this last network is a scalar multiple of the covariance matrix of the stimuli. With simple independent (or near orthogonal) stimuli this matrix will have a diagonal (or near diagonal) structure and reduces to a network of simple integrators.

4.3. Stimulus representation assumptions

Connectionist networks are non-symbolic processing systems and contrast with the symbolic sequential rule-based approach. While in a rule-based system an arbitrary symbol can be assigned a specific (arbitrary) meaning, within the connectionist paradigm "similar" stimuli have "similar" representations. The "meaning" is not independent of the representation and the similitarity between two stimuli can be evaluated by some measure of correlation or distance between their respective representations.

To model a cognitive process, assumptions need to be made about the stimulus representation and the level of processing needs to be clearly specified. For example, in a word recognition model the input could be letter features, simple bit-mapped graphics or whole letters. The representation should carry the characteristics relevant to the specified level of processing (e.g. feature extraction, letter

recognition, word recognition, semantics, etc.); if letter features are used, the similarity between the representations of any two letters should correspond to the perceptual similarity of these letters.

In the case of absolute identification, which involves a set of n stimulus/(correct) response pairs $S_1R_1, S_2R_2, S_3R_3, \ldots, S_nR_n$, the simplest representation that we could use is a set of n binary encoded n-dimensional orthogonal vectors, such as the ones used for the encoder problem, with each stimulus and response represented by a different vector with only one non-zero component, i.e.

STIMULUS	RESPONSE
[1000...0]	[1000...0]
[0100...0]	[0100...0]
[0010...0]	[0010...0]
[0001...0]	[0001...0]
•	•
•	•
•	•
[0000...1]	[0000...1].

But since the psychological representations of real stimuli (and responses) are not always mutually independent this representation might not be adequate. To make it more realistic from a psychological point of view, I will implement a Gaussian sensory trace. Several authors (e.g. Ashby & Gott, 1988; Ratcliff, 1978; Green & Swets, 1966; Vickers, 1979) propose that the presentation of a stimulus generates a sensorial (or psychological) trace, which for stimulus x follows a random variable X. In most of the literature, X has been assumed to have a (multidimensional)

normal distribution, as it will be here. The idea is that a specific stimulus is not associated with an exact (sensorial) representation but rather with a probabilistic (Gaussian) representation.

Here, the pairwise discriminability among stimuli depends on the amount of overlapping of the (Gaussian) distributions. While adjacent stimuli have a higher probability of being confused, the amount of confusion decreases for stimuli further apart. The discriminability of any pair of stimuli can be characterized conveniently by a measure of signal detectability d' proposed by Green & Swets (1966), where

$$d' = \frac{\mu_i - \mu_k}{\sigma},$$

with μ_i (respectively, μ_k) the mean of the Gaussian distribution associated with S_i (respectively, S_k) and σ the standard deviation common to both distributions. Green & Swets (1966) proposed d' as a measure of distances (in standard deviation units) in the *psychological space*.

The value d' depends on the sensory characteristics as well as on the psychological representation. Since the trace associated with a stimulus is not observable, the psychological d' cannot be directly calculated and is thus a quantity that might be theory dependent. Neverless, it is sometimes possible, through the experimental manipulation of the probability of presentation of the members of a stimulus

set, to evaluate d' (see Baird and Noma, 1978, for a discussion).

In the connectionist model described here the sensory trace is assumed to be normally distributed and thus when a stimulus is presented neighboring units are also activated. The level of activation is proportional to the height of a Gaussian distribution and the total level of activation (sum across input units) is equal to one with the exception that distributions located at the end of the sensorial array are truncated. A schematic view of this representation is provided in Figure 4.12. Of course this is a fairly rough approximation of a Gaussian distribution (especially if σ is assumed small) but it is a simple way to reproduce the idea of a sensory trace.

In the simulations, the distance $|\mu_i - \mu_{i+1}|$ between two adjacent stimuli is constrained to be at least 1 and the minimum number of units (input and output) needed for a simulation is n (the set size). This coding schema has limitations: while in the theory of signal detectability (TSD) a stimulus is associated with a (continuous) probabilistic distribution located on an evidence (continuous) axis varying across trials, the sensorial trace is here a discrete representation following the shape of a Gaussian distribution. In an attempt to make the representation more realistic, random noise is dynamically added to each stimulus representation on each trial. This is

a commonly used procedure with connectionist implementations that facilitates generalization (LeCun, 1989) at the cost of slightly decreasing the networks performance. It also has the advantage of adding variability to the responses and allows the generation of response probabilities. A small pseudo-random value is added to each element of the input vector. Each component of the random sequence follows a normal probability distribution with mean zero. The standard deviation of this distribution is adjusted such that the mean of the absolute value of the random numbers is 0.20. Since the maximum value for an element of the input vector is 1, this is said to correspond to 20 percent noise. This value was arbitrarily chosen and it will not be manipulated in order to improve the fit of the simulations to the behavioral data. Other models postulating sensorial Gaussian trace (e.g. Ashby & Gott, 1988; Ratcliff, 1978; Green & Swets, 1966; Vickers, 1979) do not assume this kind of additional random process since they have random sample from a normal distribution at each stimulus presentation – this sampling procedure is responsible for the variability in the input.

For some experimental tasks, such as absolute identification of pure tones (Pollack, 1953; Garner, 1953), the total range along which the stimuli are distributed plays an important role. In order to reproduce this phenomenon in the proposed connectionist model, the distance $|\mu_i-\mu_j|$ of two adjacent stimuli must be manipulated and thus the sensorial d'

(computed on the stimulus input vectors) should be allowed to vary. This can be implemented using a network with a large set of input and output units along which Gaussian distributions representing the stimuli are moved. Unfortunately, this involves large networks that take a long time to simulate and thus only a few simulations using this approach will be reported in this thesis; otherwise, in most of the other simulations the distance $|\mu_i-\mu_{i+1}|$ is equal to 1.

To be an adequate representation the sensory trace must correspond to some d' inferred from the behavioral data and there should be a fixed mapping between the physical range of the stimuli and the sensorial one. In most simulations reported in this thesis, the total sensorial range increases as the set size n increases. Corresponding to the more frequently encountered absolute identification experimental set-up, the stimuli are thought of as lights on a panel, while the responses are thought of as buttons on a panel (Teichner & Krebs, 1974); thus the number of units needed to represent a stimulus set of size n is n. For those simulations involving equally spaced stimuli, the d' (computed from the stimulus representation) associated with any two adjacent stimuli is set equal to 0.75, which, since $|\mu_i-\mu_{i+1}|=1$, corresponds to a standard deviation, σ, equal to 1.33. This is typical of values reported in several identification experiments (Tanner, Swets, & Green, 1956; Swets, 1959; Shipley, 1961; Green & Swets, 1966).

I now provide simulation results that illustrate the effect that implementation of a Gaussian sensory trace has on the learning curves and on the stimulus/response matrix. Since the target vectors used to train the network are viewed as feedback vectors from the environment, the sensorial trace assumption that holds for the primary input vectors might also hold for the feedback vectors. This is why I report results involving both Gaussian input and target vectors. Results involving Gaussian stimuli will be compared with results obtained with a binary (orthogonal) representation. Also, the effects of changing the standard deviation (and thus d') of the Gaussian distributions will be considered.

Simulations. To compare characteristics of the binary and the Gaussian representations I ran four classes of simulations according to the type of representations used for stimulus and target vectors. They are

Orthogonal stimuli / Orthogonal responses (OO),
Orthogonal stimuli / Gaussian responses (OG),
Gaussian stimuli / Orthogonal responses (GO),
Gaussian stimuli / Gaussian responses (GG).

Orthogonal stimuli (or responses) refers to binary non-overlapping stimuli while Gaussian stimuli (or responses) follow (overlapping) normal distributions. Mean-variance back-propagation was used with learning rates α=0.45 and λ=0.20 on a three layer net with 1 hidden unit. Figures 4.13 and 4.14 present plots of the MSE as a function of learning trials for the identification task with 8 and 16 stimuli for

each of the four class of simulations. A total of 10 000 learning epochs were performed in each case, with each epoch consisting of one presentation of each stimulus. As can be seen the MSE curves are substantially similar in all four conditions except for a longer plateau at the beginning of the learning process for the OG and GG conditions.

Table 4.3 (for 4-, 8- and 16-stimulus sets) and Figure 4.15 (for a 16-stimulus set) show for each of the four conditions the stimulus/response matrix computed over the last 500 learning trials where the unit with highest level of activation is considered to be the response. Table 4.4 gives the amount of information transmitted for each condition. In the Gaussian conditions a d' of 0.75 was used. In these results the MSE computed with Gaussian and binary target vectors should not be directly compared since the tasks are quite different; in the binary case the target vector is, except for one unit, made of all zero values while, in the Gaussian case, the target vector is a smooth pattern of activations spread along several units. Inspection of the incidence matrix reveals, for *the larger set size* (n=16), differences in the structure. While the array observed in the OG and GG conditions present a nice smoothly decreasing diagonal structure, the two other incidence matrices show a more erratic structure with entries away from the main diagonal (Figure 4.15). It appears that for the larger set size the stimulus/response matrix does not keep its diagonal

structure except when the feedback (target) vector is assumed to be Gaussian. Notice that increasing the noise level will also cause the stimulus/response array structure to deteriorate but should similarly affect all four conditions.

d' effect. It is expected that, for a fixed number of stimuli, as d' (from the stimulus representation) increases the amount of information transmitted will at first increase but will asymptote as d' increases further. I tested this hypothesis through two sets of simulations with set size 8. To test the effect of changing d' the standard deviation of the Gaussian distributions was modified (keeping the distance $|\mu_i-\mu_{i-1}|$ equal to one). In principle d' could also be modified by changing the distance $|\mu_i-\mu_{i+1}|$ and increasing the total range (and the total number of units). While both procedures lead to equivalent d', keeping $|\mu_i-\mu_{i+1}|$ fixed is computationaly simpler and (a lot) more economical. Notice that it is not clear whether both procedures are theoretically equivalent – in this connectionist implementation, modifying $|\mu_i-\mu_{i+1}|$ affects the total number of input units involved in representing the stimulus set, and as the number of units used is increased the number of connections is increased and presumably so is the ability of the network to learn the task.

In one case I manipulated the d' of the *input vector*, whereas in the second case I manipulated the d' of the *target vector*. Figure 4.16a and 4.16b respectively present plots of the

amount of information transmitted as a function of d' for the OG (Orthogonal stimuli / Gaussian responses) and GO (Gaussian stimuli / Orthogonal responses) conditions. It can be seen that the amount of information transmitted quickly increases and asymptotes as d' (for adjacent stimuli) increases.

Conclusion. The simulations reported here first demonstrated that adding a Gaussian filter on the input vector and/or the target vector does not substantially alter the learning curves observed in a simple absolute identification implementation. However, such filtering changes the structure of the stimulus/response matrix: the stimulus/response matrices observed with Gaussian feedback (target) vectors present a smoothly deteriorating main diagonal structure, which is not the case for large set sizes involving orthogonal feedback (target) vectors.

The results also demonstrate that adding the Gaussian structure on either the input or output representation decreases the amount of information transmitted (Table 4.4). Finally, the amount of overlap of the adjacent Gaussian distributions substantially alters the amount of information transmitted (T) with T increasing and reaching an asymptote as d' increases (Figure 4.16).

4.4. Simulation results for unidimensional stimuli

Now that we have considered the three sets of assumptions involved in building our connectionist model it is time to combine them – i.e. to combine the new MV-BP learning algorithm with the hybrid architecture using a feed-forward net and a decoding module, and with the Gaussian sensory trace. These three topics were previously discussed separately in order to isolate their respective characteristics, so I now briefly review the three sets of assumptions.

Learning algorithm. The mean-variance back-propagation learning algorithm minimizes a weighted mixture of both the mean-square error and the variance of the square error. The learning rates used for all simulations are α=0.45 and λ=.20. In all simulations 10000 epochs were performed, with each epoch consisting of the presentation of each element of the stimulus set in random order – this is equivalent to 10000*n random trials. A large number of trials was performed in order to obtain asymptotic results on which I will focus.

Stimulus representation. Both input (stimulus) and response (target) vectors are binary vectors filtered using a Gaussian filter. The mean of each stimulus (response) is associated with a single input (output) unit and the representation is in that sense binary (one unit for each stimulus). Except when stated otherwise, the number of input

(and output) units used for each simulation equals the set size with the difference in the means associated with any two adjacent stimuli being equal to 1 and with the detectability d' equal to 0.75 – i.e. the standard deviation of the Gaussian distributions is 1.33. In all simulations 20% noise was added to the input vector; as mentioned earlier this is a common (connectionist) procedure that makes processing non-deterministic and allows variability of performance.

Network architecture. The hybrid architecture consists of a feed-forward network and a decoding module. As discussed earlier, two types of decision modules adequate for binary encoded responses are used: a network of simple integrators with threshold and a winner-take-all network. Results obtained with both modules will be compared.

4.4.1. Main results

The simulation results are organized around the behavioral phenomena mentioned in Chapter 3. Two model parameters are considered: the number of hidden units used and the d' associated with two adjacent stimuli. Unless stated otherwise one hidden unit and d' of 0.75 are used.

Learning. Figures 4.17 to 4.19 present the log-log plot of the MSE and the latencies provided by the two different decoding modules: the integrators with threshold (IWT) net and winner-take-all (WTA) net. Results are reported as a function of the number of learning trials for set size 4, 8,

16 and 32 (and associated total ranges 4, 8, 16 and 32 units – see previous comments regarding relation between set size and range in these simulations) with 1,2, 3 and 4 hidden units. As usual, the learning trials were divided into epochs with each epoch involving one presentation of each member of the stimulus set. It can be seen on these graphs that most curves have the expected linear (on a log-log plot) shape with steepest slopes for smaller set sizes and a larger number of hidden units; the curves appear a lot more compressed for the network with one hidden unit but still present the linear relation. The data provided by the WTA network are different and appear much more chaotic. The relations reported in these figures are probably closer to linear than those observed for the encoder problem results reported in Section 2.4.

Set size effect. The previous figures reporting changes through learning highlight differences between simulations with various set sizes (and corresponding ranges). Furthermore, Figures 4.20 to 4.24 present graphs of the asymptotic MSE, probability of error (PE), latencies (IWT and WTA) and the amcunt of information transmitted (IT) computed after 10 000 epochs for various set sizes (and corresponding ranges) and number of hidden units. These results are based on single presentations of each stimulus. The graphs reported here show the change in performance as the set size increases, and in most cases an asymptote is observed. The

MSE, PE and the IWT latencies show the characteristic asymptotic curves that are obtain with real (behavioral) data, but the WTA latencies do not exhibit the same characteristics. Network simulations done with one hidden unit seem to provide curves which closely fit behavioral data -- for instance Figure 4.25 has asymptotic IWT latencies superimposed over Merkel's (1885) data. The fact that one hidden unit leads to a good fit is perhaps not surprising since the stimuli and responses vary along a unidimensional continuum.

The asymptotic curves of the information transmitted (Figure 4.24) observed for networks with 3 or 4 hidden units show a slight decline for large set size (n=32) and although scant behavioral data exist for such large sets, Luce (1986) reports identification experiments for which this decline in information transmitted is observed. However, various data are actually best fit by the simulations with one hidden unit, so this decline with 3 or 4 hidden units is perhaps irrelevant.

Effect of changes in d'. Figure 4.26 presents curves of the amount of information transmitted as a function of the set size for various d' (0.33, 0.66, 0.75) values with 1 hidden unit. The simulation stimulus sets used here are equivalent to "lights" arranged on a panel where the total range increases as n increases. One hidden unit was chosen since the amount of information transmitted for the 2 hidden

units simulations is well over what is observed in the behavioral data. Modifying the d' parameter is like changing the resolution of the sensorial input. The smaller is d', the larger is the amount of overlapping between adjacent stimuli, and the higher the probability of confusion. The simulation results present a somehow chaotic structure – this is (probably) because each simulation data point is a single realization and, as explained earlier, the time needed to run each simulation makes it impossible to replicate all the simulations several times. Nonetheless, as it is seen in Figure 4.26, the simulation data is sensitive to both set size and d effects. In particular, the information transmitted asymptotes as the set size and associated range increases.

Range Effect. The total range along which the stimuli are distributed (for a fixed d' between adjacent stimuli) plays an important role in some behavioral experiments such as absolute identification of pure tones of varying intensities. Smaller ranges are associated with higher degrees of confusion and smaller amounts of information transmitted, but increasing the range has little or no effect beyond a certain point. Figure 4.27 adapted from Braida & Durlach (1972) presents the amount of information transmitted in a 10-stimulus set pure tone identification experiment as a function of the total range. Marley & Cook (1984) demonstrated that their *anchor model* can fit these behavioral

data. This model, like the implementation presented here, postulates a Gaussian sensory trace which, for the Braida & Durlach data, is estimated (taking into account the scale difference between their implementation and the one reported here) to have a standard deviation roughly equal to 1.

In an attempt to replicate this data with my connectionist implementation I performed a series of five simulations each involving a 10-stimulus set. Each stimulus has the Gaussian structure previously discussed with a standard deviation equal to 1. In each of the five conditions the distance $|\mu_i-\mu_{i+1}|$ between any two adjacent (input) stimuli is different ranging from $|\mu_i-\mu_{i+1}|=1$ to $|\mu_i-\mu_{i+1}|=5$. In all conditions the same set of target vectors (with $|\mu_i-\mu_{i+1}|=1$ and $d'=1$) is used. Figure 4.28 presents the results of the simulations. As can be seen, the total amount of information transmitted for the simulation results agrees very well with the behavioral data. Since in my connectionist implementation a 10-stimulus set requires at least 10 units, it was not possible to run simulations for ranges smaller than 10, which is why no simulation points are provided on Figure 4.28 for ranges smaller than 10 units.

Conclusion for main results. From these results we conclude that the simulation model replicates the major features of some important behavioral data on absolute identification. The architecture, learning algorithm, and stimulus representation allow the replication of (1) the

power law of learning for latencies, (2) the set size effect on latency, and (3) the characteristic curve relating information transmitted to stimulus range. The results appear to be especially adequate using one hidden unit and integrators with thresholds in the decision module. In the following paragraphs I provide additional results, continuing to use a feed-forward network with one hidden unit and a IWT decision module.

4.4.2. Additional results

End Anchor effect. The effect is basically not present in the simulated data. Figure 4.29 shows line plots of the probability correct (although most researchers use d' to document this phenomenon) observed for set sizes 4, 8 and 16 with one hidden unit and d'=0.75. The x-coordinate on the graph is the position of the stimulus along the sensorial input range. The end anchor effect is observed only for the 8-stimulus set. Notice that contrary to other modeling approaches to absolute identification, such as Marley & Cook's (1984) or Vickers' (1979) models, no assumption was made in this connectionist implementation that would "naturally" replicate this effect.

Relation between latencies and MSE. Could MSE be a good predictor of the decision time provided by the decoding module connected at the output of the feed-forward network? This, as mentioned earlier, is an important question for anyone interested in connectionist models of latency. Figure 4.30 and 4.31 show, for various set sizes, correlation plots between MSE (x axis) and IWT latencies (y axis) computed over the last 100 learning epochs for simulations with 1 and 2 hidden units. Each plot has 100*n data points, each point on each graph representing an individual MSE and decoding time. The relation is in all cases strongly linear and leads to the conclusion that the MSE is overall highly correlated with the

decoding time provide by the WTA net. Notice that the (IWT) decoding time depends on the (single) activation of the output unit with the highest level of activation, while the mean-square error jointly depends on the activation of all the output units; thus the decoding time and the MSE do not necessarily have to be highly correlated.

Latency distributions. Latency distributions observed in absolute identification tend to follow a log-normal probability distribution (Ratcliff, 1978). One of the main contributions of Ratcliff's memory retrieval model is to fit these distributions. Although no specific attempt was made in the simulations to reproduce this result, the simulated model provides the expected distributions. Figure 4.32 and 4.33 present the latency distributions obtained for stimulus set sizes from 4 to 32. The distributions are from the results of the last 100 learning epochs and represent the response times computed over the stimulus set. Figure 4.34 and 4.35 show the QQ-probability plot based on the rank order of the data (x-axis: theoretical log-normal probability; y-axis: observed frequency) for the distributions with 1 and 2 hidden units. It can be seen that the overall fit is very good except for larger set sizes. On these plots deviations from a 45 degree straight lines indicate deviations from the theoretical distribution. The higher slopes for larger set sizes indicate observed distributions with limited ranges. From these results we can conclude that, for the set sizes usually

studied ($n \leq 10$), the distributions of simulation latencies follow the log-normal distributions observed in the behavioral data.

Correct versus incorrect responses. The data from absolute identification experiments usually show observed (mean) latencies differing for correct and incorrect responses. Overall, incorrect responses usually take longer to provide than do correct ones (Hick, 1952; Hale, 1969; Stanovich, Pachella, & Smith, 1977) unless the subjects are specifically instructed to respond as fast as possible (Hale, 1969; Stanovich, Pachella, & Smith, 1977). Figure 4.36 and 4.37 presents respectively bar graphs of mean-square errors and mean latencies (IWT) observed for various set sizes computed over the last 1000 learning epochs; in all cases the mean response time is smaller for the correct responses.

When subjects are instructed to respond faster, or when the experimental set-up forces them to do so, the increase in speed is usually inversely proportional to the decrease in the information transmitted (Hick, 1952; Hale, 1969); i.e. the plot of mean response time against the amount of information transmitted is linear. In the simulations reported here no attempt was made to replicate this experimental manipulation, although a possible implementation might be achieved through systematic manipulation of the thresholds of the decoding module.

Compatibility effect. As discussed earlier in the thesis, mental compatibility is an important variable when studying absolute identification. The stimulus/response representation used in the simulations reported earlier assumed perfect compatibility with both the stimuli and responses being "simple". But what if, as in Theios' (1975) experiment, the input vector is made "complex" and the output vector either "simple" (incompatible) or "complex" (compatible)? As previously discussed, in the complex stimulus / complex response case, the behavioral data do not show the usual increasing relation between set size and response time (see Figure 3.2).

I ran simulations where the type of input and output representations (simple or complex) were manipulated yielding four possible conditions, i.e.

Simple Stimuli / Simple Responses (Compatible)
Simple Stimuli / Complex Responses (Incompatible)
Complex Stimuli / Simple Responses (Incompatible)
Complex Stimuli / Complex Responses (Compatible).

The simple stimulus and response representations were the previously discussed Gaussian unidimensional representation and are thought of as arrays of "lights" (stimuli) and of "keys" (responses), while the complex representation is either based on an orthographic (word stimulus) or phonemic (naming response) representation.

The complex representations use features and involve a position specific (slot) encoding scheme where n slots are

required to represent n letters (phonemes). Each letter used in the input vector is encoded using Gibson's (1969) features representation while phonemes used to encode complex responses are represented using features proposed by Rumelhart & McClelland (1986). A total of 11 features are used to represent each phoneme while 28 features are used for each letter.

The (visual) words "one" to "ten" are represented by up to 5 letters (140 features) while the (spoken) words "one" to "ten" are represented by up to 7 phonemes (77 features)[3]. If the representation needed is shorter than the maximum, a blank code (all features off) is used to complete the string. These complex representations are the same used in my work on word recognition presented in Chapter 5.

Obviously, with the complex response vectors (phonemic), a simple IWT decision module is inadequate, so in that case I use a BSB network. The BSB network was trained on the phonemic regularities of more than 2000 English words over a total of 500 epochs. Within each epoch 200 phonemic vectors were presented, each one picked randomly without replacement and with probability proportional to the relative frequency of the word as reported by Kucera & Francis (1967). This procedure is described in more detail in Chapter 5. The

---

[3]in most experiments digits are used not words.

decoding module was trained independently of the feed-forward network using the correlational learning rule for matrix model described in Section 2.2 of the thesis.

For all simulations the feed-forward net was trained for a total of 10 000 learning epochs. For set size 2 {"one", "two"} is used while {"one", "two", "three"} is the 3-stimulus set and so on. The feed-forward network was trained *independently* for each stimulus set while the *same* decoding module, trained on the overall English phonemic regularities, is used in all cases. I assume this to be equivalent to real conditions where human subjects have access to already learned phonemy but learn specific mappings depending on the task requirements. When the response is complex, the decoding time supplied by the BSB network is taken as response latency, while for the simple responses an IWT net is used. Figure 4.38 and 4.39 report the asymptotic mean-square error and mean reaction time as a function of the set size for the four conditions. As can be seen, the complex stimuli / complex responses condition presents a flatter MSE curve and essentially constant latency as the set size increases. When compared with Figure 3.2, it appears that the latency results provided here follow quite consistently the behavioral data except for the simple/complex condition which (in the behavioral light-voice condition) is associated with the largest latencies.

Laming (1968) and Luce (1986) suggest that the flatter relation observed for the digit/voice condition can be explained by the fact that the stimuli (digits) are so overlearned that, despite a subset of the digits being used, the subjects continue to behave as if there were 10 stimuli. Since the latencies in the digit-key condition increase with set size, the controlling factor must be the response aspect, not the decision (or identification) one. This is supported by the previously discussed experimental results from Theios (1975) presented in Figure 3.3.

The same explanations could certainly be used to account for the simulation results; when complex output is required the BSB network which was trained extensively on English phonemic regularities is used and despite an increase in mean-square error as set size increases, the time needed to provide the phonemic feature vector for the words {"one", "two", ..., "ten"} remains constant independently of set size. The observed latency curve for the simple/complex condition is harder to explain; the increase in latency as set size increases is small (although larger than in the complex-complex case). The extensively learned phonemic regularities can probably explain the small magnitude of this increase while the effect itself could be attributed to the more complicated mapping (shown by larger MSE). This agrees with the behavioral data for which the light-voice condition is associated with the largest latencies, indicating a more

complicated mapping, although, in the simulated experiment, the simple/complex condition is not associated with the largest latencies.

Obviously, the results presented here cannot account fully for the behavioral data, but it appears that the model is sensitive to the complexity and compatibility factors.

4.5. Extension to multidimensional stimuli

The simulated model presented in the previous section could fit many behavioral phenomena concerning unidimensional absolute identification. The results reported in this section suggest that the proposed approach can also be used to model absolute identification of separable multidimensional stimuli.

Most data gathered on the identification of multidimensional stimuli concern the structure of the stimulus/response matrix. The *biased choice model* for identification axiomatized by Luce et al. (1963), following a first formulation by Shepard (1957), leads to an impressive fit of such behavioral data (e.g. Townsend, 1971, Smith, 1980; Townsend & Ashby, 1982; Townsend & Landon, 1982, Nosofsky, 1985). According to this model the probability that a subject makes response j given stimulus k, $P(R_j|S_k)$, is given by

$$P(R_j|S_k) = \frac{\beta_j \eta_{kj}}{\sum_{m=1}^{n} \beta_m \eta_{km}}$$

where $0 \leq \beta_j$, $\eta_{kj} \leq 1$, $\sum_{j=1}^{n}\beta_j = 1$, $\eta_{kj} = \eta_{jk}$, and $\eta_{kk} = 1$. The β_j parameters are interpreted as response biases while the η_{kj} parameters are interpreted as the similarity between the stimuli S_k and S_j. The number of parameters involved in this model is large: there are n-1 freely varying response bias parameters and n(n-1)/2 freely varying stimulus similarity parameters.

Nosofsky (1985, 1986, 1987) uses a multidimensional scaling (MDS) approach to replace the stimulus similarity parameters by a smaller set of coordinate parameters. The η_{kj} are then seen as a function of the distance in the psychological space, i.e.

$$\eta_{kj} = f(d_{kj}),$$

where d_{kj} is usually taken to be the Euclidean distance between stimulus k and stimulus j. Nosofsky showed that the best fit of this combined *MDS-choice model* is achieved with Euclidean distances and a Gaussian similarity function, that is with

$$\eta_{kj} = e^{-(d_{kj})^2}$$

This new parsimonious model gives very good fit to the relevant data. For instance, Nosofsky (1985) presents experimental data on a set of bi-dimensional stimuli and

provides comparison between the MDS-choice model and Luce's biased choice model demonstrating that his approach also provides very good fit to his data. I propose a connectionist implementation of the same task and will compare the MDS-choice solution and the internal representation developed by the network. As I will demonstrate, the connectionist implementation provides a stimulus/response matrix similar to that obtained from a person. The connectionist network also develops a representation that closely matches the psychological representation inferred from the MDS-choice model.

The task I implemented is analogous to the identification task described by Nosofsky (1985). In this experiment, subjects were presented with one of 16 stimuli. The stimuli varied along two independent dimensions: size (four levels) and angle of orientation (four levels). For both dimensions the progression from one level to the next is linear in the physical scale. The values (4X4) were combined orthogonally to yield 16 stimuli. The subjects were instructed to press one out of 16 buttons arranged in a 4X4 panel. This button arrangement was compatible (in the obvious sense) with the stimulus arrangement and each stimulus is associated with a single button. Feedback was given after each presentation. Since the structure of the stimulus/response matrix is the dependent variable and since little data exist on the latency of absolute identification of multidimensional stimuli, the

implementation described here does not incorporate a decoding module and no attempt is made to predict response times.

Stimulus representation. Nosofsky's task involved two separable dimensions. The network implementation is also based on separable dimensions. A set of 16 input units is used. Each unit corresponds to one of the possible inputs. The units are assumed to be arranged in a 4X4 array – i.e. the stimuli have the following arrangement:

a b c d
e f g h
i j k l
m n o p

Following the general approach used for the unidimensional case, the sensory trace is assumed to follow a 2-dimensional normal distribution; also, on each dimension the detectability, d', between any 2 adjacent stimuli is set equal to 0.75. Sections of the Gaussian representation that do not fall within the 4X4 array are truncated. The target vector is also assumed to be 2-dimensional normal and has the same structure as the input vector. Figure 4.40 shows the 16 bi-dimensional normal stimuli.

Network architecture. The network implementation used a feed-forward network with either one or two hidden units.

Learning. The mean-variance back-propagation algorithm previously described is used. A total of 10 000 learning

epochs were performed, where, as usual, each epoch consists of one presentation of each member of the stimulus set. Twenty percent noise was added to the input vector.

Results. Figure 4.41 shows the learning curves plotted on a log-log graph and demonstrates the expected linear curve for MSE versus learning trial for the simulations with one and two hidden units. As expected, the results obtained with the 2-hidden units network present a much steeper slope. Table 4.5 (respectively, Table 4.6) presents the stimulus/response frequencies computed over 500 trials after the learning phases for 1 (respectively, 2) hidden units, while Table 4.7 adapted from Nosofsky (1985) shows the same stimulus/response matrix produced by a human for the 2-dimensional identification task. As can be seen the 2 hidden units simulation and the real data matrices present a similar structure with performances appearing slightly better in the simulated results. But remember, the 20% noise added in the simulation was quite "arbitrary", and not selected to give the best fit to the real data. Clearly the 1 hidden unit solution is not adequate – as can be seen the matrix presents several null (all zeros) columns, thus some responses are never produced by the network while other responses are produced with high probability. It seems that the network is unable to learn correctly the mapping for the 16 stimuli and ignores some responses.

Following the approach described in Luce, Bush & Galanter (1963), I estimated the values of the bias parameters and fitted the choice model to the 2 hidden-units simulation results presented in Table 4.6. The fit is plotted in Figure 4.42 while Figure 4.43, adapted from Nosofsky (1985), presents the fit of Luce's model to the behavioral data of Table 4.7 – the x axis represents the expected frequency while the y axis is the observed one. As can be seen both fits are very good.

Using the simulated stimulus/response matrix and following the approach described by Nosofsky (1985) with his assumptions about the similarity/distance function (Gaussian) and distance computation (Euclidean), I computed a MDS solution based on Torgeson's (1958) classical approach (Splus statistical software – Becker, Chambers & Wilks, 1988). I obtained a configuration (Figure 4.44) very similar to the one reported by Nosofsky (Figure 4.45, adapted from Nosofsky, 1985). Figure 4.46 presents the one dimensional (1 hidden unit) internal representation developed by the network while Figure 4.47 gives the two dimensional (2 hidden units) internal representation both before (left figure) and after (right figure) removing the non-linearity – the non-linearity is removed by applying the inverse of the squashing function to the internal representation. As can be seen, the one hidden unit internal representation does not allow good discrimination among stimuli while the 2 dimensional (2

hidden units) internal representation built by the network is very regular and allows good discrimination. This internal representation (Figure 4.44), like a MDS configuration, is invariant under rotation and appears (except for rotation) very close to the configuration provided by the MDS-choice model for the real data (Figure 4.45) as reported by Nosofsky (1985). Moreover, the one dimensional solution reveals the source of the null columns observed in the one dimensional stimulus/response matrix – several stimuli (e.g. m,j,g,d) are represented in a very similar way and thus will lead to the same response. Note also that with one hidden unit the network learned to represent the stimuli following one of the two relevant (physical) dimensions – namely, the vertical dimension in table on page 92, giving no confusion within the stimulus subsets {a,b,c,d}, {e,f,g,h}, {i,j,k,l,} or {m,n,o,p}..

Conclusion. The results reported in this section demonstrate that the proposed connectionist model for absolute identification can be successfully extended to bi-dimensional stimulus sets using a network with 2 hidden units. It is reasonable to believe that these results could be extended to n-dimensional stimulus sets with n hidden units. The number of hidden units involved is very important; as demonstrated here, if the network does not have the relevant number of hidden units it cannot learn the task while, as demonstrated previously with the unidimensional case, if the number of

hidden units is too large the performance of the network is too good to be a realistic model. Finally, the performance observed in the simulation demonstrates a striking resemblance to Nosofsky's (1985) behavioral data and can be modeled using Luce's biased choice model. Moreover, the internal representation developed by the network is extremely similar to the ones inferred from the MDS analysis of the stimulus/response matrix gathered from simulation and behavioral observation.

CHAPTER 5

COMPLEX IDENTIFICATION: A MODEL OF WORD RECOGNITION AND NAMING

The previously proposed model was applied to simple absolute identification, with the results showing that it could fit several phenomena for uni- and bi-dimensional stimuli. An important question to ask is how connectionist models can be used to model identification of stimuli of greater complexity. This chapter presents such a model of word recognition and naming; Lacouture, 1988, gives the results in a condensed form.

The proposed connectionist model of word recognition and naming is based on a hybrid architecture similar to the one used in the previous sections, in which a layer of processing units (a decision module) maps, in real time, the noisy output from a feed-forward network onto a binary valued (multidimensional) response vector.

5.1. Historical summary and the behavioral phenomena

Many studies have explored the effects of orthographic redundancy, orthographic-phonological regularity, and relative frequency of words on recognition latencies (see Seidenberg, 1985, for a review). In this Chapter I will be concerned with word regularity and word frequency effects on naming latency. Regular words contain a spelling pattern, common to a large pool of words, which always has the same

pronunciation (e.g. "MUST" is regular like "RUST" or "DUST"). Exception words, in contrast, have a spelling pattern not pronounced according to the spelling-sound rules of English (e.g. "HAVE" is irregular in opposition to "PAVE" or "DAVE"). The large set of studies examining the processing of regular and irregular words in conjunction with their relative frequencies provide some well established results. I will be concerned with two central phenomenon. First, there are *frequency effects*: higher frequency words are named faster than lower frequency words (Waters & Seidenberg, 1985). Second, there is the *regularity effect*: longer latencies for exception words are specific to lower frequency words (Seidenberg et al., 1984; Seidenberg, 1985; Waters and Seidenberg, 1985).

The early attempts to model latency in the word identification and naming paradigm postulated the use of pronunciation rules to map orthography to phonemy (see Venezky, 1970; Wijk, 1966, Hanna, Hanna, Hodges and Rudorf, 1966). This view was influenced by the very popular information processing approach advocated by Newell and Simon (1963). According to the rule-based approach, a set of rules is applied to the orthographic representation to derive the corresponding pronunciation. Complex mappings require more processing and thus lead to longer response latencies.

The difficulty in establishing an exhaustive set of pronunciation rules, and the fact that a large body of words

are not pronounced following the known rules, seriously challenged this approach. Coltheart (1978) proposed an extension of the rule-based approach whereby two competitive processes are involved, one being a rule-based mechanism, the other an exception look-up process. This new view, called the *dual-route* model, is able to explain the processing of exception words which do not follow the pronunciation rules. The look-up mechanism is slower so that most of the time the response is provided by the rule-based system. If the rule mechanism does not lead to a phonemic interpretation or if the response involves very extensive computation, then the look-up system wins the race and provides the response.

In a controversial paper, Glusko (1979) challenged the dual route model. He demonstrated that pronunciation is influenced by the knowledge of similarly spelt words and that latencies can be explained by a single mechanism. He developed the idea of pronunciation *consistency* which is defined in terms of between word support and competition. The proposed *activation synthesis model* replaced rule-based decisions by a process involving a set of between word connections or associations. Regular consistent words get support from similarly pronounced words while inconsistent words are opposed to many similarly spelled words with a different pronunciation. Glusko's model is part of the "associationist" zeitgeist that is well illustrated by Ratcliff's (1978) work.

Although studied further by other authors (e.g. Marcel, 1980; Kay & Marcel 1981) the consistency effect proved to be difficult to delineate and was only clearly demonstrated recently by Jared, McRae and Seidenberg (1990).

5.2. Connectionist models of word recognition

NetTalk. Sejnowski and Rosenberg (1987) supplied the first implementation of a naming task in a connectionist network. Their network, NetTalk, is a feed-forward net which produces the phonetic representation associated with the orthographic representation of a letter according to context (the immediately adjacent letters). The network has an input windowing system. The letters of the word to be pronounced are scrolled such that three letters are fed into the network at a time – the one to be pronounced, the preceding one (previous context) and the following one (following context). Blank codes are used to mark the beginning and the end of a word. The letters are encoded in a simple binary fashion. There is a unit for each possible letter plus one for the blank code. Each input vector is made of 3*29=87 units, and each output vector is made of 33 (binary) units and allows the representation of 32 phonemes and a blank code. Learning is through back-propagation.

Although limited, this implementation proved that a connectionist model could be used to learn the mapping from orthography to phonemy. NetTalk could adequately map

orthography to phonemy for regular and irregular words. Moreover, the output level provided by the network for each phoneme proved to be roughly proportional to the probability of confusion (from behavioral data) among phonemes given a specific letter in a specific context.

Seidenberg and McClelland's model. At the same time that I was implementing the model reported here, Seidenberg and McClelland (1989) developed a similar model. Although both models use a feed-forward network and the back-propagation learning algorithm they are distinct in several respects. First, the implementations use completely different stimulus representations. Seidenberg & McClelland specifically encoded the context of graphemes and phonemes using the "wickelphones" and "wickelgraphs" representations. Wickelgraphs were proposed by Rumelhart & McClelland (1986), while wickelphones, proposed by Seidenberg & McClelland, are their phonemic equivalent. The wickel approach allows the representation of several letters (or phonemes) and their respective context at the same time using one representation vector. By contrast, I used a simple left-justified position specific encoding scheme with a blank filler which does not provided any built-in context information. Thus, in my implementation, the network must itself extract the graphical and phonemic regularities. Second, while Seidenberg's model was trained on monosyllabic words, the lexicon that I used consist of over 2000 mono- and multi-syllabic English words

of from one to seven letters. Third, while Seidenberg used the mean-square error measure to predict naming latencies, the implementation that I proposed uses a decoding module to map, in real time, the noisy output from the feed-forward network to the response.

The final difference has to do with the size and scope of the model. The implementation built by Seidenberg and McClelland used a network of 1000 units (it runs on a CRAY-class computer) while the model that I proposed used less than 500 units (it runs on a SUN 3-80 computer). My goal was to study possible implementations of large scale identification tasks in a connectionist architecture, whereas the Seidenberg and McClelland model was extended and used to fit a set of phenomenon beyond the scope of the model that I present here. Nevertheless, despite the differences in implementation both models led to some remarkably similar results.

5.3. Architectural assumptions

The model that I propose uses a modified three layer feed-forward network (see Figure 5.1). An input layer of units codes orthographic information. This layer is connected to a layer of hidden units which output to a third layer used to code phonemic information. The units in the output layer are connected in a one-to-one fashion with an added layer of *completely interconnected* units. These interconnections include feedback connections and allow each of these units to

reach, with a variable number of iterations, either their maximum or minimum firing rate. This interconnected layer is equivalent to a Brain-State-in-a-Box (BSB) model (Anderson et al., 1977; Anderson, 1982; see Section 2.3). The BSB model takes as input a noisy stimulus (here the output from the feed-forward network) and maps it into a more probable state where all units are close to their maximum or minimum firing rate. Two factors control the latency of this process: first, the amount of noise in the input and second, which features are in error. Because the BSB has within layer connections which provide between feature support, the same amount of error on different units can lead to different reaction times. For instance, two output patterns P_a and P_b from the feed-forward network each with associated square error $E_a = E_b$, when fed into the BSB can lead to different response times if one of the patterns is closer to some of the distinctive features (the eigenvectors of the connection matrix; see Section 1.2) than is the other.

5.4. Stimulus representation assumptions

The model was trained on 2106 words (see Appendix 3). These words, chosen from Kucera and Francis (1967), include all uninflected English words up to seven letters in length with a relative frequency greater than 32. Conjugated verbs and compound expressions were excluded. Also, approximatively 500 low frequency words were included to complete the lexicon. These words were represented using a position specific (or

"slots") encoding schema. The orthographic code was derived from Gibson's (1969) features representation, with a total of 28 features used for each letter. Since the stimuli are up to 7 letters long, a total of 112 units is needed. To encode words shorter than seven letters blank codes (all features off) are used. The phonemic code is derived from Rumelhart and McClelland's (1986) phoneme classification with the features used corresponding to the place of articulation, voice or voiceless characteristic of the phoneme, etc. Following this schema 11 features are used to encode each phoneme. The phonemic code is also position specific. Up to 77 units are used to encode the phonemic information.

5.5. Learning assumptions

Learning occurs through a series of epochs. Each epoch consists of the presentation of 400 words. A single word can only be seen once in the same epoch. Words are selected with a probability proportional to their relative frequency, without replacement.

Training involves two independent processes. First, the connections within the decoding module are learned using the delta rule following the general method described by Golden (1985, 1986). A noisy normalized phonemic stimulus is presented, the system is allowed to settle (up to 10 iterations are executed), and correlational learning (see Section 2.2) is applied on the resulting state and weights

are updated. For this learning procedure the learning rate was fixed and equal to 0.01. A total of 100 epochs were run.

The second learning phase is based on the back-propagation learning algorithm (Rumelhart et al., 1986). This procedure concerns the three feed-forward layers of the network. In fact, in this phase of the learning the fourth layer is ignored. Mapping between the orthographic and phonological codes is learned. A total of 250 epochs were run, and 150 hidden units were used. Throughout the learning process, the learning rate was fixed and set to 0.45 and the connections were updated after each stimulus presentation. The task involved here is complex. A large set of hidden units (150) was used. Because the resources were large, only slight differences were observed between the modified back-propagation proposed in this thesis and the standard one. Results reported here were obtained with the standard back-propagation learning algorithm.

5.6. Simulation results

The model was tested periodically during the second learning phase. For each test stimulus the orthographic code was fed into the network. The forward pass was done through the hidden units and a phonemic code was computed. From this output an error score was computed, this score being equivalent to the error score provided by the Seidenberg and McClelland model. This output is also the starting "state

vector" from which the decoding module (the BSB) will iteratively settle with all units close to their minimum or maximum firing rates. The dynamics of this process can simply be represented by a single equation. If we represent the connections within this fourth layer by a square (77x77) matrix **A** where the element $\mathbf{a}_{ij}$ is the connection between unit i and j, and if we represent the level of activity in the layer by a real valued "state vector" V[t] where $v_i[t]$ is the activity of unit i, then the state vector at time t+1 evolves dynamically as a function of the state vector at time t following the equation:

$$V[t+1] = \text{trunc}(\mathbf{A}V[t])$$

where V[t] and V[t+1] are column vectors, **A** is the connectivity matrix and trunc is the squashing function which limits the activity of the units in the interval [0, 1] (see Section 2.3).

As the system dynamically evolves, the length of the state vector (the level of activity) will grow. As an example of this process the change in length after the presentation of the word "AISLE" is shown in Figure 5.2. The system is said to have reached an interpretation when the length reaches an asymptote – defined by a length increase in an iteration of less than .05. As discussed in Section 2.3 of the thesis the system will then have reached a corner of the space.

The test stimuli were the 44 words used by Waters & Seidenberg (1985) in a series of experiments (see Appendix 2). Four types of test word were used: Regular Low frequency, Regular High frequency, Irregular Low frequency and Irregular High frequency. Two dependent variables were measured: An error score at the output from the feed-forward net and the number of iterations the decoding module took to settle. The system was tested periodically while learning the orthographic/phonemic correspondence. Results for both the error scores (Figure 5.3) and the latencies (Figure 5.4) are presented. It can be seen that for the error score both the frequency effect and the regularity by frequency interaction developed through learning – i.e. the model, in concordance with the behavioral data, has better performance for high frequency and/or regular words but does substantially worse on low frequency irregular words.

As an indication of the stability of the learning, the correlations were computed between simulation results for individual words and the median from the distribution of latencies obtained from human subjects. The change in this correlation as learning progresses is presented in Figure 5.5. It provides a quantitative indication of the goodness of fit of the model. The results show that the correlation increases with learning and that a similar fit is obtained for both the error scores and the latencies.

5.7. Conclusion.

Three conclusions can been drawn from these results. First, as far as error scores are concerned, the Seidenberg and McClelland (1989) results on frequency and regularity effects are replicated using a completely different representation and a different subset of English vocabulary. Second, the noisy output from the feed-forward network can be adequately decoded by an additional module which can provide real time latencies. This decoding process maps a noisy output vector to a binary feature vector with decoding time proportional to the mean-square error of the starting vector. Finally, at a more general level and in concordance with other results reported in this thesis, this implementation of a word recognition model demonstrates the adequacy of the connectionist paradigm to model the latencies of complex cognitive processes.

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1. Summary

The main goal of the work reported here was to study possible implementations of time dependent processes in connectionist models. In Chapter 2 I reviewed the major connectionist models and presented results documenting some characteristics of a connectionist implementation of the encoder problem. Simulation results were provided to give the reader a feeling for the functioning of the feed-forward network and to allow subsequent comparison with a connectionist implementation of absolute identification.

The results demonstrated that for this problem the mean-square error computed at the output of the network decreases with learning trials and increases as the number of hidden units is decreased and as the set size is increased. The observed learning curves, when plotted on a log-log graph, appeared similar to the straight lines (following the power law of practice) observed in the behavioral data.

Chapter 3 was devoted to a review of the absolute identification paradigm, data, and non-connectionist models and in Chapter 4, the core of this thesis, I presented a connectionist model of absolute identification. For each of the three sets of assumptions involved in building a connectionist model – i.e. learning, architectural and

stimulus representation assumptions – relevant considerations were discussed. First, in Section 4.1, I looked at learning and documented a learning characteristic of back-propagation and feed-forward networks (Rumelhart et al., 1986) that does not match behavioral data. In an implementation with limited resources (limited number of hidden units) of a simple identification task (based on the encoder problem), it was shown that that the network tends to devote its resources to learning a subset of stimuli while ignoring the others.

I proposed a modification of the back-propagation learning algorithm, *mean-variance back-propagation* (MV-BP), which allocates the resources of the network in such a way that it tends to perform equally well on each member of the stimulus set. This new algorithm was said to implement a form of *selective attention* whereby the adaptive modification of the network's weighted links depends on how much the square error for a particular stimulus deviates from the overall mean-square error.

With standard back-propagation, the network tends first to learn a subset of the stimuli and the difference in squared error observed between these stimuli and the others is large, giving a large variance for the average squared error. The idea underlying my revised back-propagation is simply to attempt to keep this variance small through adaptive change of the weighted links while, at the same time, also keeping the overall mean-square error small. The algorithm thus

minimizes a weighted mixture of both mean-square error and variance of the squared error.

The simulation results showed that the proposed *mean-variance back-propagation* (MV-BP) learning algorithm initially has faster learning, but when the stimulus set is large relative to the number of hidden units MV-BP asymptotes faster and has a larger asymptotic mean-square error. Also, the results demonstrated that the amount of information transmitted tends to be larger and the probability of error to be smaller when the MV-BP is used. Overall, the new MV-BP learning algorithm proved to be a better model of cognitive learning than did the "classical" BP learning algorithm.

In Section 4.2 I presented some considerations regarding the architecture of the network. To model latency I proposed a hybrid architecture made of a mapping module (a feed-forward network) and a decoding (or decision) module (a feedback network); thus the mapping and decoding processes are implemented in different structures.

Three decoding devices were considered, each made of a single layer of units with recurrent connections: 1) a network of simple integrators with thresholds similar to the cascade units proposed by McClelland (1979), 2) the Koch-Ullman winner-take-all (WTA) net (Koch & Ullman, 1985) and 3) the Brain-State-in-a-Box (BSB) matrix model (Anderson et al.,

1977). This hybrid architecture was subsequently used to model absolute identification and word recognition.

In Section 4.3 of the thesis I discussed stimulus representations for absolute identification. The stimuli and responses used in absolute identification are (usually) unidimensional. To make the representation realistic from a psychological point of view, a Gaussian sensory trace implementation was used. A similar sensory trace is assumed in several non-connectionist models of identification (e.g. Braida & Durlach, 1972; Ashby & Gott, 1988; Ratcliff, 1978; Green & Swets,1966; Vickers, 1979; Marley & Cook, 1984) where the presentation of a stimulus generates a sensorial (or psychological) trace which follows a normal distribution.

The simulations reported in this section first demonstrated that adding a Gaussian filter on the input vectors and/or the target vectors does not substantially alter the learning curves observed in a simple identification implementation. However, it was shown that the filtering does change the structure of the stimulus/response matrix observed *for large set sizes*. While the matrices observed with 16 orthogonal stimulus and/or target vectors have an erratic structure, those observed when the target vectors follow a Gaussian sensory trace present a nice diagonal pattern with larger values on the main diagonal smoothly decreasing as one leaves the diagonal.

The results also demonstrated that adding the Gaussian filter increases the stimulus confusion and decreases the amount of information transmitted. Changing the amount of overlap of the adjacent Gaussian distributions substantially altered the amount of information transmitted (T), with T increasing and reaching an asymptote as d' increases.

In Section 4.4, the previously described mean-variance back-propagation, hybrid architecture and Gaussian sensory trace were used to model absolute identification. The simulation results replicated some important behavioral data on absolute identification - (1) the typical power law of learning (linear log-log plots of performance) was clearly replicated for both mean-square error and decoding time using a network of simple integrators with thresholds (IWT); (2) the set size effect on reaction time was clearly replicated; (3) the characteristic asymptotic information transmitted curves (as a function of the set size and range) was also replicated. The results appeared to be especially adequate with one hidden unit and IWT net as the decision module.

Additional simulation results were provided that demonstrated that the model (with one hidden unit) could very well replicate real latency (log-normal) distributions and some correct-incorrect response effects in reaction times. A strong relation (close to linear) was also demonstrated between mean-square error and decoding time provided by the IWT net for stimulus set sizes in the usual range. On the

other hand, the model does not produce the end anchor effect in a consistent manner and simulations performed using the complex representations showed limited success in replicating compatibility effect.

In Section 4.5 the simulated model was extended to absolute identification of separable bi-dimensional stimuli. The results reported demonstrated that the proposed connectionist model can be successfully extend to bi-dimensional stimuli using a network with 2 hidden units. It is reasonable to believe that these results could be extended to n-dimensional stimulus sets with n hidden units. The stimulus/response matrix provided by the simulation bore a striking resemblance to Nosofsky's (1985) behavioral data and could be modeled using Luce's biased choice model. Also, the internal representation developed by the network appeared extremely similar to those inferred from MDS analyses of the stimulus/response matrices gathered from simulated and behavioral observations.

In Chapter 5 of the thesis I presented results of a simulated model of word recognition and naming that was intended to replicate two central phenomena - frequency effects (longer latencies for low frequency exception words) and the regularity effect (longer latencies for exception words) as described by several authors (e.g. Seidenberg, 1985; Seidenberg et al., 1984; Waters and Seidenberg, 1985).

To model word recognition I used a hybrid architecture similar to that described previously, now using a brain-state-in-a-box network for the decoding module. The stimulus representation was based on a position specific (or "slots") encoding schema. The network was trained with more than 2000 English words. Three conclusions were drawn from the results. First, as far as error scores are concerned, frequency and regularity effects (and their interaction) are replicated. Second, the noisy output from the feed-forward network proved to be adequately decoded by the BSB module to provide latencies. This decoding process allowed the network to map a noisy output vector to a binary feature vector with decoding time proportional to the mean-square error of the starting vector. Finally, at a more general level and in concordance with other results reported in this thesis, this implementation of a word recognition model demonstrated the adequacy of the connectionist paradigm to model latencies of (complex) cognitive processes.

6.2. Scope, limits and possible extensions

6.2.1. On resources and learning

Despite some limitations, the reported results strongly support the use of connectionist networks to model identification tasks. It was demonstrated that a connectionist implementation can account for both cognitive learning and performance in identification tasks. The

implementation provided explicit processing mechanisms and representations to explain cognitive learning, response times and limits on performance. On the other hand, the results showed that a connectionist model does not necessarily give a good fit to behavioral data. As shown in the study of the encoder problem, the ubiquitous power law of practice was not very well reproduced with the standard back-propagation learning algorithm. Moreover, in Section 4.1, standard back-propagation proved to be a poor model of learning and performance on simple identification – the performance of the network was either too good with large resources or was very deficient for a subset of the stimuli when the resources were limited.

The connectionist implementation makes explicit the nature of resource limitations. Resources are limited by the number of processing elements (hidden units) available. First, the dimensionality of the internal representation is limited by the number of hidden units; moreover, as the number of possible signals (stimuli) increases the processing load on each hidden unit increases and the performance deteriorates. It was seen that unidimensional absolute identification can be modeled using a network with one hidden unit while two hidden units were needed when bi-dimensional stimuli were used.

When limited, resources must be used parsimoniously. I believe that the proposed MV-BP algorithm entails a more

efficient use of the computational power of the network by distributing the resources over the whole stimulus set. This algorithm is not based on a software "hack" nor does it rely on an ad hoc modification – theoretical and empirical considerations guided its development and in its actual form the mean-variance back-propagation implements a form of selective attention via the well developed steepest descent approach.

In this connectionist implementation, the mechanism implementing learning is explicit: weighted links are strengthened and an internal representation developed in order to minimize an error criterion. The choice of the appropriate error criterion is thus of paramount importance. The mean-square error criterion used in back-propagation was chosen for mathematical and computational simplicity and not for psychological adequacy. We should consider that mean-square error might not be a good learning criterion for modeling cognitive learning. Mean-square error is not a natural performance indicator; in that respect other criteria such as information transmitted might be more appropriate. Lisker (1989) has already demonstrated the possibility of linking weight changes with overall information transmitted in a feed-forward network. Since behavioral data demonstrate that, as they learn, human beings tend, within their capacities, to maximize the amount of information

transmitted, this criterion might prove to be more suited to modeling cognitive learning.

Perhaps the main limitation of the model proposed here comes from the necessity to decide a priori on the number of hidden units to be used. While one hidden unit appears sufficient to model unidimensional absolute identification, the network needed two units for bi-dimensional stimuli and 150 to model word recognition. Instead of a "pre-wired" approach, it is easier to conceive of cognition in a device that dynamically allocates resources as requested by the task. Ash (1989) demonstrated that this could be implemented in a feed-forward network where new nodes (hidden units) are added to the network as requested by the task. Ash demonstrated that this approach allows the network to develop a minimal dimensionality solution, although it is not clear that this approach could be used to model cognitive learning – especially one can ask what would prevent the system from allocating enough units to always get perfect performance?

The question of resources is also an important consideration in non-connectionist models of absolute identification. Vickers (1979) and Marley & Cook (1984) both used a limited capacity system to model absolute identification. Both models postulate limited resources to explain the end anchor effect. In Vickers' conception the availability of a fixed number of parallel processes explains the limitation while in Marley & Cook's model there is limited representational capability

that bounds the discriminability among stimuli in the psychological space.

Although it has limited capacity (limited number of hidden units), the model of absolute identification that I proposed could not replicate the end anchor effect. A possible way to extend the proposed architecture in order to explain this effect would be to use a network with a *limited number of connections*. In the simulations reported here the tasks associated with larger set sizes were simulated using a larger number of input and output units. This means that for a fixed number of hidden units the number of weighted connections increases proportionally with the set size. In a more constrained implementation each hidden unit would have a fixed number of connections with the input and output layers - thus associated with each hidden unit would be a limited *receptive field,* with these receptive fields overlapping more for smaller set sizes than for larger ones. Since the receptive fields located at the ends of the input and output vectors would have smaller overlaps with other fields it is reasonable to assume that performance would be better at the ends. A similar mechanism was proposed by Murdock (1960) to account for stimulus discrimination. In any case, it is clear that the proposed connectionist approach is valuable in implementing and testing models with limited resources.

As demonstrated, not only using the right amount of resources is important to modeling cognitive processes, but also the

representation used to encode the stimulus and response vectors must also be carefully studied, and this even when implementing a simple task such as absolute identification. The proposed representation for absolute identification of simple stimuli used Gaussian sensory traces. The standard deviation, σ, was set at 1.33 giving a d' of 0.75 between adjacent stimuli; this value of d' was used as it is commonly reported in simple psychophysical identification experiments. In most simulations networks with n input and n output units were used (n being the set size), keeping the sensorial d' for adjacent stimuli constant while the total range increased with n. The task implemented was conceived as a behavioral experiment involving arrays of "lights" and "buttons". It was shown that the power law of learning and the set size effect on latency could be replicated. On the other hand, this stimulus representation might not be adequate for modeling absolute identification of stimuli such as pure tones varying in intensity where the total range has a significant influence on the level of performance obtained (Garner, 1953).

The reported simulations have shown that an increase in information transmitted is observed as a function of set size for stimulus sets with variable range and fixed d' for adjacent stimuli. Also, decreasing the d' (for adjacent stimuli) given a specific range and set size decreases the amount of information transmitted (Figure 4.26).

When all the stimuli for different set sizes are bounded in the *same* range (as is typical in the absolute identification of pure tones of varying intensities), the theory of signal detectability predicts that the d' associated with adjacent stimuli decreases as n increases. Simulations performed using 10-stimulus sets of *variable* range (thus variable d' for adjacent stimuli) demonstrated that the simulated model yields range effects similar to what is observed in the behavioral data (see Figure 4.27). We can thus be reasonably confident that the simulated model of absolute identification can handle both size and range effects.

Unfortunately, as mentioned earlier the simulation of stimulus sets with different d' values requires much larger networks (typically with 10*n input and output units) for accurate results and is computationally very costly. This is why the range effect and its interaction with set size were not investigated further in this thesis (although I am currently performing additional simulations with various set sizes with a fixed total range).

6.2.2. On reaction time

The hybrid architecture that I proposed proved to be a simple and efficient way to implement time-dependent processes in a connectionist network. It was demonstrated that the decoding time is proportional to mean-square error and that decoding latencies match behavioral data. This implementation agrees

with a modular view of cognition (e.g. Fodor, 1983) whereby specialized structures are devoted to different tasks – in this case either mapping or decoding. The proposed network is thus interpreted as part of a larger structure where specialized modules are activated according to task requirements (e.g. categorization or identification; verbal or motor responses). This view agrees with a suggestion by Nosofsky (1988) that categorization and identification are both based on the use of the same internal representations but with different decision processes.

Modularity is a very important concern of connectionism. Single networks have limited applications. While a single feed-forward network can be used to model word recognition, it is hard to conceive that a similar (feed-forward?) device could also model sentence or semantic processing at any interesting level. A system with structural constraints where sub-structures are devoted to different tasks is easier to imagine. Although the architecture proposed in this thesis (with mapping and decoding modules) is a step toward modular connectionism, difficulties need to be overcome before one can implement larger scale modular networks. For instance, one major problem is the implementation of dynamic interactions between sub-networks such as those known to exist between orthographic, phonemic and semantic processes.

Regrettably, my proposed architecture currently lacks an "interactive" component and the decoding process is

completely determined by the output from the feed-forward network. Additionally, for each trial, the process is independent of the previous ones. As a consequence, this implementation does not produce "top-down" effects (such as priming), neither does it replicate sequential effects (Luce, 1986) where the decoding process is altered by the previous stimuli and responses.

An appealing extension of the proposed architecture would involve adding a recurrent structure to the network whereby (partial) output from the feed-forward network could become a component of the input fed to the network at the next time step. Several researches are underway to determine the conditions for stability of such recurrent networks and to evaluate their learning characteristics (see Pineda, 1987; Jordan, 1988). To simulate this kind of network several problems need to be overcome: since most neural network simulators use fixed stimulus sets and do not allow recurrent connections new software has to be developed; moreover, it has not been shown that available learning algorithms (such as back-propagation) can be applied successfully to tasks where the set of possible input vectors changes dynamically. Nevertheless, I believe that recurrent networks will prove very useful for the implementation of time dependent processes and might allow the development of highly modular connectionist architectures.

6.3. Conclusion

The work reported here has demonstrated a possible implementation of time dependent identification processes in a connectionist model. A new connectionist learning algorithm (the MV-BP) has also demonstrated that it can be used to model cognitive learning in simple absolute identification tasks. Finally, the implementation of a word recognition model has proved the feasibility of building a connectionist model of more complex identification tasks.

REFERENCES

Ackley, D.H., Hinton, G.E. & Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 147-169.

Anderson, J.A. (1983). Cognitive and psychological computation with neural models. *IEEE Transactions on Systems, Man, and Cybernetics*, 5, 799-815.

Anderson, J.A. & Bower, G.H. (1973). *Human Associative Memory*. Washington, D.C.: V.H. Winston.

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.

Anderson, J.R. (1982). Acquisition of cognitive skills. *Psychological Review*, 89, 369-406.

Ash, T. (1989). Dynamic node creation in backpropagation networks. Technical Report ICS-8901. Institute of Cognitive Science, University of California, San Diego.

Ashby, F.G. & Gott, R.E. (1988). Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology: Memory and Cognition*, 14, 33-53.

Baird, J.C., & Noma, E. (1978). *Fundamentals of Scaling and Psychophysics*. New York, NY: Wiley.

Baum, E.B., & Haussler, D. (1989). What size net gives valid generalization? In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems I*. San Mateo, CA: Morgan Kaufmann.

Becker, R.A., Chambers, J.M. & Wilks, A.R. (1988). *The New S Language. A Programming Environment for Data Analysis and Graphics*. Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Braida, L.D. & Durlach, N.I. (1972). Intensity perception II. Resolution in one-interval paradigms.

Christie, L.S. & Luce, R.D. (1956). Decision structure and time relations in simple choice behavior. *Bulletin of Mathematical Biophysics*, 18, 69-76.

Cohen, J.D., Dunbar, K. & McClelland, J.L. (1990). On the control of automatic processes: A parallel distributed processing account of the Stroop effect. *Psychological Review*, In press.

Coltheart, M. (1978). Lexical access in simple reading tasks. In G. Underwood (Ed.), *Strategies of Human Information Processing*. London, England: Academic Press.

Conover, W.J. (1973). *Practical Nonparametric Statistics*. New York, NY: Wiley.

Crossman, E.R.F.W. (1955). The measurement of discriminability. *Quarterly Journal of Experimental Psychology*, 7, 176-195.

Crossman, E.R.F.W. (1958). A theory of the acquisition of speed-skill. *Ergonomics*, 2, 153-166.

Donders, F.C. (1969). On the speed of mental processes. [Translation of the original 1868 paper]. In W.G Koster (Ed.), *Attention and performance II, Acta Psychologica*, 30, 432-435.

Eich, M.A. (1982). A composite holographic associative recall model. *Psychological Review*, 89, 627-661.

Feldman, J.A. & Ballard, D.H. (1982). Connectionist models and their properties. *Cognitive Science*, 6, 205-154.

Fodor, J.A. (1983). *Modularity of Mind*. Cambridge, MA: MIT Press.

Fong, Y.S. & Jensen, E. (1988). A study of the delta rule. Technical Report, Department of Electrical and Computer Engineering, Yale University.

Gabor, D. (1948). A new microscopic principle. *Nature*, 161, 777-778.

Gabor, D. (1949). Microscopy by reconstructed wavefronts. *Proceedings of the Royal Society, Series A*, 147, 454-487.

Gabor, D. (1968a). Holographic model of temporal recall. *Nature*, 217, 584(a).

Gabor, D. (1968b). Improved associative holographic model of temporal recall. *Nature*, 217, 1288-1289.

Garner, W.R. (1953). An information analysis of absolute judgements of loudness. *Journal of Experimental Psychology*, 46, 373-380.

Gibson, E.J. (1969). *Principles of Perceptual Learning and Development*. New York, NY: Meredith Corp.

Gluck, M.A. & Bower, G.H. (1988a). Evaluating an adaptative network model of human learning. *Journal of Memory and Language*, 27, 166-195.

Gluck, M.A. & Bower, G.H. (1988b). From conditioning to category learning: An adaptative network model. *Journal of Experimental Psychology: General*, 3, 227-247.

Gluck, M.A. , Bower, G.H. & Hee, M.R. (1989). A configural-cue network of animal and human associative learning. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Glusko, R.J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance,* 5, 674-691.

Golden, R.M. (1985). A developmental neural model of word perception. *Proceedings of the Seventh Annual Meeting of the Cognitive Science Society,* Hillsdale, NJ: Erlbaum.

Golden, R.M. (1986). The "Brain-State-in-a-Box" neural model is a gradient descent algorithm. *Journal of Mathematical Psychology,* 30, 73-80.

Green, D.M. & Swets, J.A. (1966). *Signal Detection Theory and Psychophysics.* New York, NY: Wiley.

Hale, D.J. (1969). Speed-error tradeoff in a three-choice serial reaction task. *Acta Psychologica,* 81, 428-435.

Hanna, P.R., Hanna, J.S., Hodges, R.E. & Rudorf, E.H. (1966). Phoneme-grapheme correspondences as cues to spelling improvement. Washington, DC: U.S. Department of Health, Education and Welfare.

Hanson, S.J. & Pratt, L.Y. (1988). Comparing biases for minimal network construction with back-propagation. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems I.* San Mateo, CA: Morgan Kaufmann.

Hebb, D.O. (1949). *The Organization of Behavior.* New York, NY: Wiley.

Hick, W.E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology,* 4, 11-26.

Hinton, G.E. (1981). Implementing semantic networks in parallel hardware. In G.E. Hinton & J.A. Anderson (Eds.), *Parallel Models of Associative memory*. Hillsdale, NJ: Erlbaum.

Hinton, G.E. (1984). Distributed representations. Technical Report CMU-CS-84-157, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

Hinton, G.E. (1986). Learning distributed representations of concepts. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Hinton, G.E. & Anderson, J.A. (Eds.) (1981). *Parallel Models of Associative Memory*. Hillsdale, NJ: Erlbaum.

Hopfield, J.J. (1986). Neurons with graded response have collective properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 79, 2554-2558.

Hopfield, J.J. & Tank, D.W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52, 142-152.

Hornik, K., Stinchcombe, M., & White, H. (1988). *Multilayer feed-forward networks are universal approximators*. Unpublished Manuscript, Department of Economics, University of California, San Diego.

Hoskins, J.C. (1989). Speeding up artificial neural networks in the "real" world. MCC Technical Report STP-049-89, Austin, TX: MCC corporation.

Jared, D., McRae, K. & Seidenberg, M. (1990). Consistency effects in reading aloud. Unpublished Manuscript, Department of Psychology, McGill University.

Jenkins, G.M. & Watts, G. (1968). *Spectral Analysis and its Applications*. San Francisco, CA: Holden-Day.

Jordan, M.I. (1988). Supervised learning and systems with excess degrees of freedom. MIT/COINS Technical Report 88-27. MIT, Department of Computer and Information Science, MIT, Cambridge, MA.

Kawamoto, A.H. & Anderson, J.A. (1984). Lexical access using a neural network. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Kay, J. & Marcel, T. (1981). One process, not two, in reading aloud: Lexical analogies do the work of nonlexical rules. *Quarterly Journal of Experimental Psychology*, 33A, 397-413.

Kirkpatrick, S., Gellatt, C.D. & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.

Koch, C. & Ullman, S. (1985). Shifts in selective visual attention: Toward the underlying neural circuity. *Human Neurobiology*, 5, 219-227.

Kohonen, T.(1977). *Associative Memory: A System-Theoretical Approach*. New York, NY: Springer-Verlag.

Kohonen, T.(1984). *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag.

Kolen, J.F. & Pollack, J.B. (1990). Back-propagation is sensitive to initial conditions. Technical Report 90-JK-BPSIC, Laboratory for Artificial Intelligence Research, Department of Computer and Information Science, Ohio State University.

Kolers, P.A. (1975). Memorial consequences of automatized encoding. *Journal of Experimental Psychology: Human Learning and Memory*, 1, 689-701.

Krishnaiah, H. & Kanal B. (1982). *Handbook of Statistics, Vol 2: Classification, Pattern Recognition, and Reduction of Dimensionality*. Amsterdam, The Netherlands: North-Holland.

Kucera, H. & Francis, W.N. (1967). *Computational Analysis of Present Day American English*. Providence, R.I.: Brown University Press.

Lacouture, Y. (1988). From mean square error to reaction time: A model of word recognition. *Proceedings of the Second Summer School on Connectionist Models*. San Mateo, CA: Morgan-Kaufmann.

Laird, F., Rosenbloom, B. & Newell, A. (1981). *Universal Subgoaling and Chunking. The Automatic Generation and Learning of Goal Hierarchies*. Hingham, MA: Kluwer Academic Publisher.

Laming, D.R.J. (1966). A new interpretation of the relation between choice reaction time and number of equiprobable alternatives. *British Journal of Mathematical and Statistical Psychology,* 19, 139-149.

Laming, D.R.J. (1968). *Information Theory of Choice-Reaction Times*. London, England: Academic press.

Le Cun, Y. (1988). A theoretical framework for back-propagation. *Proceedings of the Second Summer School on Connectionist Models*. San Mateo, CA: Morgan-Kaufmann.

Le Cun, Y. (1989). Generalization and network design strategies. Technical Report CGR-TR-89-4, Department of Computer Science, University of Toronto.

Lisker, R. (1989). How to generate ordered maps by maximizing the mutual information between input and output signals. Technical Report RC-14624, Yorktown, NY: IBM Research Division, Watson Research Center.

Logan, G.D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95, 1-36.

Longuet-Higgins, H.C. (1968). Holographic model of temporal recall. *Nature*, 217, 104.

Luce, R.D. (1986). *Response Times: Their Role in Inferring Elementary Mental Organization*. New York, NY: Oxford University Press.

Luce, R.D., Bush, R.R. & Galanter E. (Eds.) (1963). *Handbook of Mathematical Psychology Vol I*. New York, NY: Wiley.

Marcel, T. (1980). Surface dyslexia and beginning reading: A revised hypothesis of the pronunciation of print and its impairments. In M. Coltheart, K. Patterson & J.C. Marshall (Eds.), *Deep Dyslexia*. London, England: Routledge and Kegan Paul.

Marcus, C.M. & Westervelt, R.M. (1989). Dynamics of analog neural networks with time delay. In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems I*. San Mateo, CA: Morgan Kaufmann.

Marley, A.A.J. & Cook, V.T. (1984). A fixed rehearsal capacity interpretation of limits on absolute identification performance. *British Journal of Mathematical and Statistical Psychology*, 37, 136-151.

Marquard, D.W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of Industrial and Applied Mathematics*, 2, 431-441.

McClelland, J.L. (1979). On the time-relations of mental processes: An examination of systems of processes in cascade. *Psychological Review*, 86, 287-330.

McClelland, J.L., Rumelhart, D.E. & Hinton, G.E. (1986). The appeal of parallel distributed processing. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing. Explorations in the Microstructure of Cognition, Vol 2.* Cambridge, MA: MIT Press.

McCulloch, W.S. & Pitts, W.H. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 6, 114-133.

McNaughton, B.L.& Morris, R.G.M. (1987). Hippocampal synaptic enhancement and information storage within a distributed memory. *Trends in Neuroscience*, 10, 408-415.

Merkel, J. (1885). Die zeitlichen Verhaltnisse der Willensthatigkeit. *Philosophische Studien*, 2, 73-127.

Minsky, M. (1954). *Neural nets and the brain-model problem.* Ph.D. Thesis, Department of Mathematics, Princeton University, Princeton, N.J.

Minsky, M. & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* Cambridge, MA: MIT press.

Minsky, M. & Papert, S. (1988). *Perceptrons: An Introduction to Computational Geometry, Expanded Edition.* Cambridge, MA: MIT press.

Moody, J. & Darken C. (1989). Fast learning in multi-resolution hierarchies. Technical Report YALEU-DCS-RR-654, Department of Computer Sciences, Yale University.

Moran, T.P. (1980). *Compiling cognitive skill.* AIP memo 150, Xerox PARC, Palo Alto, CA.

Morrison, D.F. (1967). *Multivariate Statistical Methods.* New York, NY: McGraw-Hill.

Mumme, D.C. (1988). Storage capacity of the linear associator: Beginnings of a theory of computational memory. Technical Report 88-1485, Department of Computer Science, University of Illinois at Urbana-Champaign.

Murdock, B.B. (1960). The distinctiveness of stimuli. *Psychological Review*, 67, 16-31.

Murdock, B.B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, 89, 609-626.

Myers, D.E., Schvaneveldt, R.W. & Ruddy, M.G. (1974). Functions of graphemic and phonemic codes in visual word recognition. *Memory and Cognition*, 2, 309-321.

Neisser, U., Novick, R. & Lazar, R. (1963). Searching for ten targets simultaneously. *Perceptual and Motor Skills*, 17, 427-432.

Neves, D.M. & Anderson, J.R. (1981). Knowledge compilation: Mechanisms for the automatization of cognitive skills. In J.R. Anderson (Ed.), *Cognitive Skills and their Acquisition*. Hillsdale, NJ: Erlbaum.

Newell, A. & Rosenbloom, R. (1981). Mechanisms of skill acquisition and the law of practice. In J.A. Anderson (Ed.), *Learning and Cognition*. Hillsdale, NJ: Erlbaum.

Newell, A. & Simon, H.A. (1963). GPS, a program that simulates human thought. In E. Feigenbaum and J. Feldman (Eds.), *Computers and Thought*. New York, NY: McGraw-Hill.

Nosofsky, R.M. (1984). Choice, similarity and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 10, 104-114.

Nosofsky, R.M. (1985). Overall similarity and the identification of separable-dimensions: A choice model analysis. *Perception and Psychophysics*, 38, 415-432.

Nosofsky, R.M. (1986). Attention, similarity and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115, 39-56.

Nosofsky, R.M. (1987). Attention and learning in the identification and categorization of integral stimuli. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 13, 87-108.

Nosofsky, R.M. (1988). Exemplar-based accounts of relations between classification, recognition and typicality. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 14, 700-708.

Nosofsky, R.M. (1989). Further tests of an exemplar-similarity approach to relating identification and categorization. *Perception and Psychophysics*, 42, 279-290.

Pavel, M. & Gluck M.A. (1988). Constraints on adaptative networks for modeling human generalization. *Proceedings of the November 1988 NIPS conference*. San Mateo, CA: Morgan Kaufmann.

Pike, R. (1984). Comparison of convolution and matrix distributed memory systems for associative recall and recognition. *Psychological Review*, 91, 316-338.

Pineda, F.J. (1987). Generalization of back-propagation to recurrent and higher order neural networks. *Physical Review Letters*, 59, 2229-2232.

Pollack, I. (1953). The information of elementary auditory displays, II. *Journal of the Acoustical Society of America*, 25, 765-770.

Pollack, J.B. (1987). Cascade back-propagation on dynamic connectionist networks. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Pollack, J.B. (1989). No Harm Intended, Book Review of M.L. Minsky & S.A. Papert (1988) Perceptrons: An Introduction to Computational Geometry, Expanded Edition. *Journal of Mathematical Psychology*, 33, 358-365.

Proulx, R. (1986). Etude des propriétés de selectivité et de catégorization d'un système de mémoire parallèle et distribuée et de son application à une tâche de reconnaissance de lettres. Unpublished Ph.D. thesis, Department of Psychology, University of Montreal.

Rapoport, A. (1959). A study of disjunctive reaction times. *Behavioral Science*, 4, 299-315.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, 85, 59-108.

Ratcliff, R. (1990). Connectionist models of recognition and memory - constraints imposed by learning and forgetting functions. *Psychological Review*, 97, 285-308.

Ratcliff, R. & Murdock B.B. (1976). Retrieval processes in recognition memory. *Psychological Review*, 83, 190-214.

Rosenberg, C.R. (1987). Revealing the structure of NetTalk's internal representation. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Rosenblatt, F. (1959). Two theorems on statistical separability in the perceptron. In *Mechanization of Thought Processes: Proceedings of a Symposium held at the National Physical Laboratory, November 1958. Vol 1.* London: HM Stationery Office.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington D.C.: Spartan.

Rumelhart, D.E., Hinton G.E. & William R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing. Explorations in the Microstructure of Cognition, Vol 1.* Cambridge, MA: MIT Press.

Rumelhart, D.E. & McClelland, J.L. (1986). On learning past tenses of English verbs. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing. Explorations in the Microstructure of Cognition, Vol 2.* Cambridge, MA: MIT Press.

Rumelhart, D.E. & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science,* 9, 75-112.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. Technical Report CU-CS-435-89, Department of Computer Science, University of Colorado at Boulder.

Saund, E. (1987). Dimensionality-reduction using connectionist networks. Technical Report AI-941, Artificial Intelligence Laboratory ,MIT, Cambridge, MA.

Seibel, R. (1963). Discrimination reaction time for a 1,023-alternatives task. *Journal of Experimental Psychology,* 66, 215-226.

Seidenberg, M.S. (1985). The time-course of phonological code activation in two writing systems. *Cognition*, 19, 1-30.

Seidenberg, M.S. & McClelland, J.L. (1989). A distributed developmental model of word recognition and naming. *Psychological Review*, 96, 523-568.

Seidenberg, M.S., Waters, G.S., Sanders, M., & Langer, P. (1984). When does irregular spelling or pronunciation influence word recognition? *Journal of Verbal Learning and Verbal behavior*, 23, 383-404.

Sejnowski, T.J. & Rosenberg, C.R. (1987). Parallel networks that learn to pronounce English text. *Complex Systems*, 1, 145-168.

Shannon, C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379-423.

Shepard, R.N. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22, 325-345.

Shipley, E.F. (1961). *Detection and Recognition with Uncertainty*. Ph.D. Thesis, Department of Psychology, University of Pennsylvania, Philadelphia, PA.

Smith, J.E.K. (1980). Models of identification. In R. Nickerson (Ed.), *Attention and Performance VIII*. Hillsdale, NJ: Erlbaum.

Snoddy, G.S. (1926). Learning and stability. *Journal of Applied Psychology*, 10, 1-36.

Stanovich, K.E., Pachella, R.G. & Smith, J.E.K. (1977). An analysis of confusion errors in naming letters under speed stress. *Perception & Psychophysics*, 21, 545-552.

Swets, J.A. (1959). Indices of signal detectability obtained with various psychophysical procedures. *Journal of the Acoustical Society of America,* 31, 511-513.

Tanner, W.P., Swets, J.A., & Green, D.M. (1956). Some general properties of the hearing mechanism. Technical Report 30, Electronic Defense Group, University of Michigan, Ann Arbor, MI.

Teichner, W.H. & Krebs, M.J. (1974). Laws of visual choice reaction time. *Psychological Review,* 81, 75-98.

Theios, J. (1975). The components of response latency in simple human information processing tasks. In P.M.A. Rabbitt and S. Dornic (Eds.), *Attention and Performance V.* London, England: Academic Press.

Torgeson, W.S. (1958). *Theory and Methods of Scaling.* New York, NY: Wiley.

Townsend, J.T. (1971). Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics,* 9, 40-50.

Townsend, J.T. & Ashby, F.G. (1982). Experimental tests of contemporary models of visual letter recognition. *Journal of Experimental Psychology: Human Perception and Performance,* 8, 834-864.

Townsend, J.T. & Ashby, F.G. (1983). *Stochastic Modeling of Elementary Psychological Processes.* Cambridge, England: Cambridge University Press.

Townsend, J.T. & Landon, D.E. (1982). An experimental and theoretical investigation of the constant-ratio rule and other models of visual letter confusion. *Journal of Mathematical Psychology,* 25, 119-162.

Tversky, A. (1977). Features of similarity. *Psychological Review,* 84, 327-352.

Venezky, R.L. (1970). *The Structure of English Orthography.* The Hague, The Netherlands: Mouton.

Vickers, D. (1972). Some general features of perceptual discrimination. In E.G. Asmussen (Ed.), *Psychological Aspects of Driver Behaviour.* Institute for Road Safety Research, The Netherlands: Voorlung.

Vickers, D. (1979). *Decision Processes in Visual Perception.* New York, NY: Academic Press.

Volper, D.J. & Hampson, S.E. (1987). Learning and using specific instances. *Biological Cybernetics, 56,* 204-228.

Ward, L.M. & Lockhead, G.R. (1970). Sequential effects and memory in category judgments. *Journal of Experimental Psychology,* 84, 27-34.

Waters, G.S. & Seidenberg, M.S. (1985). Spelling-sound effects in reading: Time-course and decision criteria. *Memory & Cognition, 6,* 557-572.

Welford, A.T. (1960). The measurement of sensory-motor performance. *Ergonomics,* 3, 189-230.

Welford, A.T. (1968). *Fundamentals of Skill.* London, England: Methuen.

Widrow, G. & Hoff, M.E. (1960). Adaptative switching circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4,* 96-104.

Wijk, A. (1966). *Rules of Pronunciation for the English Language.* Oxford, England: Oxford University Press.

APPENDIX 1

ZIP_NET A NEURAL NETWORK SIMULATOR.

A1.1. Introduction.

Zip_net is a fast (up to 30 000 weight updates per second on a Sun 3), easy to use, "batch" neural network simulator. It was developed to provide a convenient way to perform a series of simulations while varying one or several simulation parameters. Zip_net is a three layer feed-forward back-propagation (and mean-variance back-propagation) network simulator. It allows the use of a real time decoding module to build hybrid architectures. The software allows the user to periodically save the network weights, output and hidden vectors as well as performance indicators such as mean-square error, probability correct, and variance. Additionally, it provides a convenient way to restart a crashed simulation from partial results and powerful trace facilities for debugging. Zip_net does not provided any fancy graphics (as do Neuralware or RSC) and its scope of application is limited (only three layer feed-forward networks). Nonetheless it has the advantage that a simulation can be set up rapidly without complicated declaration and without recompilation.

A1.2. History.

In October 1987 I started developing software to simulate feed-forward networks and the back-propagation learning algorithm. I used C on a SUN-2 computer. The program was

modified and recompiled for each new simulation. To make the modification easier the basic parameters were implemented using <#define> macros. By January 1988 the software had been largely rewritten, optimized and ported to a SUN-3: Zip_net version 1.0 was born. It was first used for simulation of large networks implementing a model of word recognition and naming. The simulator proved to be fast and reliable but the user interface was clumsy, and in case of a system crash restarting a run from partial results was a difficult task. It became a critical problem since a single run of the word recognition model could take several days. As the complexity and number of simulations I ran increased it became clear that I needed a more convenient and reliable tool.

In October 1988 I started developing a better user interface, implemented a small parser, added dynamic allocation of memory and corrected some underflow problems. After several updates of version 1, Zip_net version 2.0 was completed in the beginning of 1989. From then on, there was no need to recompile the program to run new simulations. The simulation parameters were defined in an input file through simple keywords and parameters. The format of the definition file is inspired by those used in popular statistical packages. In the following months I started distributing the simulator to students and researchers. Since then, the simulator has had minor updates. Zip_net Version 2.1c is the latest update.

A major update is under preparation. Version 3.0, which should be available by Summer 1990, will have a more homogeneous presentation (as it is the output is bilingual English/French), will allow the simulation of arbitrary connected networks (including recurrent nets) and will have extended graphic capabilities. Also, some problems that have been reported with the parser will be corrected and a complete manual will be provided.

A1.3. Setting the run.

The following is an example task definition to run three simulations (of the encoder problem) using different numbers of hidden units over the same stimuli set:

```
*= example de simulation
*= avec Zip_net
TITLE = example de trois simulations...
NIU   = 10
NOU   = 10
NHU   = 1
EPOCHN= 1000
REPORT= 100
SAVE  = 500
STIMF = encoder10
EROUT = output1
NETOUT= net1
PRINT = 2
+=
NHU   = 2
EROUT = output2
NETOUT= net2
+=
NHU   = 2
EROUT = output2
NETOUT= net2
```

The syntax is [keyword]=[parameter]↵. Blanks are optional. Parameters can be specified in any order. The keywords used here have the following meanings:

*	comment line, skipped
TITLE	title of the run
NIU	number of input units
NOU	number of output units
NHU	number of hidden units
EPOCHN	number of epochs to be performed
REPORT	interval at which the network status is reported
SAVE	interval at which the network structure is saved
STIMF	name of the file containing the stimuli and target vectors
EROUT	file used to save goodness of fit indicators when reports are done.
NETOUT	file used to save the network structure (weighted links)
PRINT	amount of information printed on the output listing. 0 is minimum 5 is maximum details
+=	new task: all previous parameters are reused except for the ones redefined.

Other parameters needed for the simulation such as the back-propagation learning rate have default values. The input file starts a first simulation with a network of one hidden, 10 input and 10 output units. The number of hidden units and the output files used are redefined for the second and third task. All other parameters keep the same value. This example input file required PRINTlevel=2. It generated the following output:

```
*****************************************************************************
*                                                                           *
*                  ZIP_NET version 2.1c by Yves Lacouture 1989              *
*                              McGill University                            *
*                                                                           *
*****************************************************************************

       parse: this is task # 1:
       parse: les variables sont::
   0                   NOU = 10
   1                   NHU = 1
   2                   NIU = 10
   3                   MAX = 0
   4                 TRACE = 0
   5                EPOCHN = 1000
   6                EPOCHS = 1
   7                REPORT = 100
   8                  SAVE = 500
   9                NETOUT = net1
  10                 NETIN =
  11                 STIMF = encoder10
  12                 TITLE = example de trois simulations...
  13                 EROUT = output1
  14                TEMPER = 0.45
  15                 ALPHA = 0.0
  16                 NOISE = 0.0
  17              FILLRAND = 0.5
  18                     * =
  19                SMOUTH = 0
  20                   NEW =
  21            PRINTLEVEL = 2
  22             RESETFILE = 1
  23                 BSBIN =
  24                FFORCE = 1.0
  25                 GAMMA = 1.0
  26              SAVEOUT = save0.out
  27            SAVEHIDDEN =
  28            WINNERGAIN = 0
  29                MAXBSB = 0
  30            RUNNERGAIN = 0.0
  31                   ERD = detail_out1
  32                 MAXWC = 500
  33                  SEED = 0
  34                LAMBDA = 0
  35                 DELTA = 0
  36              STREPORT = 0
  37               PROBOUT =
  38                STPROB = 0
  39              SMOUTH_O = 0
  40           STARTSAVEOUT = 0
  41                 LEARN = 1
  42              LOADREAL = 0
  43                GAININ = 1


*****************************************************************************
*                                                                           *
*                       example de trois simulations...                     *
*                                                                           *
*****************************************************************************
      fillrnd: the range is [-0.5000 -- 0.5000]
 loadstimuli: loaded 20 stimuli from file encoder10
        main: error[100]= 0.289609
        main: error[200]= 0.280941
```

```
        main: error[300]= 0.266635
        main: error[400]= 0.257478
        main: error[500]= 0.252269
     savenet: reseting file net1
        main: error[600]= 0.248932
        main: error[700]= 0.246470
        main: error[800]= 0.244480
        main: error[900]= 0.242804
        main: error[1000]= 0.241390
     savenet: reseting file net1
        main: closing save0.out
        main: task done required 59 secondes (real time)
            : (~ 8474.58 weigth update per sec.)

*******************************************************************************
*                                                                             *
*                  ZIP_NET version 2.1c by Yves Lacouture 1989                *
*                              McGill University                              *
*                                                                             *
*******************************************************************************

       parse: this is task # 2:
.
.
```

description of task #2 and #3

```
.
.
     savenet: reseting file net3
        main: closing save0.out
        main: task done required 82 secondes (real time)
            : (~ 11345.9 weigth update per sec.)
        main: end of input file
        main: l'ensemble des taches a requis 229 secondes (temps reel)
        main: SORTIE NORMALE
```

The first part of the listing is the values set or assumed by default for the 43 keywords used by the simulator. As demonstrated by this example run not all parameters need to be defined. The second part of the listing presents a progress report as the simulation is performed.

Zip_net being a large structured program, the procedure issuing the message is always mentioned. This allows better control of the simulated process and makes it easier to find and correct bugs. As can be seen numerous things happen after the procedure *parse* has set up the task parameters. First,

procedure *fillrnd* allocates random values to all weighted connections in the specified net. Unless specified in the description file (e.g. FILLRND=0.1), this procedure uses ±0.5 bounds. If NETIN=<filename> is specified, then instead of being assigned random values, the weights are loaded from <filename> by procedure loadnet.

Then the stimulus/response pairs are loaded. The line STIMF=encoder10 in the task description file specified that these are loaded from the file encoder10. The format is assumed to be binary (%1d) using alternative lines for stimulus/response vectors. The file encoder10 has the following structure:

```
100000000
100000000
010000000
010000000
001000000
000100000
000100000
000010000
000010000
000001000
000001000
...
```

This input file could also be free format, in which case LOADREAL=1 must be specified in the task definition. The procedure loadstimuli reports that 20 vectors (10 stimulus/response pairs) are loaded.

Once the network has been set up, the learning process starts unless learn=0 is specified. In that case no weight

modifications occur. The number of epochs to be performed in this case is EPOCHN=1000. Each epoch consist of the whole stimulus set unless the keyword EPOCHS is specified (e.g. EPOCHS=5), in which case a subset of the stimuli is randomly selected within each epoch.

The keyword REPORT=100 specifies that progress information is to be reported every 100 epochs. When reports are performed global performance indicators (e.g. MSE, variance...) are saved in the file specified by the keyword EROUT (e.g. EROUT=out1). At the same time detailed performance indicators for individual stimuli are saved in the file specified by ERD=<filename>. If no filename is supplied a default name is used. The printed output shows MSE computed over the epoch when a report is done. If STREPORT=n (start report) is specified then the report will not be done until the number of epochs performed has exceeded this value, e.g. with STREPORT=400 no reports are performed unless at least 400 epochs were performed.

Other keywords allow the user to save output vectors (SAVEOUT) or hidden vectors (SAVEHIDDEN) in specified files. Similarly, the keyword PROBOUT allows the user to save the stimulus/response array (based on the maximum response of the output units) in the specified file. The keyword STPROB (start probability) specifies the number of epochs to be performed before the matrix is recorded.

The line SAVE=500 indicates that the weighted connections of the network are saved after each 500 learning epochs. The declaration NETOUT=net1 indicates that network weighted links values are to be save into file "net1". If the file exists it is overwriten unless RESETFILE=0 is specified, then the weight structure is happened at the end of the file. The output format is the same as the one used by the program to load a network structure when the keyword NETIN is specified. This allows easy use of previously learned networks. Each time the net is save either

```
savenet: reseting file net1
```

or

```
savenet: happen to file net1
```

is reported depending on whether the file is overwritten or not. If NEW=1 is specified, every time the network is saved different files are used. A suffix indicating the epoch number is automatically added to the output filename (e.g. net1.500, net1.1000, ...).

At the end of the task, the time required and a simulation performance indicator are reported. Then either the program exits or if the string "+=" is encountered in the input file, then another task is setup by the parser. For the new task all previously defined parameters are reused unless changes are specified. When all the tasks have been completed, the exit condition and the overall running time are reported.

A1.4. Additional controls and options.

Although Zip_net allows the users to load any real valued vector there might be situations where alterations of the vectors should be done once the vectors are loaded.

Adding noise. The most useful alteration is to add white noise to the input vector. This is done dynamically for each stimulus presentation. The keyword used is NOISE and the numerical value supplied is the approximate percentage of noise (length of noise vector divided by length of input vector) added (e.g. NOISE=0.20, add 20% noise).

Gaussian filters. If the input or target vectors are binary encoded, than either of them can be filtered in order to present a smooth normal shaped distribution. Keywords are SMOUTHI and SMOUTHO to respectively filter the input and target vectors. The numerical value supplied corresponds to half the total range of the gaussian distribution. The total area under the Gaussian curve is kept equal to 1.

Gain on the input vector. Because the input vectors are often imported from other applications, it is sometime useful to scale the original vector. The real value provided through the keyword GAININ allows this - e.g. if GAININ=0.5 is specified the input vector is simply multiplied by 0.5.

The specific models I developed required the use of a hybrid architecture where a decoding module is connected at the

output of the feed-forward network. Zip_net allows the use of three types of decoding modules: Anderson's Brain-State-in-a-Box, Koch-Ullmann's Winner-Take-All network, and a network of simple integrators with thresholds. No one or several decoding modules can be used parallely. This is useful when decoding times are to be compared. For all three modules the decoding time is supplied in the EROUT and ERD file when reports are provided.

The BSB decoding module. Although the program supports the BSB decoding module it does not allow modification of the BSB weighted links. Learning must first be done using other software. The keyword BSBIN=<filename> is used to specify the name of the file containing the values of the BSB weighted links. The weighted links must be written in free format. An upper limit to the maximum number of iterations performed by the BSB is set through MAXBSB. For example, if MAXBSB=100 is specified, the decoding process will be stopped after 100 iterations. This is useful to prevent the endless loop encountered with stationary state vectors that do not converge. In that case the reaction time provided is MAXBSB.

The main diagonal of the connection matrix **A+I** can be scaled, the keyword being FFORCE (feed-back force); when FFORCE=0 the main diagonal values are set to zero. The output vector from the feed-forward net can also be scaled – this is done through the use of the keyword GAMMA, with default value 1.0 meaning no scaling.

Winner-Take-All decoding module. This module is easily activated through the keyword WINNERGAIN. If WINNERGAIN=0 (default) the module is not used. The maximum number of decoding cycles is set through MAXWC.

Integrators with thresholds. The declaration RUNNERGAIN=<n> activates the decoding network of integrators with threshold. The value supplied fixes the gain of the feedback. Again if RUNNERGAIN=0 (default) the module is not activated.

Control over the learning rate. Zip_net implements the standard back-propagation as well as my mean-variance-back-propagation. Step size relative to mean-squares error and variance of error are respectively defined through the keywords TEMPER and LAMBDA. If LAMBDA=0 standard back-propagation is used. Default values are TEMPER=0.45 and LAMBDA=0.

Resetting the random number generator. Zip_net's random number generator uses the time of day (in seconds) to initiate the pseudo-random sequence. On some occasions it might be useful to run several simulations with the same starting random weights and same noise vectors. This can be easily done if, for all simulations, the random number generator is reset with the same seed. The keyword SEED=<n> specifies that the integer n is used to reset the generator. If SEED=0, time of the day is used instead.

Control over the printed output.

Three keywords allow the user to modify the printout: TITLE=<str>, PRINT=<n>, and TRACE=<n>. TITLE puts the specified string (in a box) at the beginning of the task output. PRINT (PRINTlevel) controls the amount of information provided on the listing from 0 (minimum printout) to 4 (maximum printout details). For debugging purposes the TRACE parameter provides 5 levels of tracing (0 minimum, 4 maximum). Note that the trace facility can generate a lot of output.

A1.5. Description of the control language.

*	Comment line: skipped. Default=NULL
ALPHA	Real number setting the learning rate associated with the mean-square error gradient of back propagation. Default=0.45
BSBIN	File name of the BSB weighted links structure. If non-null the BSB connections are loaded and BSB decoding is performed. Default=NULL
EPOCHN	Number of epochs to perform. Default=1
EPOCHS	Number of stimuli presented in each epoch. If EPOCHS=0 the whole stimuli set is presented. Default=0
ERD	File name, where the detailed performance indicators are outputted. Default=NULL
EROUT	File name where the global performance indicators are outputted. Default="ERout"
FFORCE	Real number setting the strength of the feed back force used for the BSB. Default=1.0
FILLRAND	Real number setting the bounds of the random number used as starting values for the weighted links. Not used if the parameter NETIN=<filename> is specified. Default=0.5

GAININ	Real number setting the gain factor applied to the input vector. Default=1.0
GAMMA	Real number setting the gain at the input to the BSB decoding module. Default=1.0
LAMBDA	Real number setting the learning rate associated with the variance gradient of the minimum variance back-propagation. If LAMBDA=0 standard BP is used. Default=0
LEARN	Integer switching on and off the learning process: 0=off, 1=on. Default=1.
LOADREAL	Integer specifying the type of stimulus vectors loaded. 0=binary packed, 1= free format. Default=0
MAXBSB	Integer specifying the maximum number of iterations performed by the BSB decoding module. Default=500
MAXWC	Integer specifying the maximum number of iterations performed by the winner-take-all net. Default=100
NETIN	File name indicating where the network weighted links are to be loaded from. Default=NULL
NETOUT	File name indicating where the network weighted links are to be saved. Default=NETout
NEW	Integer specifying if new files are to be used each time the weighted links are saved. 0=same filename, 1=new filename. Default=0
NHU	Integer number of hidden units.
NIU	Integer number of input units.
NOISE	Real number: percentage of noise dynamically added to the input vector at each stimuli presentation. Default=0
NOU	Integer number of output units.
PRINT	Integer setting the detail level in the printed output. 0=minimum, 4=maximum. Default=1
PROBOUT	File name use to save the stimulus/responses incidence matrix. Default="PROBout"

REPORT	Integer number indicating the interval of reports. Default=50
RESETFILE	Integer number indicating whether NETOUT is overwritten. 0= data happened, 1=overwritten. Default=1
RUNNERGAIN	Real number that sets the gain of the decoding network of integrators with threshold. For RUNNERGAIN=0 this decoding device is not used. Default=0
SAVE	Integer number indicating the interval for saving the network weights. If SAVE=0 no savings are done. Default=0
SAVEHIDDEN	File name indicating where the network hidden vectors are to be saved. If not specified the hidden vectors are not saved. Default=NULL, e.g. not saved
SAVEOUT	File name indicating where the network output vectors are to be saved. If not specified the output vectors are not saved. Default=NULL, e.g. not saved
SEED	Integer used to reset the random number generator.[If SEED=0 (Default) time of day is used.
SMOUTHI	Integer value specifying the range of the Gaussian filter applied to the input vectors. If SMOUTHI=0 no filtering is done. Default=0
SMOUTHO	Integer value specifying the range of the Gaussian filter applied to the target vectors. If SMOUTHO=0 no filtering is done. Default=0
STARTSAVEOUT	Integer number of epochs indicating when to start to save the output vector. Default=1
STIMF	File name indicating where the stimuli are to be loaded from. Default=NULL
STPROB	Integer number of epochs indicating when to start to compute the stimulus/response matrix Default=1
STREPORT	Integer number of epochs indicating when to start to perform reports. Default=1
TEMPER	Real number. Same as ALPHA. Default=0.45 (this is for historical reasons)

TITLE	String delimited by the end-of-line character, indicates the title of the task.
TRACE	Integer setting the amount of detail provided by the debugging trace facility. 0=trace off, 4=maximum . Default=0
WINNERGAIN	Real number setting the gain of the winner-take-all decoding network. For WINNERGAIN=0 this decoding device is not used. Default=0

The 15 most commonly encountered error messages:

```
1     "     getfile: file <> not found"
2     " loadstimuli: loading in real mode...binary file found"
3     " loadstimuli: loading in binary mode...float file found"
4     " loadstimuli: odd number of stimuli",
5     "     loadnet: _NIU_ does not match declaration"
6     "     loadnet: _NHU_ does not match declaration"
7     "     loadnet: _NOU_ does not match declaration"
8     "     loadnet: unexpected EOF"
9     "        main: unexpected EOF"
10    "        main: parameters exceed maximum declaration"
11    "        main: out of memory"
12    "       parse: Error in parsing line <>: unknown keyword |<>|"
13    "       parse: unexpected EOF"
14    "       <fct>: SORTIE en ERREUR(<code>)"
15    "       <fct>: undocumented error"
```

APPENDIX 2

LIST OF TEST WORDS

	Regular	Irregular
Low Frequency	MODE	DEAF
	DOCK	WORM
	PEST	PHASE
	HIKE	PLAID
	MATH	TOMB
	GREED	SOOT
	CHORE	WAND
	GRILL	SEW
	BAKES	WAN
	FERN	CASTE
	TILE	STEAK
	RUST	GROSS
High Frequency	STILL	GIVE
	FEEL	SAYS
	THIN	BREAK
	CORN	TOUCH
	NINE	LOSE
	RACE	CHOOS
	LEAST	WATCH
	FACE	HEARD
	WAKE	DOLL
	THESE	SOME
	BEACH	WOOL
	SHELL	WASH

APPENDIX 3
TRAINING SET FOR
THE WORD RECOGNITION MODEL

A computerized version of this lexicon including relative word frequency and corresponding phonemy is available from the author.

A	ALFRED	ARTICLE	BEAM	BLOW	BUILT
ABILITY	ALIVE	ARTIST	BEAR	BLUE	BUREAU
ABLE	ALL	ARTS	BEAT	BOARD	BURNED
ABOUT	ALLOW	AS	BEAUTY	BOARDS	BURNING
ABOVE	ALMOST	ASIDE	BECAUSE	BOAT	BURST
ABROAD	ALONE	ASK	BECOME	BOATS	BUS
ABSENCE	ALONG	ASKING	BECOMES	BOB	BUSH
ACCEPT	ALREADY	ASPECT	BED	BODIES	BUSY
ACCOUNT	ALSO	ASPECTS	BEDROOM	BODY	BUT
ACHIEVE	ALWAYS	ASSUME	BEEN	BOMB	BUY
ACRES	AM	ASSURE	BEER	BOND	BY
ACROSS	AMERICA	ASSURED	BEFORE	BONDS	CALL
ACT	AMONG	AT	BEGAN	BONE	CALLING
ACTING	AMOUNT	ATOM	BEGIN	BOOK	CALLS
ACTION	AMOUNTS	ATOMIC	BEGINS	BOOKS	CALM
ACTIONS	AN	ATOMS	BEGUN	BORN	CAMERA
ACTIVE	ANCIENT	ATTACK	BEHIND	BOTH	CAMP
ACTS	AND	ATTEMPT	BEING	BOTTLE	CAN
ACTUAL	ANGELES	ATTEND	BEINGS	BOTTOM	CANE
ADAM	ANGER	AUGUST	BELEIF	BOUGHT	CANNOT
ADAMS	ANGLE	AUTHOR	BELEIVE	BOUND	CAPABLE
ADD	ANGRY	AVENUE	BELONG	BOWL	CAPTAIN
ADDRESS	ANIMAL	AVERAGE	BELOW	BOX	CAR
ADMIT	ANIMALS	AVOID	BENCH	BOY	CARE
ADVANCE	ANNUAL	AWARD	BENEATH	BOYS	CAREER
ADVICE	ANODE	AWARE	BENEFIT	BRAIN	CAREFUL
ADVISED	ANOTHER	AWAY	BENT	BRANCH	CARRY
AFFAIR	ANSWER	AXIS	BESIDE	BREAD	CARS
AFFAIRS	ANSWERS	BABY	BEST	BREAK	CASE
AFFECT	ANY	BACK	BETTER	BREAK	CASH
AFFORD	ANYONE	BAD	BETWEEN	BREATH	CAST
AFRAID	ANYWAY	BADLY	BEYOUND	BRIDE	CASTE
AFRICA	APART	BAG	BIBLE	BRIDGE	CATCH
AFTER	APPEAL	BAKER	BIG	BRIEF	CAUSE
AGAIN	APPEAR	BAKES	BIGGER	BRIEFLY	CELL
AGAINST	APPEARS	BALANCE	BILL	BRIGHT	CELLS
AGE	APPLY	BALL	BILLION	BRING	CENT
AGENCY	APRIL	BALLET	BILLS	BRISK	CENTER
AGENT	ARC	BAND	BIRD	BRITISH	CENTERS
AGENTS	AREA	BANK	BIRDS	BROAD	CENTRAL
AGES	AREAS	BANKS	BIRTH	BROKE	CENTURY
AGREE	ARM	BAR	BIT	BROKEN	CHAIN
AHEAD	ARMED	BARS	BITTER	BROOD	CHAIR
AID	ARMS	BASE	BLACK	BROTHER	CHAMBER
AIM	ARMY	BASIC	BLAME	BROUGHT	CHANGE
AIR	AROUND	BATTLE	BLIND	BROWN	CHAPTER
AISLE	ARRIVED	BAY	BLOCK	BRUSH	CHARGE
AISLE	ART	BE	BLOCKS	BUDGET	CHARTER
ALERT	ARTERY	BEACH	BLOOD	BUILD	CHECK

CHEST	COMPANY	DANCING	DOCK	EGGES	FAILURE
CHICKEN	CONCEPT	DANGER	DOCTOR	EIGHT	FAIR
CHIEF	CONCERN	DARE	DOG	EITHER	FAIRLY
CHILD	CONCERT	DARK	DOGS	ELEMENT	FAITH
CHINA	CONDUCT	DATA	DOING	ELEVEN	FALL
CHINESE	CONTACT	DATE	DOLL	ELSE	FALLEN
CHOICE	CONTAIN	DAY	DOLLAR	EMOTION	FALLING
CHOIR	CONTENT	DAYS	DOLLARS	EMPTY	FAMILY
CHOOSE	CONTEXT	DEAD	DONE	END	FAMOUS
CHORE	CONTROL	DEAF	DOOR	ENDS	FAR
CHRIST	COOK	DEAL	DOORS	ENEMY	FARM
CHURCH	COOL	DEALING	DOTS	ENERGY	FARMERS
CHUTE	COOLING	DEAN	DOUBLE	ENGINE	FASHION
CIRCLE	COPE	DEAR	DOUBT	ENGLISH	FAST
CITIES	COPY	DEATH	DOWN	ENJOY	FAT
CITY	CORD	DEBT	DOZEN	ENOUGH	FATE
CIVIL	CORE	DECADE	DRAMA	ENTER	FATHER
CLAIM	CORN	DECADES	DRAW	ENTIRE	FAWN
CLAIMED	CORNER	DECIDE	DRAWING	EQUAL	FEAR
CLAIMS	CORPS	DEED	DRAWL	EQUALLY	FEARS
CLANG	CORRECT	DEEP	DREAM	ERROR	FEATURE
CLASS	COST	DEEPER	DRESS	ERRORS	FED
CLASSIC	COSTS	DEEPLY	DRILL	ESCAPE	FEDERAL
CLAY	COTTON	DEFENSE	DRINK	ESTATE	FEED
CLEAN	COULD	DEFINED	DRIVE	EUROPE	FEEL
CLEAN	COUNCIL	DEGREE	DRIVER	EVEN	FEELING
CLEAR	COUNT	DEMAND	DRIVING	EVENING	FEELS
CLEARLY	COUNTRY	DEMANDS	DROP	EVENT	FEET
CLERCK	COUNTY	DENY	DROVE	EVENTS	FELL
CLIMB	COUPLE	DEPEND	DRUNK	EVER	FELLOW
CLOSE	COURSE	DEPENDS	DRY	EVERY	FELT
CLOSELY	COURT	DEPTH	DUE	EVIDENT	FEMALE
CLOSER	COURTS	DERIVED	DURING	EVIL	FERN
CLOTH	COUSIN	DESIGN	DUST	EXACTLY	FEW
CLOTHES	COVE	DESIRE	DUTIES	EXAMINE	FEWER
CLOUDS	COVER	DESIRED	DUTY	EXCEPT	FICTION
CLUB	CRAMP	DESK	DYING	EXCESS	FIELD
CO	CRAZY	DESPITE	DYKE	EXIST	FIELDS
COAST	CREATE	DESTROY	EACH	EXISTS	FIFTEEN
COAT	CREDIT	DETAIL	EARLIER	EXPECT	FIFTY
COATING	CREW	DETAILS	EARLY	EXPENSE	FIG
CODE	CRIME	DEVELOP	EARS	EXPLAIN	FIGHT
COFFEE	CRISIS	DEVICE	EARTH	EXPOSED	FIGURE
COLD	CROSS	DID	EASE	EXPRESS	FIGURES
COLLEGE	CROWD	DIE	EASILY	EXTENT	FILE
COLONEL	CRY	DIED	EAST	EXTRA	FILL
COLOR	CUBA	DIGNITY	EASTER	EXTREME	FILLING
COLORS	CULTURE	DIME	EASY	EYE	FILM
COLUMN	CUP	DINNER	EAT	EYES	FINAL
COLUMNS	CURIOUS	DIRECT	ECONOMY	FACE	FINALLY
COME	CURRENT	DIRT	EDGE	FACING	FIND
COMEDY	CURVE	DIRTY	EDITION	FACT	FINDING
COMES	CUT	DISEASE	EDITOR	FACTOR	FINDS
COMFORT	CUTTING	DISPLAY	EFFECT	FACTORS	FINE
COMING	DAILY	DISPUTE	EFFECTS	FACTS	FINGER
COMMAND	DAMAGE	DISTANT	EFFORT	FACULTY	FINGERS
COMMENT	DAMN	DIVINE	EFFORTS	FADE	FINISH
COMMON	DANCE	DO	EGG	FAIL	FIRE

FIRM
FIRMLY
FIRMS
FIRST
FISCAL
FISH
FIT
FIVE
FLAT
FLESH
FLEW
FLIGHT
FLOAT
FLOOD
FLOOR
FLOW
FLOWERS
FLY
FLYING
FOAM
FOCUS
FOLK
FOLLOW
FOLLOWS
FOOD
FOODS
FOOT
FOR
FORCE
FOREIGN
FOREVER
FORGET
FORM
FORMAL
FORMER
FORMS
FORT
FORTH
FORTY
FORWARD
FOUND
FOUR
FOURTH
FOWL
FRAME
FRANK
FREE
FREEDOM
FREIND
FREINDS
FRENCH
FRESH
FRIDAY
FROM
FRONT
FRUIT
FULL
FULLY
FUN
FUND
FUNDS
FUNNY
FURTHER
FUTURE
GAIN
GAINED
GAME
GAMES
GARDEN
GAS
GATE
GAUGE
GAVE
GENERAL
GERMAN
GERMANY
GET
GETS
GETTING
GHOST
GIFT
GIRL
GIRLS
GIVE
GIVEN
GIVES
GIVING
GLAD
GLANCE
GLASS
GO
GOAL
GOALS
GOD
GOES
GOING
GOLD
GOLDEN
GOLF
GONE
GOOD
GOODS
GRACE
GRADE
GRAND
GRANT
GRAPE
GRASS
GRAVE
GRAY
GREAT
GREATER
GREATLY
GREED
GREEK
GREEN
GREW
GRILL
GROSS
GROSS
GROUND
GROUNDS
GROUP
GROUPS
GROW
GROWING
GROWN
GROWTH
GUARD
GUESS
GUEST
GUESTS
GUIDE
GUILE
GUILT
GULL
GUN
GUNS
GUY
HAD
HAIR
HALF
HALL
HAND
HANDLE
HANDS
HANS
HAPPEN
HAPPENS
HAPPY
HARBOR
HARD
HARDLY
HARM
HARMONY
HARRY
HAS
HAT
HATE
HAVE
HAVING
HE
HEAD
HEADS
HEALTH
HEALTHY
HEAR
HEARD
HEARING
HEART
HEAT
HEAVEN
HEIGHT
HEIR
HELD
HELL
HELP
HELPING
HENCE
HER
HERE
HERE
HERO
HERSELF
HIDE
HIGH
HIGHER
HIGHLY
HIGHWAY
HIKE
HILL
HILLS
HIM
HIMSELF
HIS
HISTORY
HIT
HOE
HOLD
HOLDING
HOLDS
HOLE
HOLES
HOLY
HOME
HOMES
HONEST
HONOR
HOPE
HOPES
HORSE
HOST
HOT
HOTEL
HOUR
HOURS
HOUSE
HOUSES
HOUSING
HOW
HOWEVER
HUGE
HUMAN
HUMOR
HUNDRED
HUNG
HUNTING
HURRY
HURT
HUSBAND
I
ICE
IDEA
IDEAL
IF
ILL
IMAGE
IMAGES
IMAGINE
IMPACT
IMPROVE
IN
INCH
INCLUDE
INCOME
INDEED
INDEX
INDIA
INDIAN
INITIAL
INNER
INSIDE
INSTANT
INSTEAD
INTENSE
INTO
IRON
IS
ISLAND
ISSUE
IT
ITALIAN
ITALY
ITEM
ITEMS
ITS
ITSELF
JACK
JACKET
JAPAN
JAZZ
JESS
JEWS
JIM
JOB
JOBS
JOHN
JOIN
JOINT
JOURNAL
JOY
JUDGE
JULY
JUNE
JUNIOR
JURY
JUST
JUSTICE
KEEP
KEEPING
KEPT
KEY
KEYS
KID
KILL
KILLED
KIND
KINDS
KING
KITCHEN
KNEE
KNEES
KNEW
KNIFE
KNOW
KNOWING
KNOWN
KNOWS
LABOR
LACK
LADY
LAID
LAKE
LAND
LARGE
LARGELY
LARGER
LASH
LAST
LATE
LATER
LATIN
LATTER
LAW
LAWS
LAWYER
LAY
LEAD
LEADER
LEADERS
LEADING
LEADS
LEAF
LEAGUE
LEANED
LEARN
LEAST
LEAVE
LEAVES
LEAVING
LED
LEE
LEFT
LEG
LEGAL
LEGS
LENGTH
LESS

LET
LETTER
LETTERS
LEVEL
LEVELS
LIBERAL
LIBERTY
LIBRARY
LICENSE
LIE
LIES
LIFE
LIGHT
LIGHTS
LIKE
LIKELY
LIMIT
LIMITS
LINE
LINES
LIPS
LIQUID
LIQUOR
LIST
LISTEN
LISTS
LITTLE
LIVE
LIVES
LIVING
LOAD
LOAN
LOCAL
LOCATED
LOGICAL
LONG
LONGER
LOOK
LOOKING
LOOKS
LOOM
LOOSE
LORD
LOSE
LOSS
LOST
LOT
LOTS
LOVE
LOVELY
LOW
LUCK
LUMBER
LUNCH
LYING
LYMPH
MACHINE
MAD
MADE
MAGIC
MAIL
MAIN
MAJOR
MAKE
MAKES
MAKING
MALE
MAMA
MAN
MANAGER
MANKIND
MANNER
MANTLE
MANY
MARCH
MARINE
MARK
MARKED
MARKET
MARTIN
MARY
MASSIVE
MASTER
MATCH
MATH
MATTER
MATTERS
MAY
MAYBE
MAYOR
ME
MEAN
MEANING
MEANS
MEANT
MEASURE
MEAT
MEDICAL
MEDIUM
MEET
MEETING
MEETS
MEMBER
MEMBERS
MEMORY
MEN
MENTAL
MENTION
MERCER
MERE
MERELY
MESSAGE
MET
METAL
METHOD
METHODS
MIDDLE
MIGHT
MIKE
MILE
MILES
MILK
MILLION
MIND
MINDS
MINE
MINIMUM
MINOR
MINUTE
MINUTES
MISS
MISSILE
MISSING
MISSION
MIST
MISTAKE
MIX
MOBILE
MODE
MODEL
MODELS
MODERN
MOIST
MOLD
MOLE
MOMENT
MOMENTS
MONDAY
MONEY
MONTH
MONTHS
MOOD
MOON
MOOSE
MORAL
MORE
MORNING
MOST
MOSTLY
MOTH
MOTHER
MOTION
MOTOR
MOTORS
MOUSE
MOUTH
MOVE
MOVES
MOVING
MUCH
MURDER
MUSCLE
MUSH
MUSIC
MUST
MY
MYSELF
MYSTERY
MYTH
NAME
NAMELY
NAMES
NARROW
NATION
NATIONS
NATIVE
NATURAL
NATURE
NAVAL
NAVY
NEAR
NEARBY
NEARLY
NECK
NEED
NEEDS
NEGRO
NEGROES
NEITHER
NET
NEUTRAL
NEVER
NEW
NEWS
NEXT
NICE
NIECE
NIGHT
NIGHTS
NINE
NO
NOBODY
NODDED
NOISE
NONE
NOR
NORMAL
NORTH
NOSE
NOT
NOTE
NOTES
NOTHING
NOTICE
NOTION
NOVEL
NOW
NUMBER
NUMBERS
OBJECT
OBJECTS
OBTAIN
OBVIOUS
OCCUR
OCEAN
OCTOBER
ODD
OF
OFF
OFFER
OFFERS
OFFICE
OFFICER
OFTEN
OIL
OLD
OLDER
ON
ONCE
ONE
ONES
ONLY
ONSET
ONTO
OPEN
OPENED
OPENING
OPENLY
OPERA
OPERATE
OPINION
OR
ORDER
ORDERS
ORGANIC
ORIGIN
OTHER
OTHERS
OUCH
OUGHT
OUR
OUT
OUTLOOK
OUTPUT
OUTSIDE
OVER
OVERALL
OWN
OWNER
OWNERS
OXYGEN
PACE
PAGE
PAID
PAIL
PAIN
PAINT
PAIR
PALACE
PALE
PALMER
PANELS
PAPA
PAPER
PAPERS
PARENTS
PARIS
PARK
PARKER
PART
PARTIES
PARTLY
PARTS
PARTY
PASS
PASSAGE
PASSED
PASSING
PAST
PAT
PATENT
PATH
PATIENT
PATTERN
PAY
PAYMENT
PEACE
PEACH
PEAR
PEEL
PEEP
PENCIL
PEOPLE
PEOPLES
PER
PERCENT
PERFECT
PERHAPS
PERIOD
PERIODS
PERMIT
PERSON
PERSONS
PEST
PETER
PHASE
PHIL
PHONE
PHRASE
PIANO
PICK
PICTURE
PIECE
PIKE
PILOT
PINK
PINT
PITCH

PLACE	PROMISE	RECORDS	ROUSE	SEIZE	SIEVE
PLAID	PROPER	RECTOR	ROUTE	SELDOM	SIGHT
PLAIN	PROTECT	RED	ROUTINE	SELF	SIGN
PLAN	PROUD	REEF	ROW	SELL	SIGNAL
PLANE	PROVE	REGARD	ROY	SENATE	SIGNS
PLANS	PROVIDE	REGION	ROYAL	SENATOR	SILENCE
PLANT	PUBLIC	REGIONS	RULE	SEND	SILENT
PLANTS	PULL	RELATED	RULES	SENDING	SIMILAR
PLATE	PUMP	RELEASE	RUN	SENIOR	SIMPLE
PLAY	PURE	RELIEF	RUNNING	SENSE	SIMPLY
PLAYER	PURPOSE	REMAIN	RUNS	SENT	SIN
PLAYING	PUSH	REMAINS	RURAL	SEPT	SINCE
PLAYS	PUT	REMARKS	RUSSIAN	SERIES	SING
PLEASE	PUTTING	REMOVAL	RUST	SERIOUS	SINGING
PLENTY	QUALITY	REMOVE	SACRED	SERVE	SINGLE
PLOT	QUARTER	REPORT	SAD	SERVES	SIR
PLOW	QUEEN	REPORTS	SAFE	SERVICE	SISTER
PLUS	QUICK	REQUEST	SAFETY	SERVING	SIT
POCKET	QUICKLY	REQUIRE	SAID	SESSION	SITE
POEM	QUIET	RESERVE	SAKE	SET	SITTING
POEMS	QUIETLY	RESPECT	SALARY	SETS	SIX
POET	QUITE	REST	SALE	SETTING	SIZE
POINT	RACE	RESULT	SALES	SETTLES	SKILLS
POINTS	RADIO	RESULTS	SALT	SEVEN	SKIN
POLICE	RAIN	RETIRED	SAME	SEVERAL	SKY
POLICY	RAISE	RETURN	SAMPLE	SEVERE	SLAM
POOL	RAISED	RETURNS	SAN	SEW	SLAVERY
POOR	RAISING	REV	SAND	SEX	SLAVES
POPE	RANDOM	REVENUE	SANK	SHADOW	SLEEP
PORCH	RANGE	REVIEW	SAT	SHALL	SLIGHT
PORK	RAPID	RICE	SAVE	SHAPE	SLIP
POSE	RAPIDLY	RICH	SAW	SHARE	SLOW
POST	RARE	RIDE	SAY	SHARES	SLOWLY
POUCH	RARELY	RIDING	SAYS	SHARP	SMALL
POUNDS	RATE	RIFLE	SCALE	SHARPLY	SMELL
POWER	RATES	RIGHT	SCENE	SHE	SMILE
POWERS	RATHER	RIGHTS	SCHEME	SHEAR	SMILED
PRECISE	RATIO	RING	SCHOOL	SHEET	SMILING
PREPARE	RAVE	RIPE	SCHOOLS	SHELL	SMITH
PRESENT	RAW	RISE	SCIENCE	SHELTER	SMOKE
PRESS	RAYBURN	RISING	SCORE	SHIFT	SMOOTH
PRETTY	REACH	RISK	SCREEN	SHIP	SNAKE
PREVENT	READ	RIVER	SEA	SHIPS	SNOW
PRICE	READER	ROAD	SEARCH	SHOE	SO
PRIDE	READERS	ROADS	SEASON	SHOES	SOAP
PRIMARY	READING	ROCK	SEAT	SHOOK	SOCIAL
PRIME	READY	ROD	SECOND	SHOP	SOCIETY
PRINCE	REAL	RODE	SECRET	SHORE	SOFT
PRINTED	REALITY	ROLE	SECTION	SHORT	SOIL
PRIOR	REALIZE	ROLES	SEE	SHORTLY	SOLD
PRISON	REALLY	ROLL	SEED	SHOT	SOLDIER
PRIVATE	REAR	ROMAN	SEEING	SHOULD	SOLID
PROBLEM	REASON	ROOM	SEEK	SHOW	SOME
PROCESS	REASONS	ROOMS	SEEKING	SHOWING	SOMEHOW
PRODUCE	RECALL	ROOT	SEEM	SHOWS	SOMEONE
PRODUCT	RECEIVE	ROSE	SEEMS	SHUT	SON
PROGRAM	RECENT	ROUGH	SEEN	SIDE	SONG
PROJECT	RECORD	ROUND	SEES	SIDES	SONGS

SOON
SOOT
SORRY
SORT
SOUGHT
SOUL
SOUNDS
SOURCE
SOUTH
SOVIET
SPA
SPACE
SPANISH
SPEAK
SPEAKER
SPECIAL
SPECIES
SPEECH
SPEED
SPEND
SPENT
SPHINX
SPIRIT
SPIRITS
SPITE
SPOKE
SPOKEN
SPOOK
SPORTS
SPOT
SPREAD
SQUARE
STAFF
STAGE
STAIRS
STAND
STANDS
START
STARTED
STATE
STATED
STATES
STATION
STATUS
STAY
STEADY
STEAK
STEMS
STEP
STEPS
STICK
STILL
STOCK
STOMACH
STONE
STOOD
STOP
STORAGE
STORE
STORES
STORIES
STORY
STRANGE
STREAM
STREET
STREETS
STRESS
STRIKE
STRONG
STRUCK
STUDENT
STUDIES
STUDY
STUNT
STYLE
SUBJECT
SUCESS
SUCH
SUDDEN
SUFFER
SUGAR
SUGGEST
SUIT
SUM
SUMMER
SUN
SUNDAY
SUPPER
SUPPLY
SUPPORT
SUPPOSE
SUPREME
SURE
SURELY
SURFACE
SURVEY
SURVIVE
SWAMP
SWARM
SWEET
SWEPT
SWITCH
SWORD
SWORE
SWUNG
SYMBOL
SYMBOLS
SYSTEM
SYSTEMS
TABLE
TABLES
TAKE
TAKEN
TAKES
TAKING
TALENT
TALK
TALKING
TALL
TAPE
TAPS
TARGET
TASK
TASTE
TAUGHT
TAX
TAXES
TEACH
TEACHER
TEAM
TEARS
TEETH
TELL
TELLING
TELLS
TEMPLE
TEN
TEND
TENDS
TENSION
TERM
TERMS
TEST
TESTED
TESTS
TEXT
THAN
THANK
THAT
THE
THEATER
THEIR
THEM
THEME
THEN
THEORY
THERE
THEREBY
THERMAL
THESE
THEY
THICK
THIN
THING
THINGS
THINK
THIRD
THIRTY
THIS
THOSE
THOUGH
THOUGHT
THREAT
THREE
THREW
THROAT
THROUGH
THROW
THROWN
THUS
THYROID
TIED
TILE
TILL
TIME
TIMES
TINT
TINY
TIRED
TISSUE
TITLE
TO
TOAD
TODAY
TOLD
TOM
TOMB
TONE
TONGUE
TOO
TOOK
TOOL
TOOLS
TOP
TORQUE
TOTAL
TOUCH
TOUGH
TOUR
TOWARD
TOWARDS
TOWN
TOWNS
TRACK
TRADE
TRAFFIC
TRAGEDY
TRAGIC
TRAIN
TRAVEL
TREE
TREES
TREND
TRIAL
TRIALS
TRIED
TRIP
TROOPS
TROUBLE
TRUCK
TRUE
TRULY
TRUNK
TRUST
TRUTH
TRY
TRYING
TRYST
TSAR
TUESDAY
TURN
TURNING
TURNS
TWELVE
TWENTY
TWICE
TWO
TYPE
TYPES
TYPICAL
UNABLE
UNCLE
UNDER
UNION
UNIQUE
UNIT
UNITS
UNITY
UNKNOWN
UNLESS
UNLIKE
UNTIL
UP
UPON
UPPER
URBAN
US
USE
USEFUL
USING
USUAL
VALLEY
VALUE
VARIETY
VARIOUS
VARY
VARYING
VAST
VERSION
VERY
VIA
VICE
VICTORY
VIEW
VIEWS
VILLAGE
VIOLENT
VIRGIN
VISIBLE
VISION
VISIT
VISUAL
VITAL
VOICE
VOLUME
VOLUMES
VOTE
WAD
WAGE
WAGON
WAIT
WAITING
WAKE
WALK
WALKING
WALL
WALLS
WAN
WAND
WANT
WANTS
WAR
WARFARE
WARM
WARNING
WARREN
WASH
WASTE
WATCH
WATER
WATERS
WAVE
WAVES
WAX
WAY
WAYS
WE
WEAPON
WEAPONS
WEAR
WEARING
WEATHER
WEEK
WEEKS
WEIGHT
WEIRD
WELCOME
WELD
WELFARE
WELL
WENT
WERE
WEST
WESTERN
WET
WHAT
WHEEL
WHEN

WHERE
WHEREAS
WHETHER
WHICH
WHILE
WHITE
WHO
WHOLE
WHOM
WHOSE
WHY
WIDE
WIDELY
WIFE
WILD
WILL
WILLING
WIN
WIND
WINDOW
WINDOWS
WINE
WING
WINTER
WIRE
WISDOM
WISE
WISH
WIT
WITH
WITHIN
WITHOUT
WOMAN
WOMEM
WON
WONDER
WOOD
WOODEN
WOOL
WORD
WORDS
WORE
WORK
WORKERS
WORKING
WORKS
WORLD
WORM
WORRY
WORSE
WORSHIP
WORST
WORTH
WOULD
WRITE
WRITER
WRITERS
WRITES
WRITTEN
WRONG
WROTE
YARD
YARDS
YEAR
YEARS
YELLOW
YELP
YES
YET
YIELD
YOU
YOUNG
YOUNGER
YOUR

LIST OF FIGURES

LIST OF TABLES

A Schematic View of a Feed-Forward Network

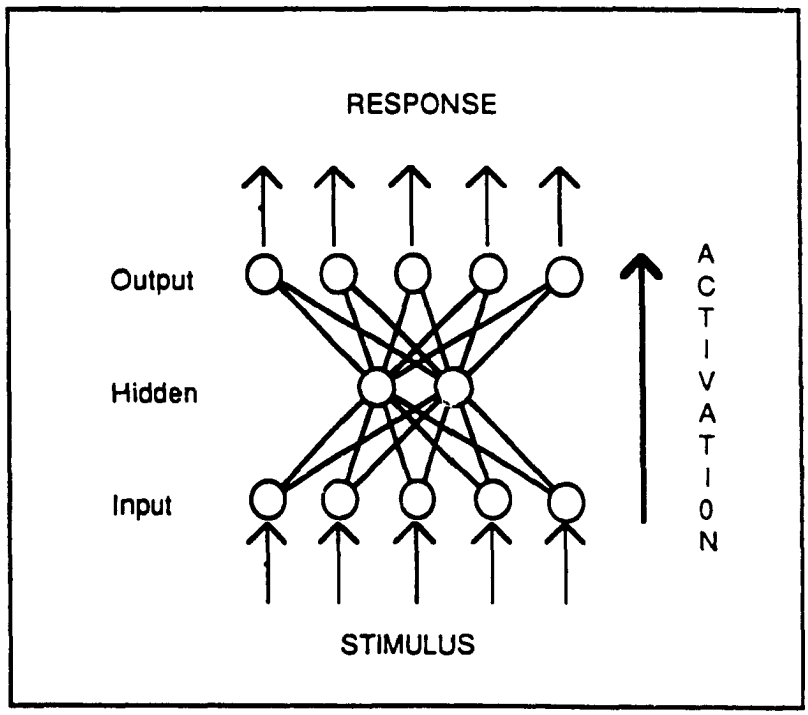

Figure 2.1

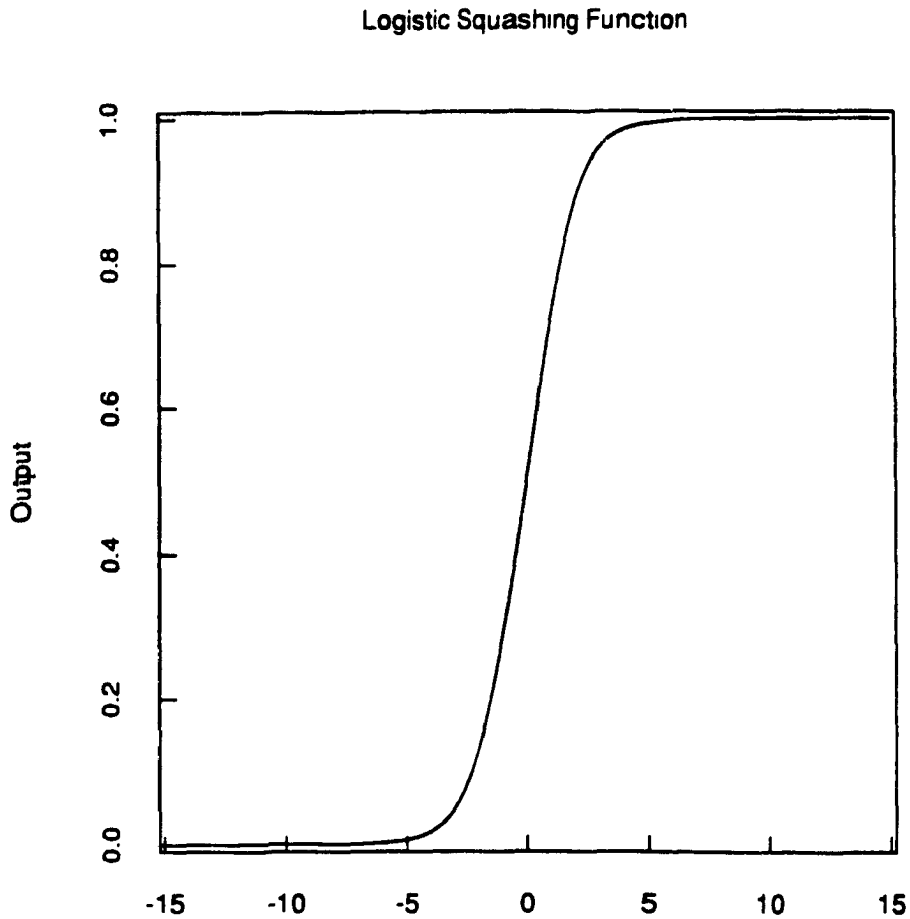

Figure 2.2

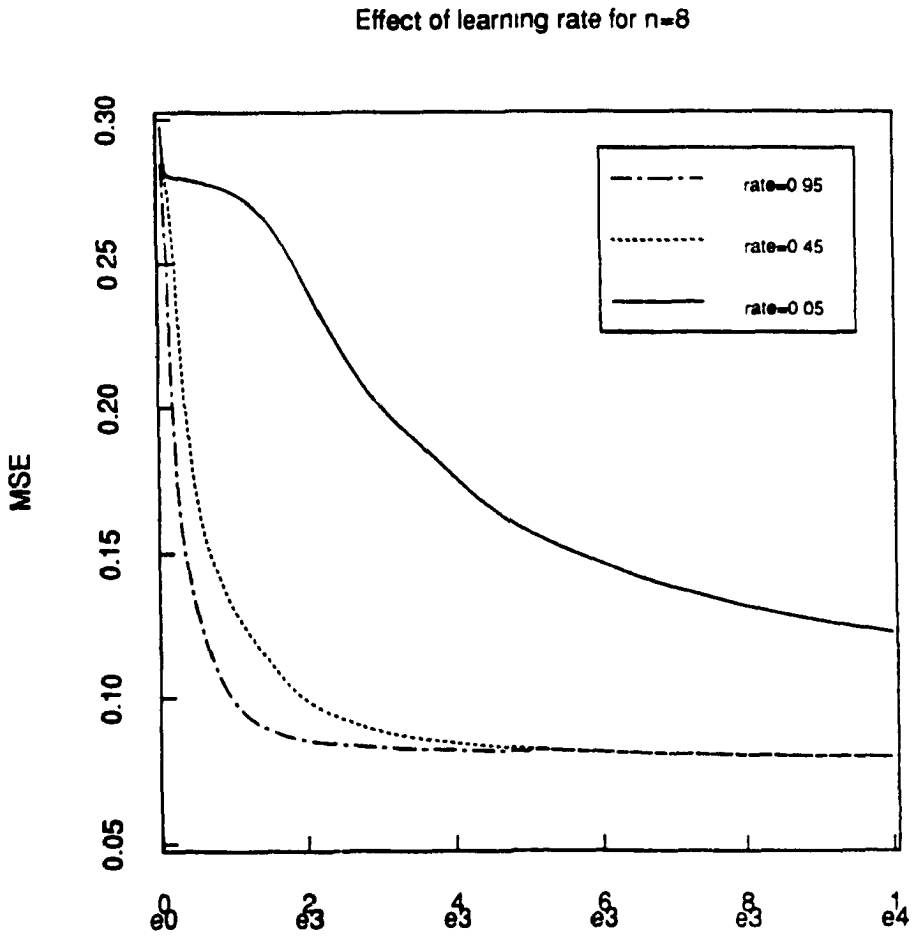

Figure 2.3

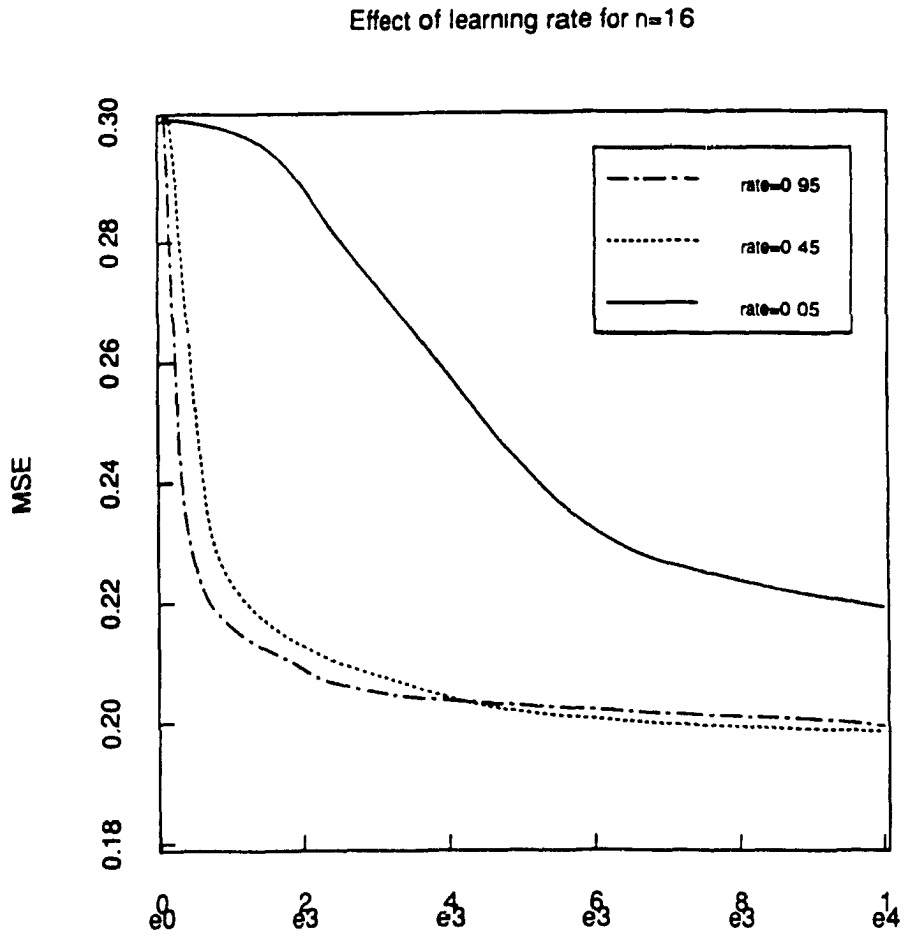

Figure 2.4

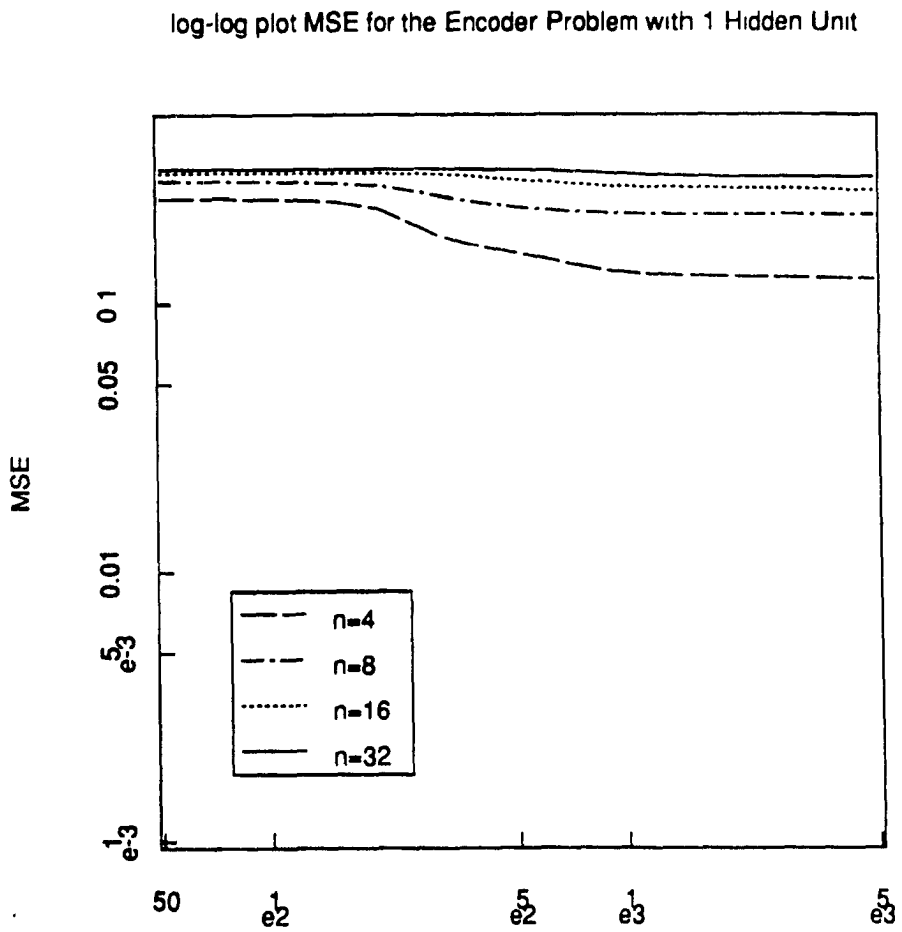

Figure 2.5

log-log plot MSE for the Encoder Problem with 2 Hidden Units

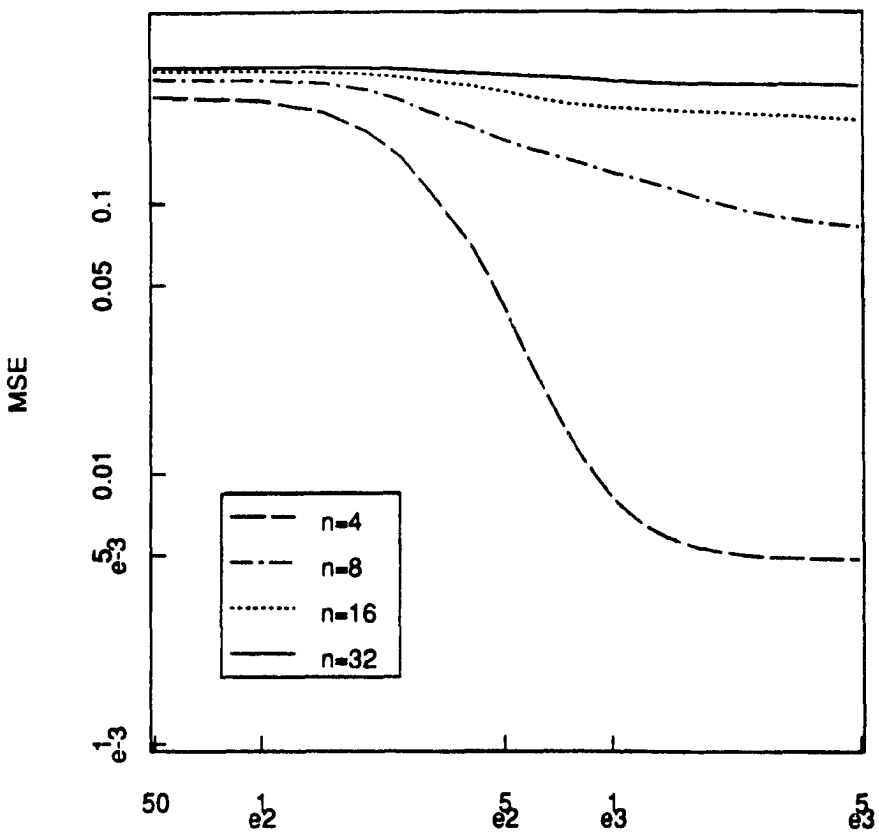

Figure 2.6

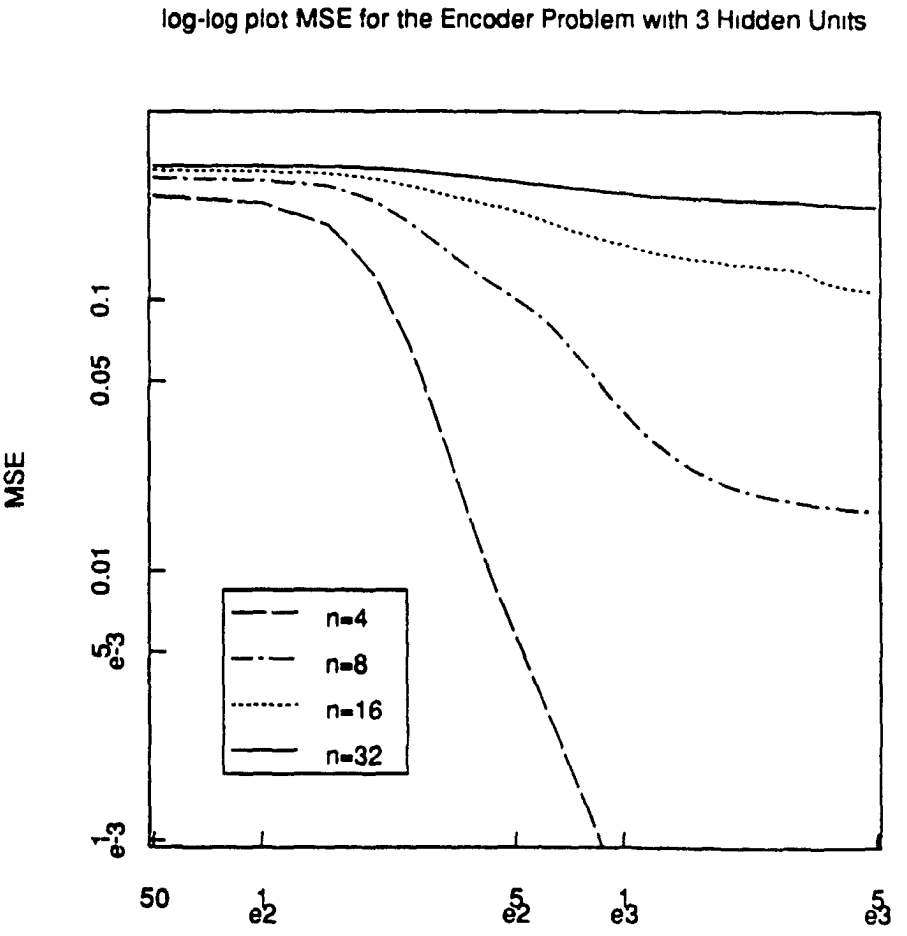

Figure 2.7

log-log plot MSE for the Encoder Problem with 4 Hidden Units

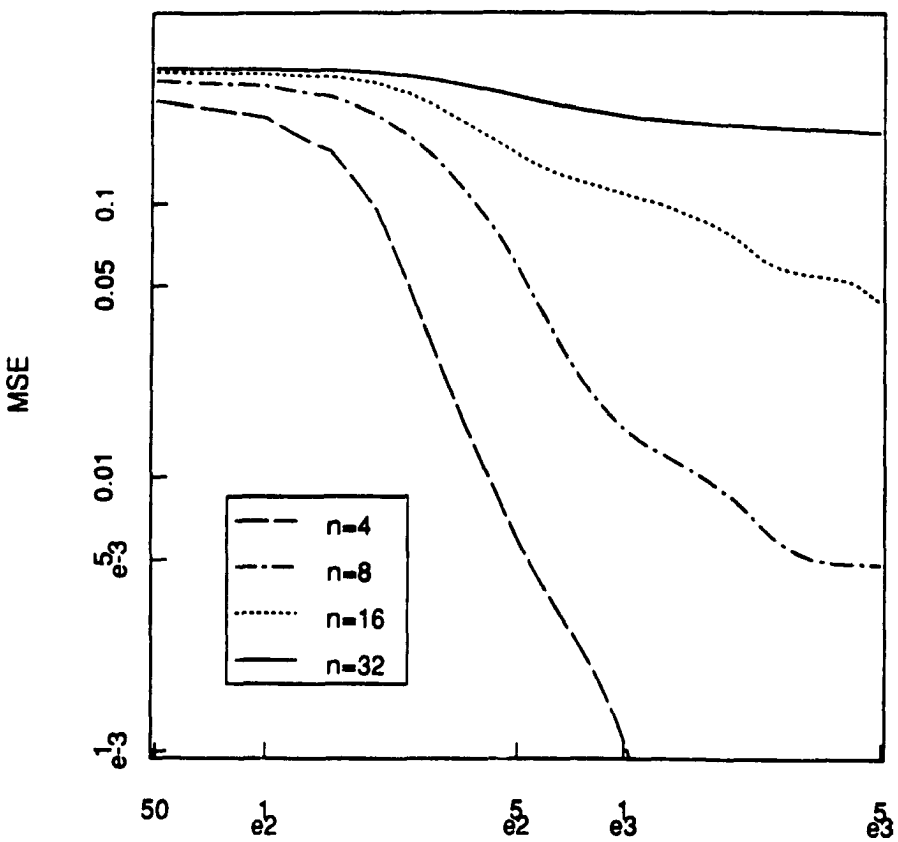

Figure 2.8

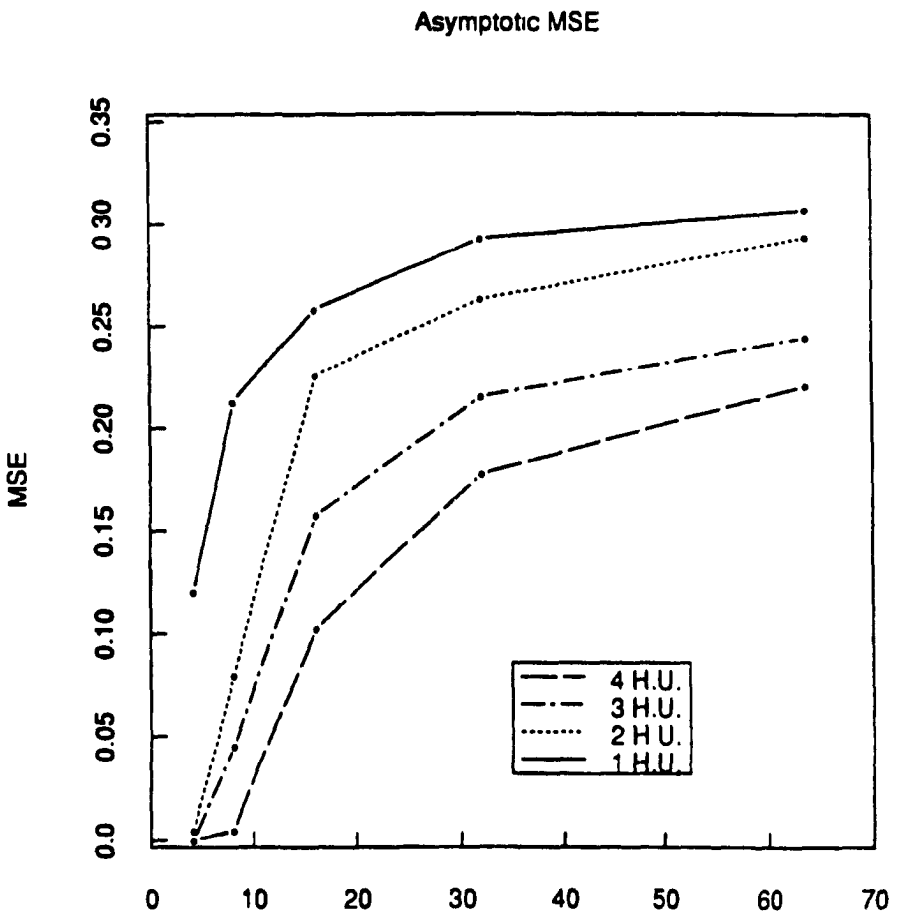

Figure 2.9

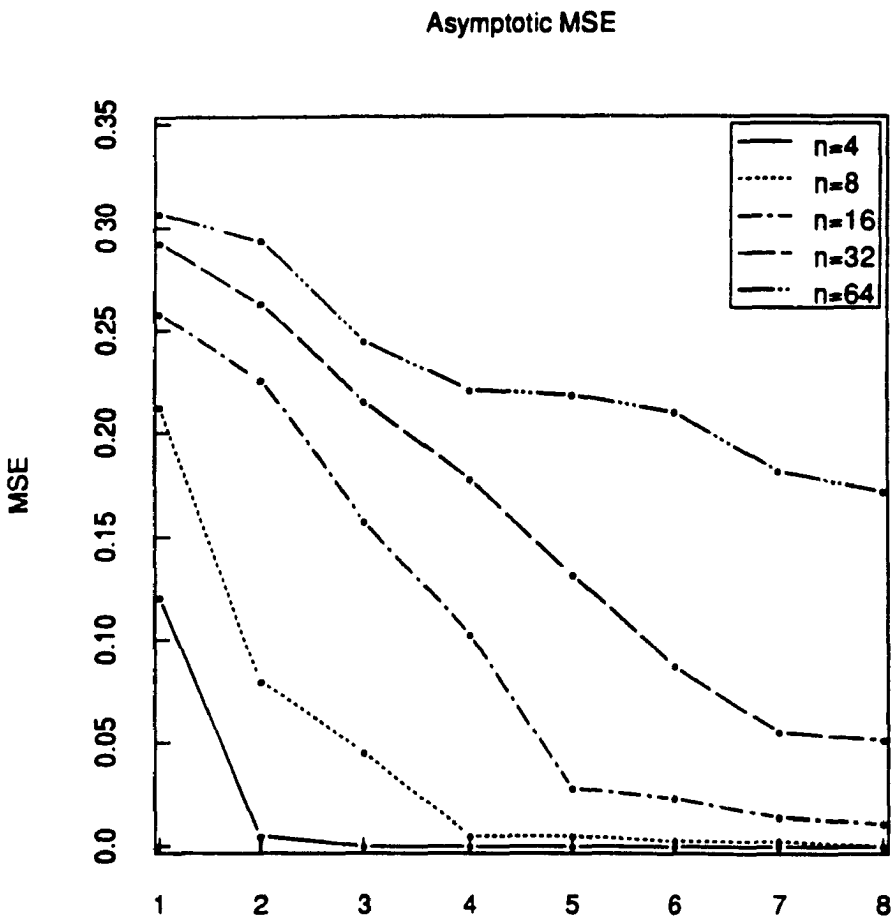

Figure 2.10

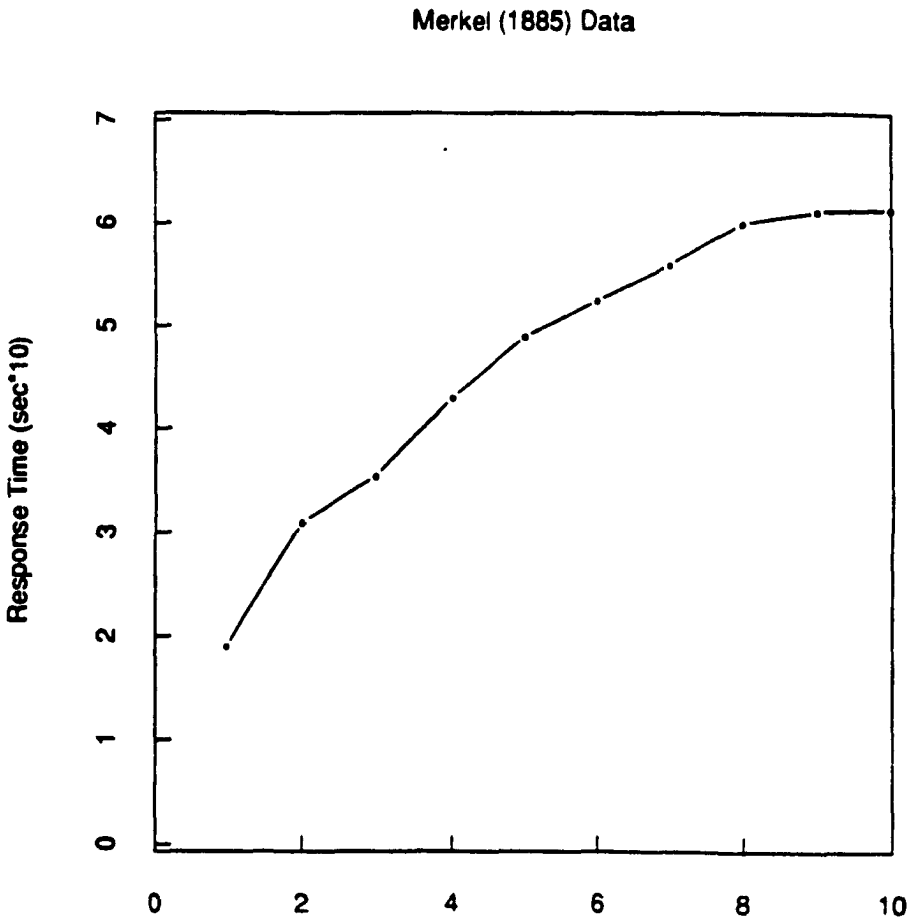

Figure 3.1

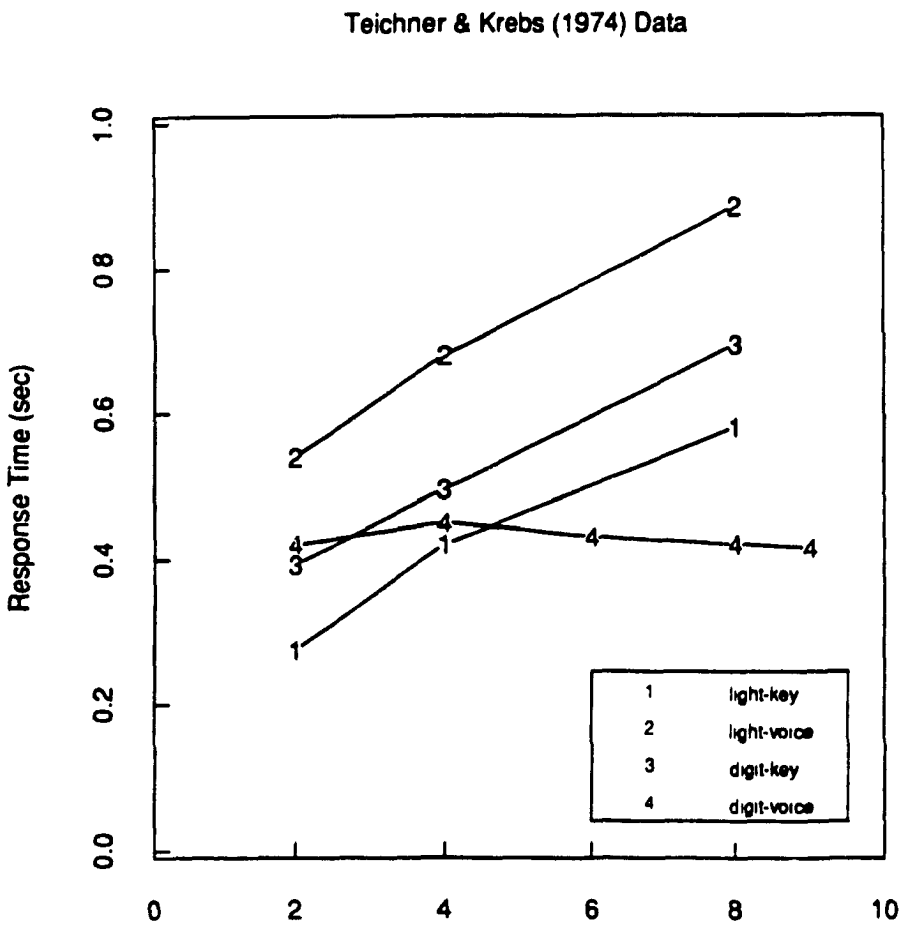

Figure 3.2

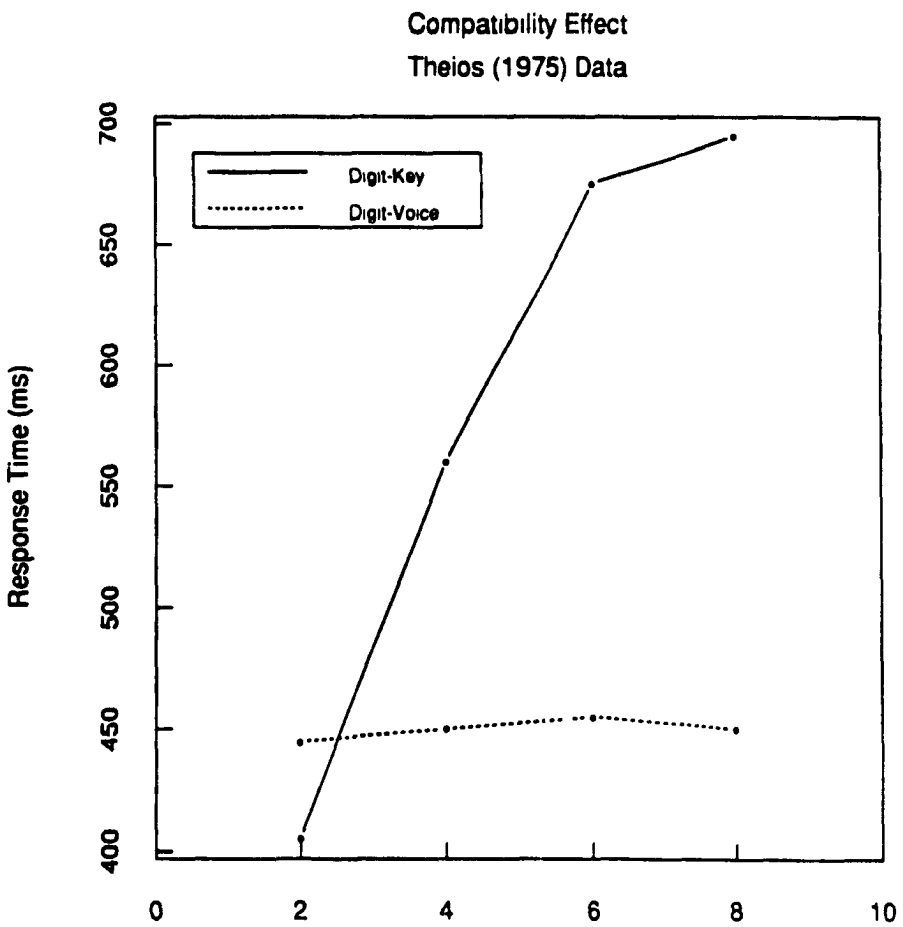

Figure 3.3

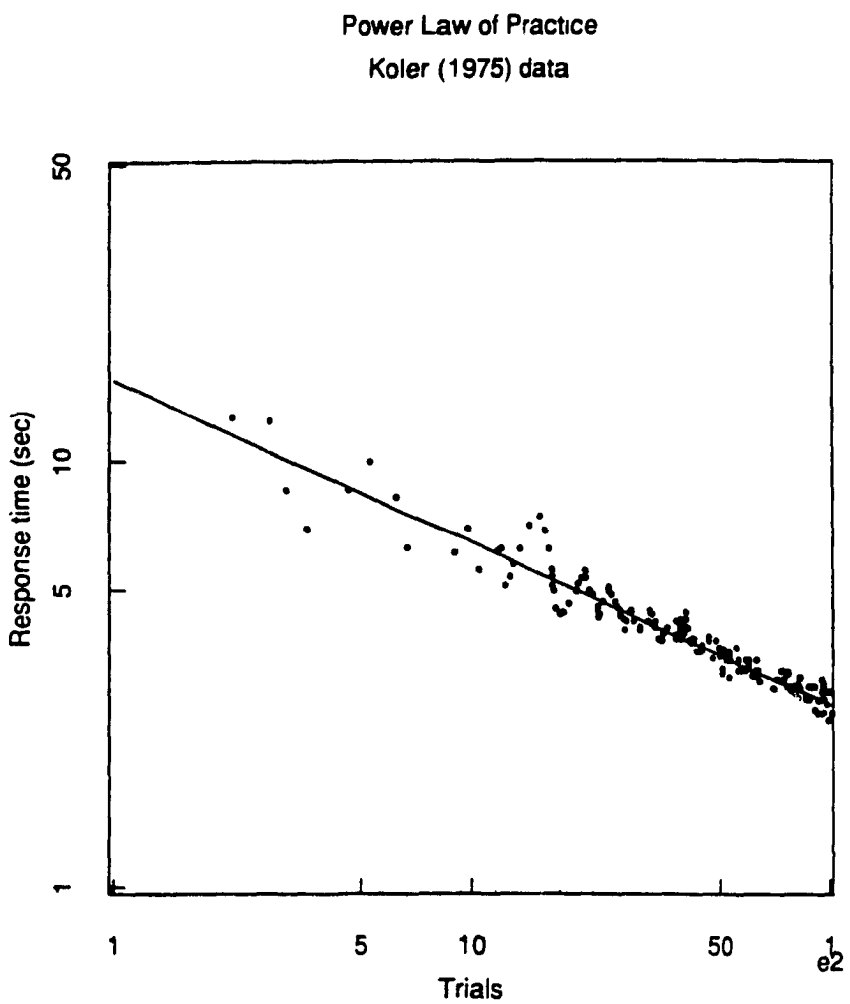

Figure 3.4

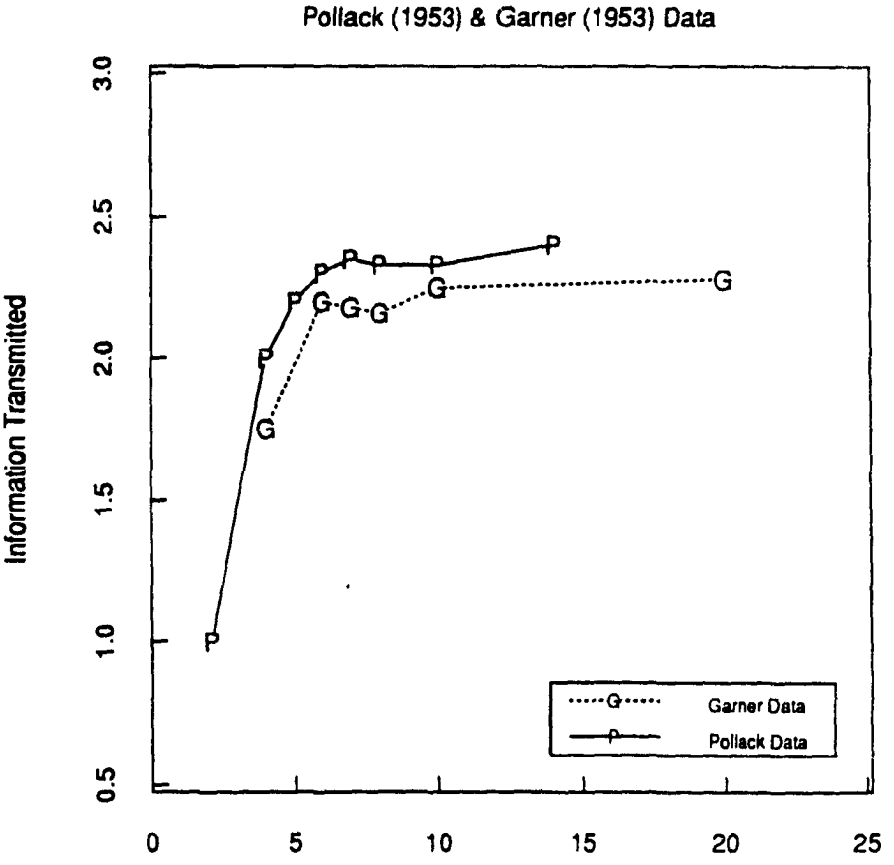

Figure 3.5

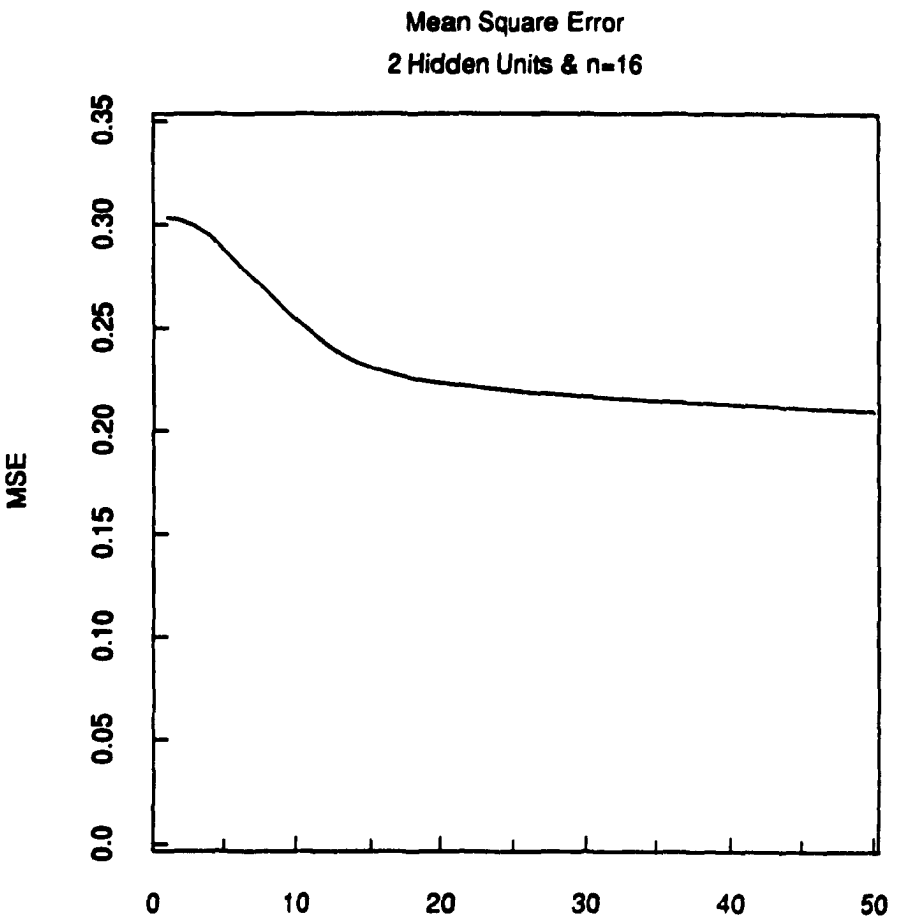

Figure 4.1

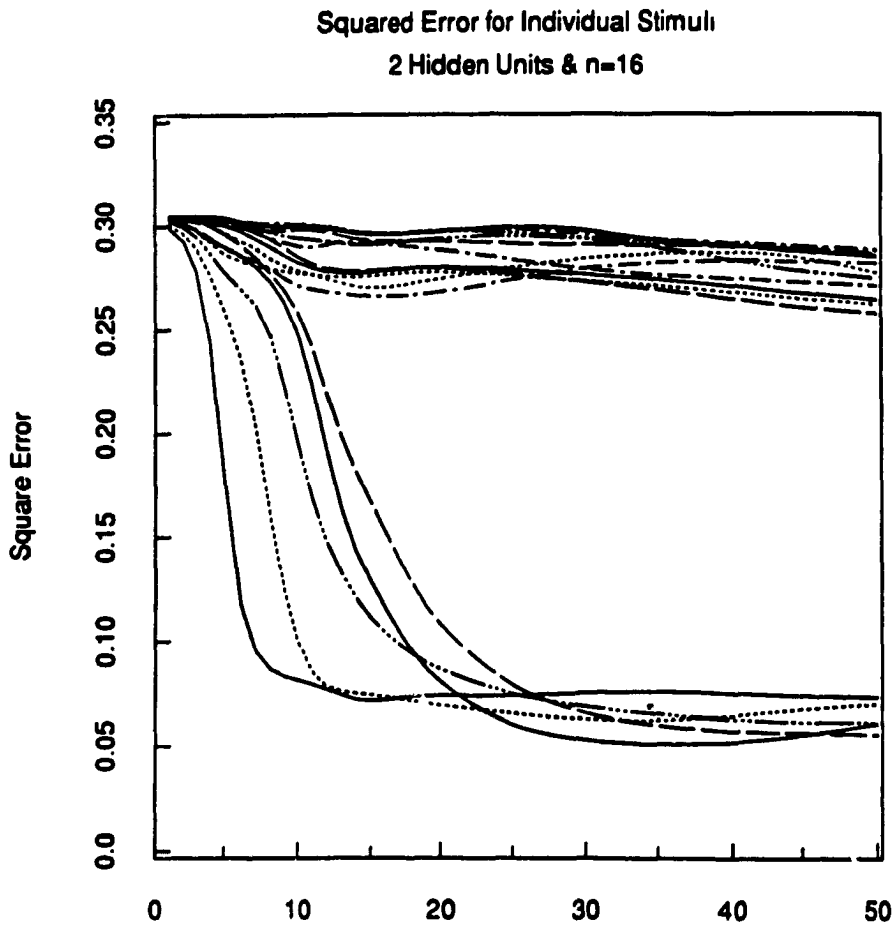

Figure 4.2

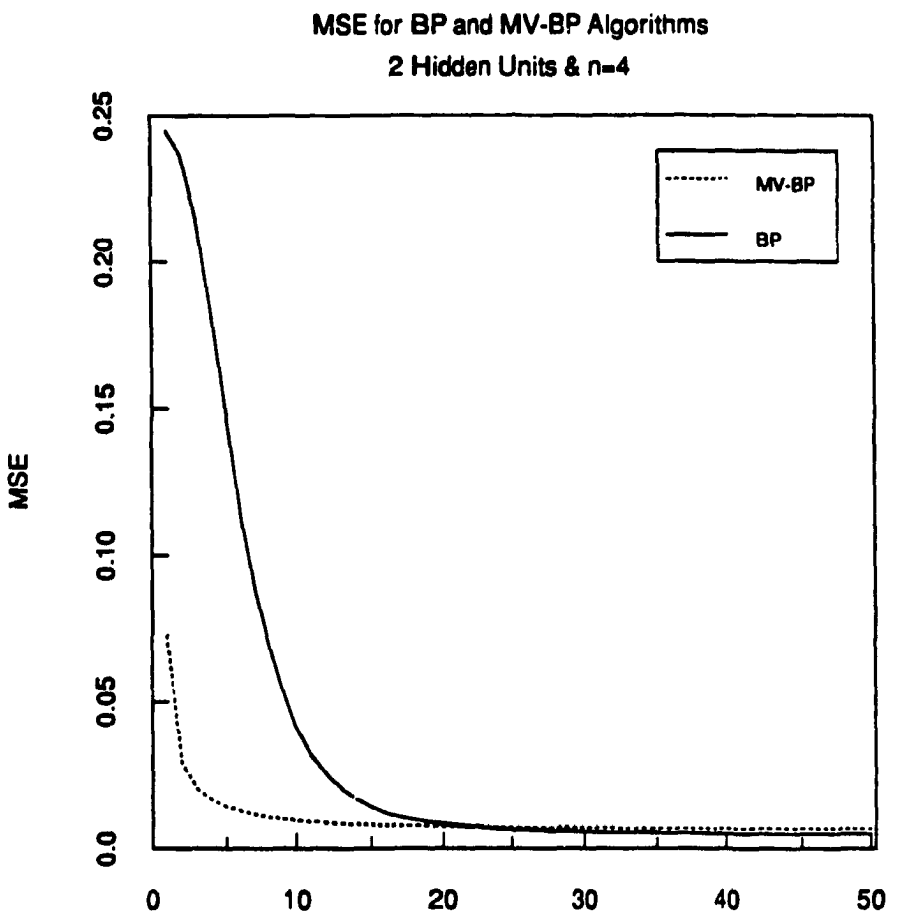

Figure 4.3

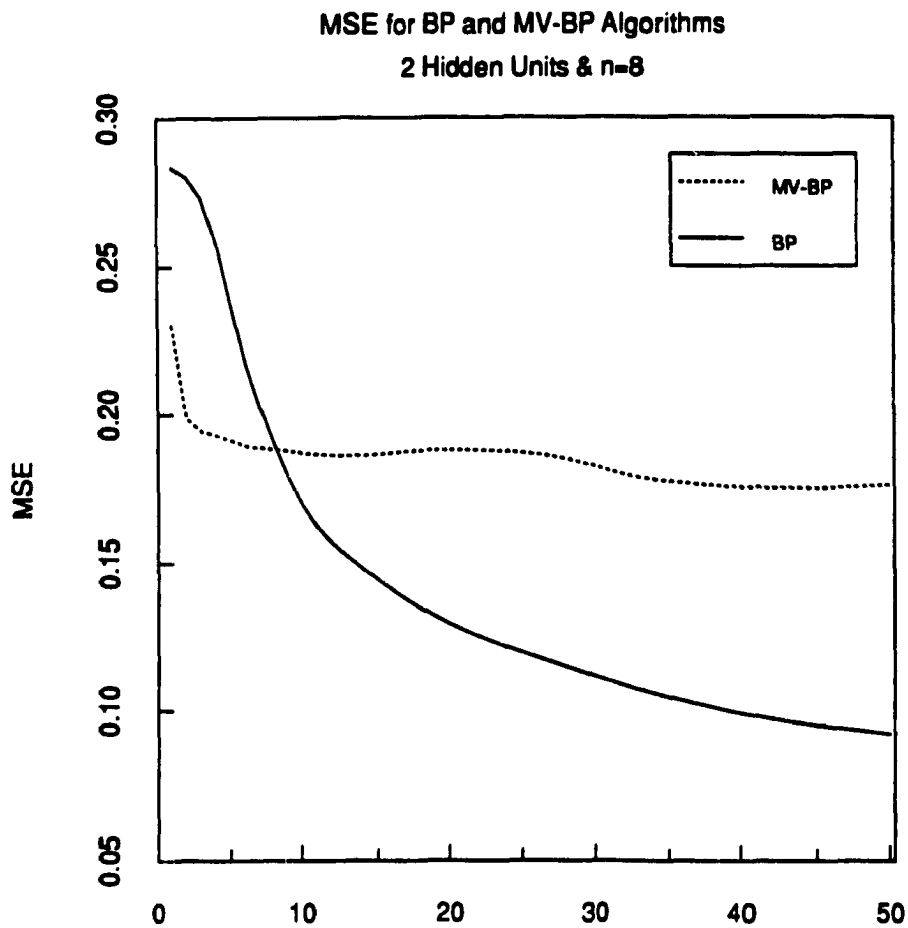

Figure 4.4

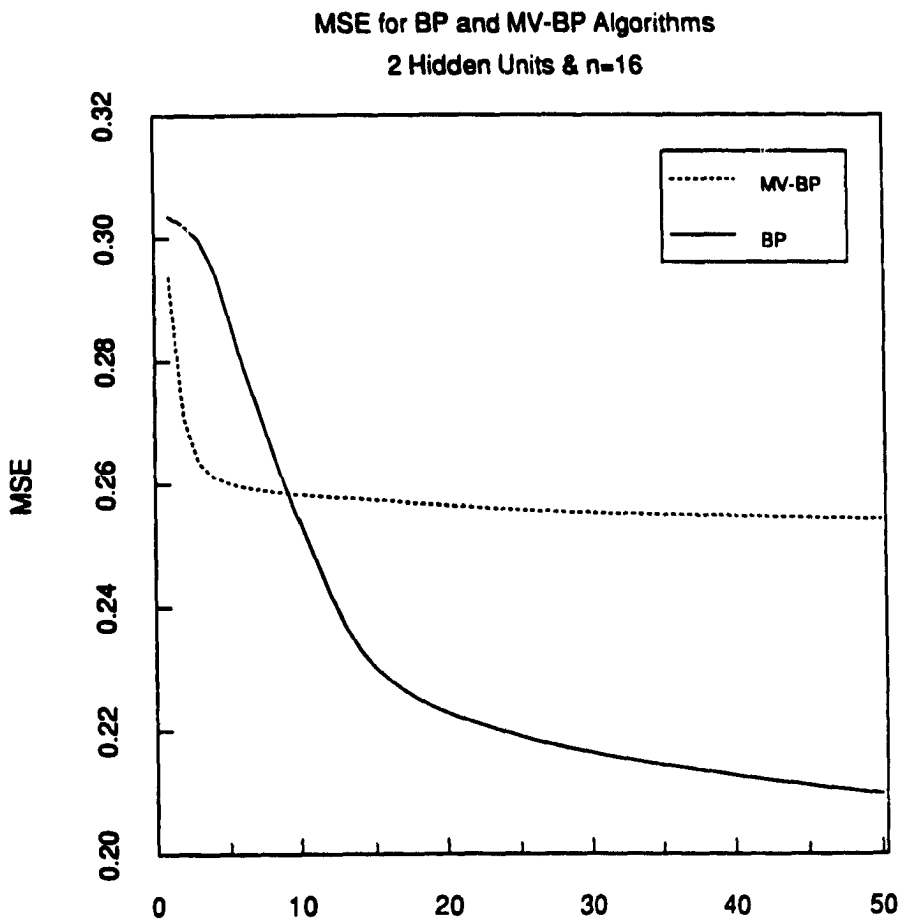

Figure 4.5

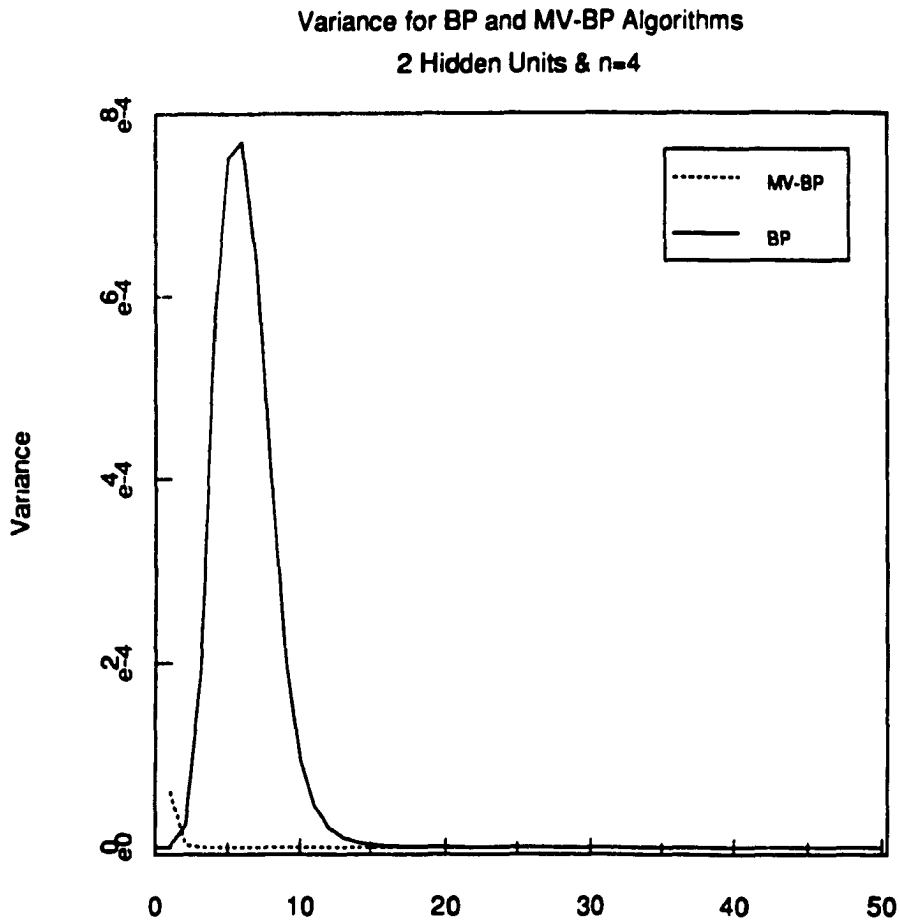

Figure 4.6

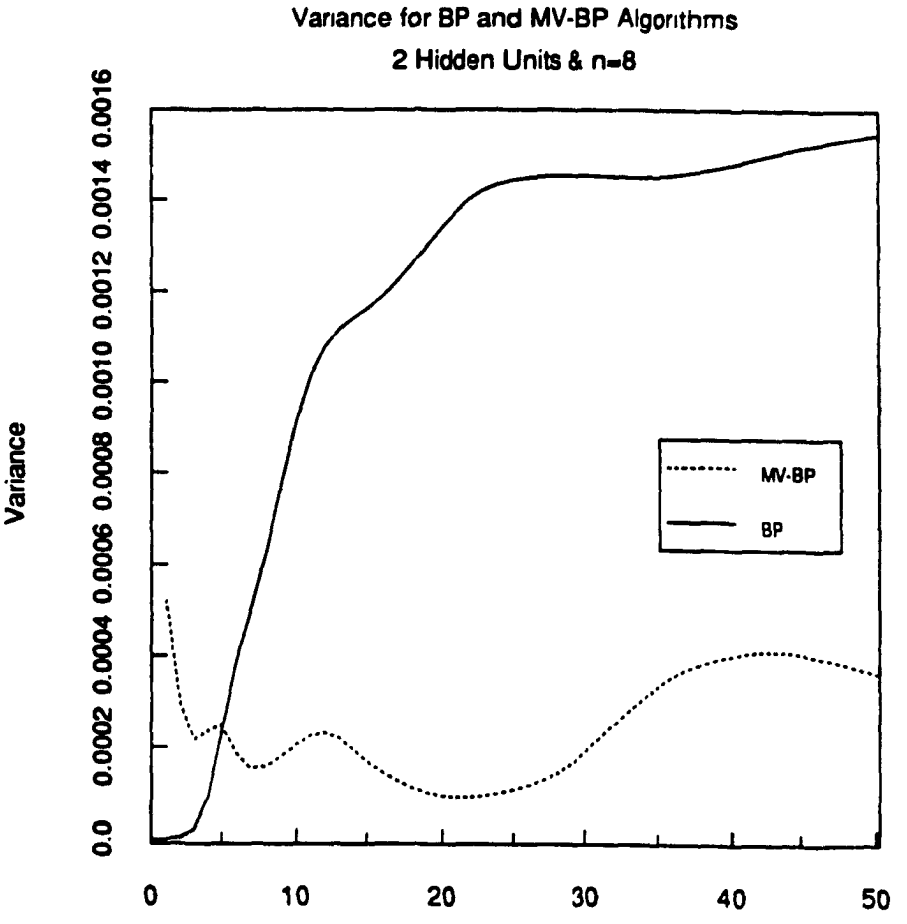

Figure 4.7

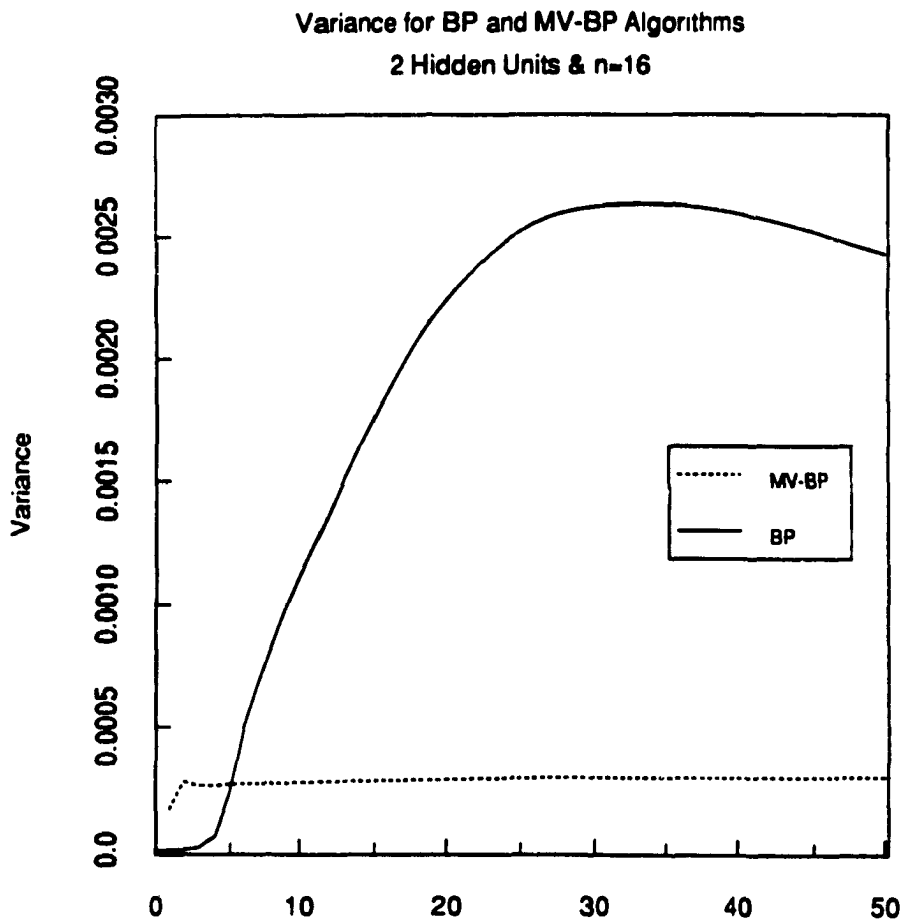

Figure 4.8

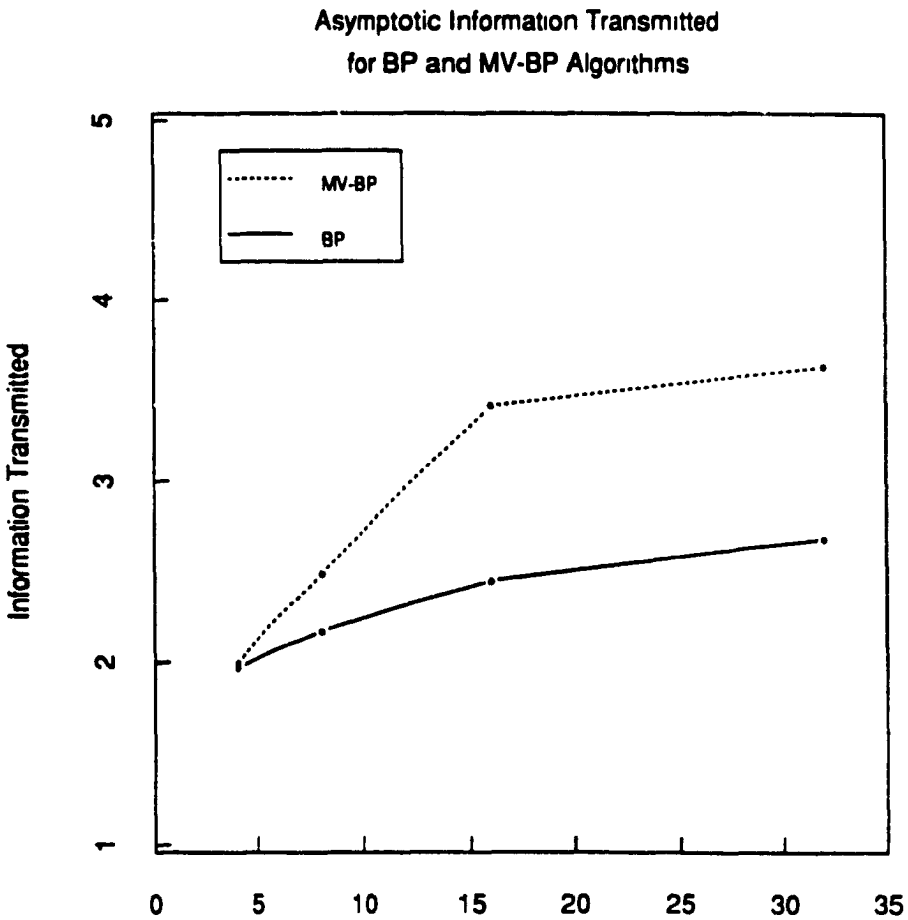

Figure 4.9

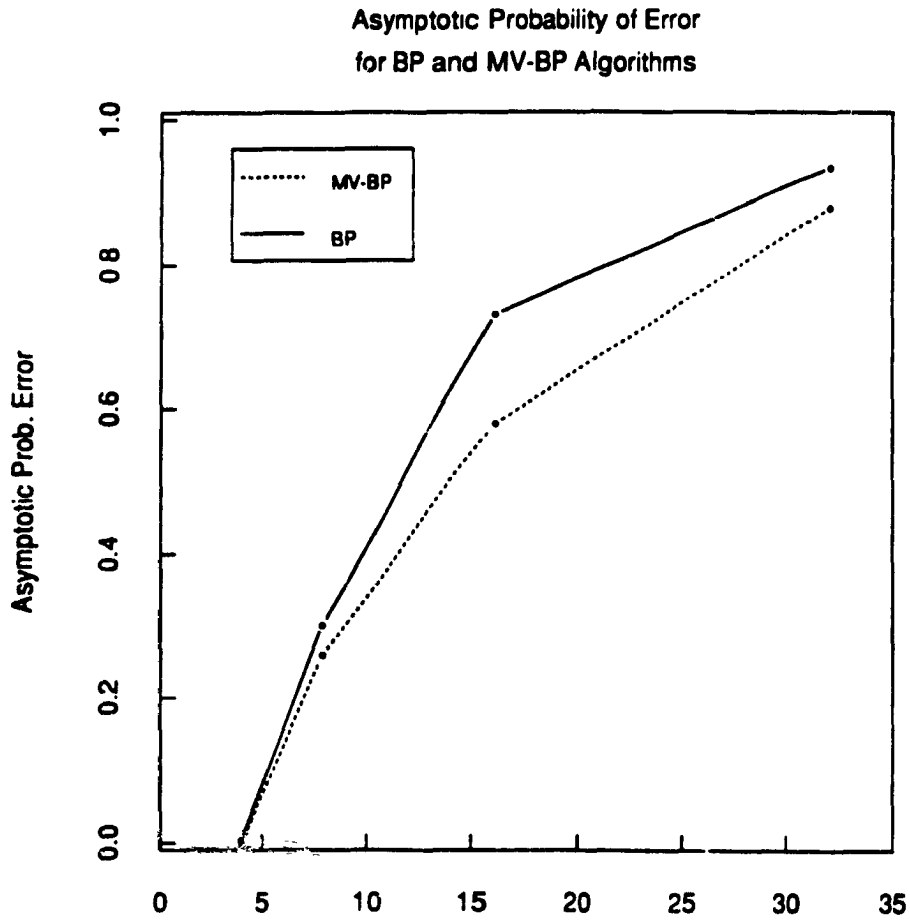

Figure 4.10

A Hybrid Architecture to Model Latencies

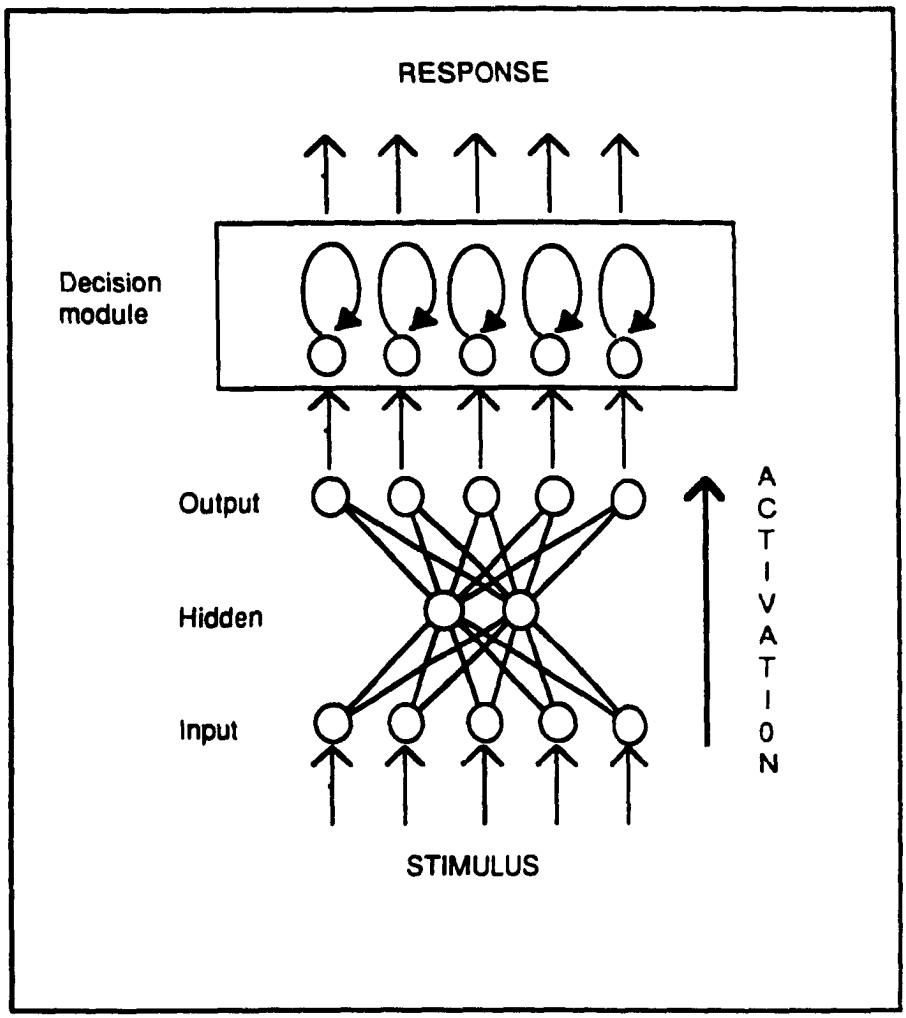

Figure 4 11

Binary and Gaussian Stimuli

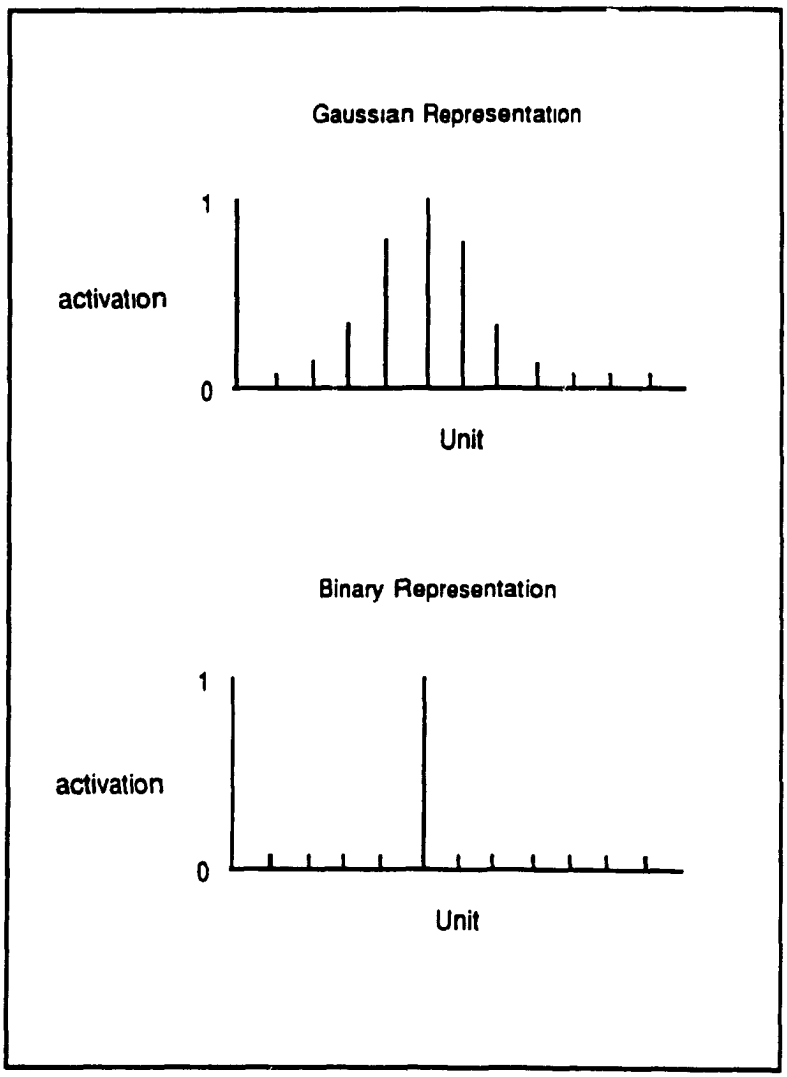

Figure 4.12

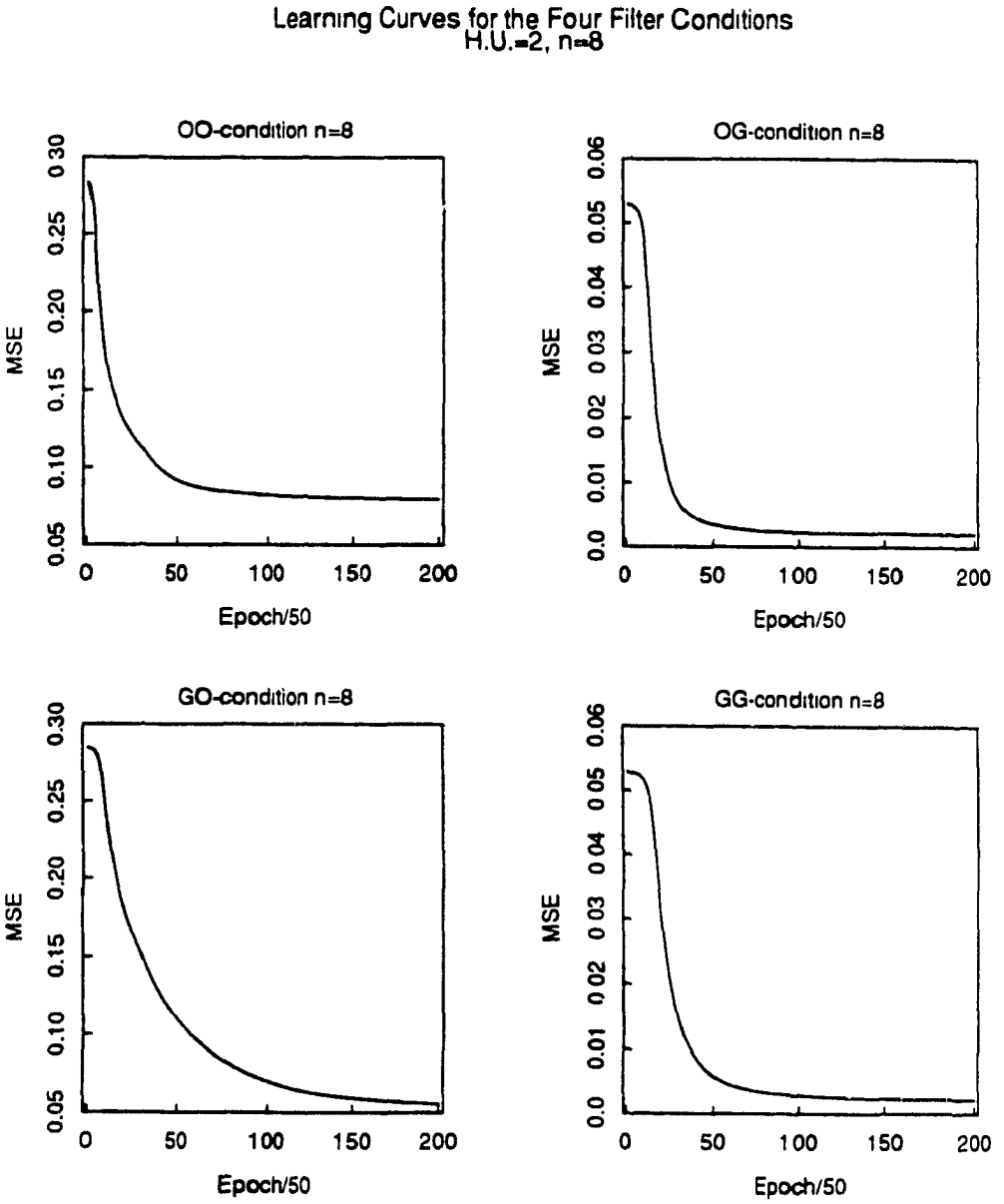

Figure 4.13

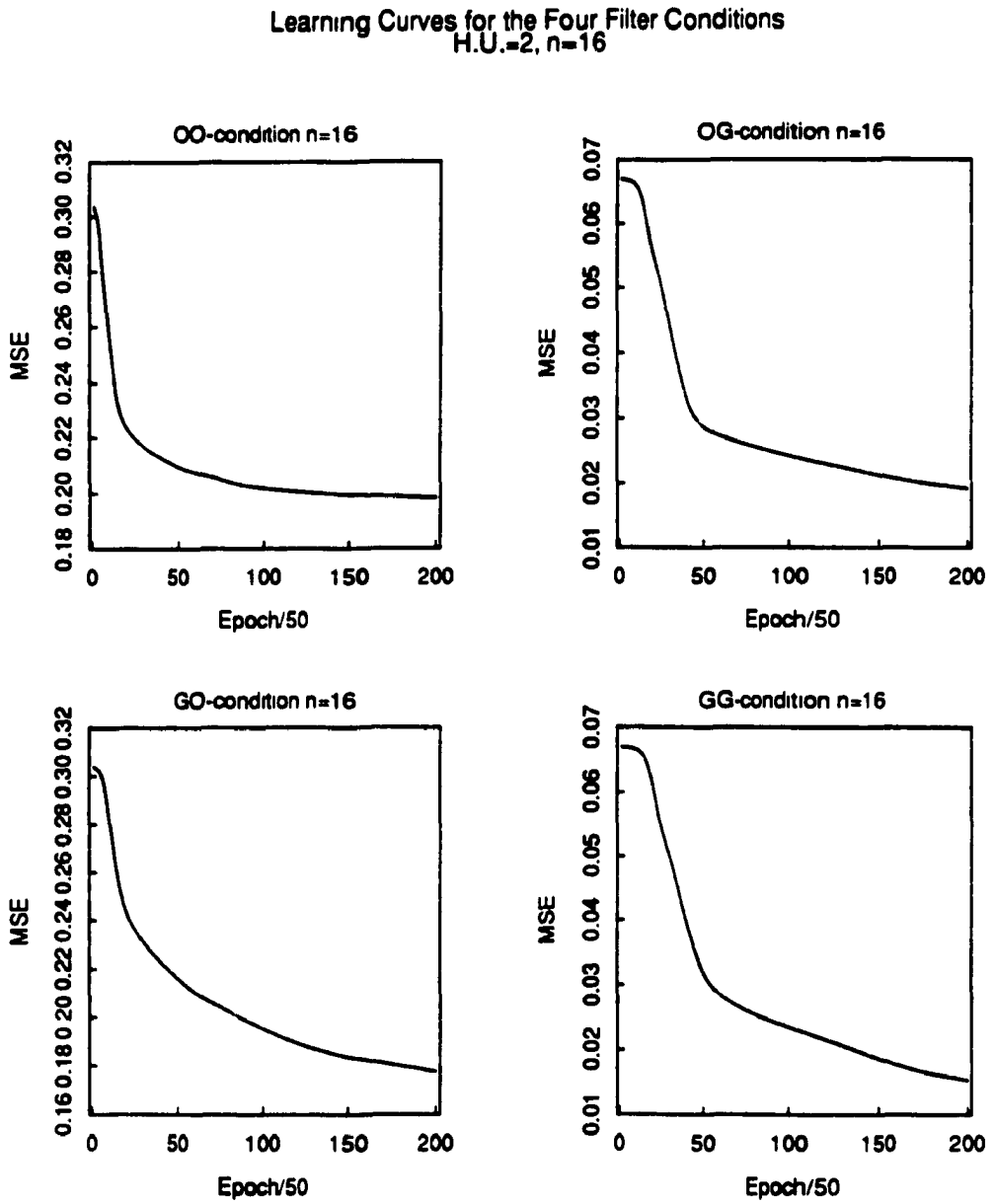

Figure 4.14

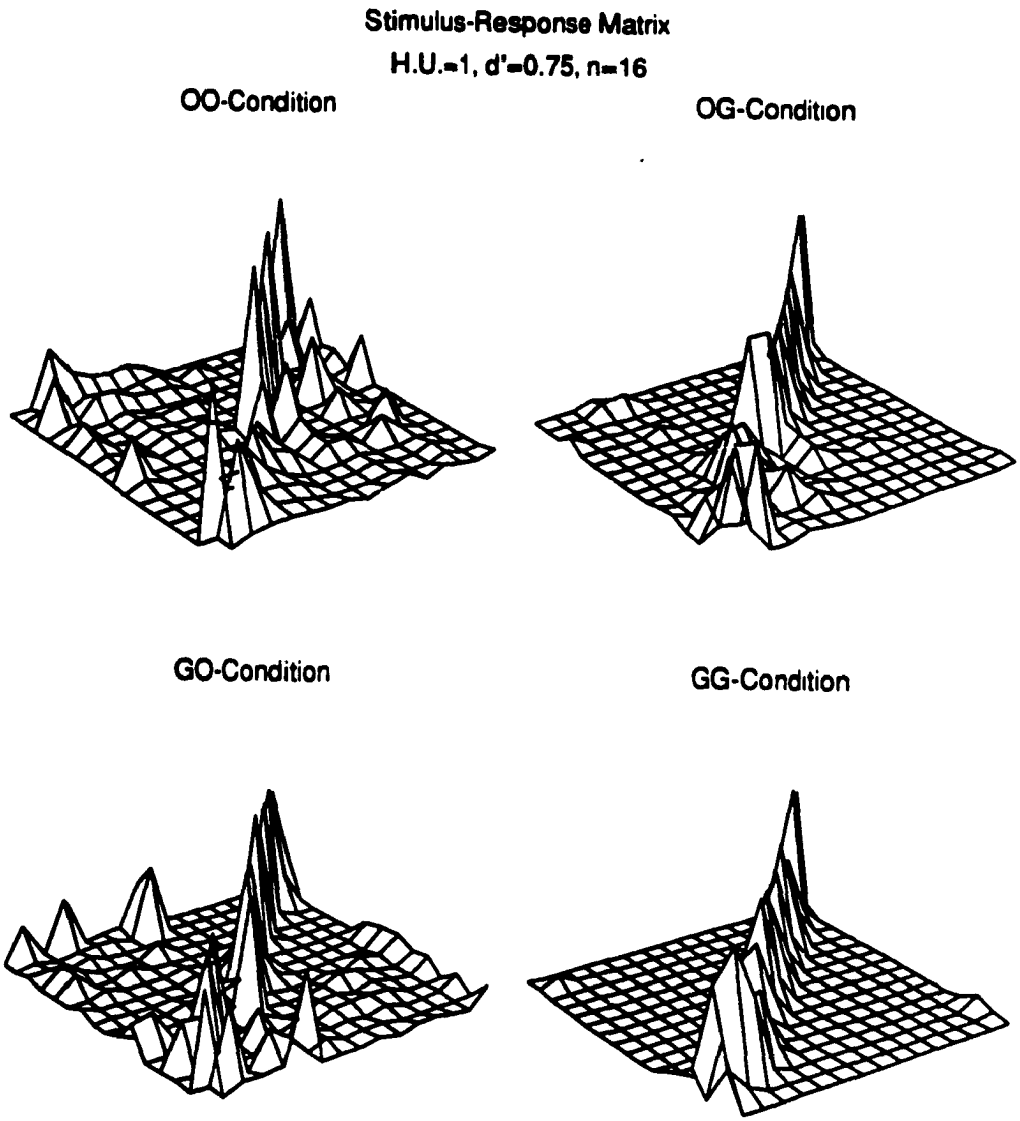

Figure 4.15

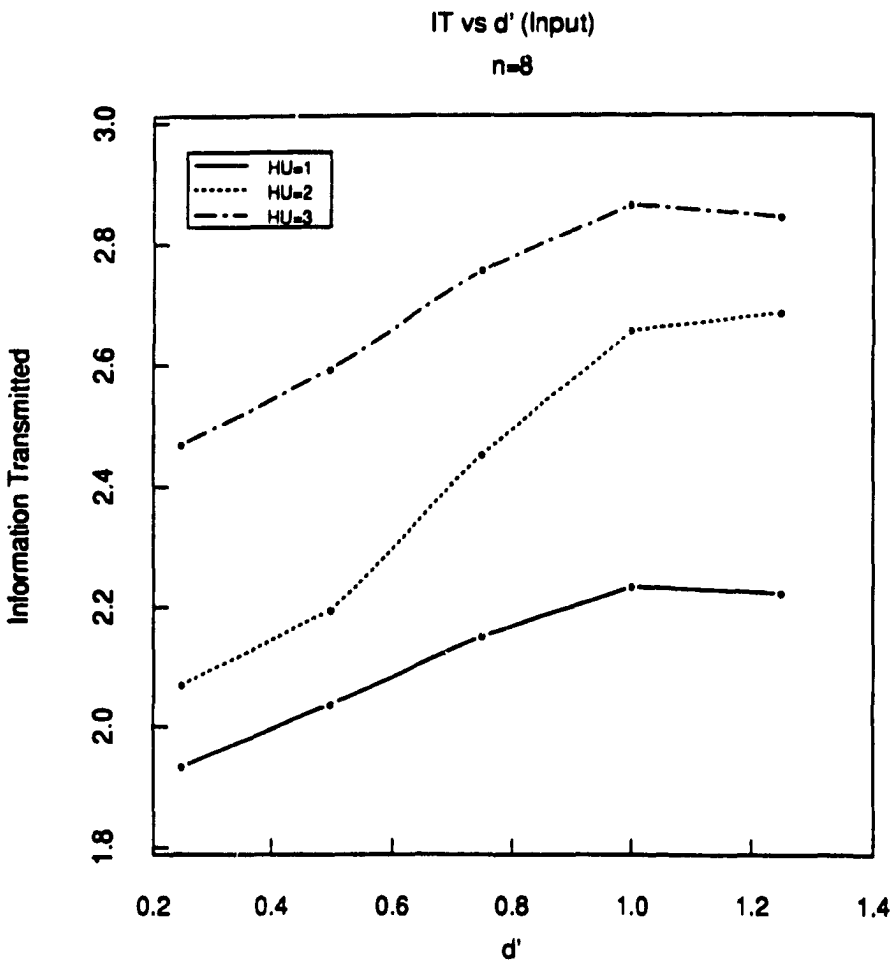

Figure 4.16a

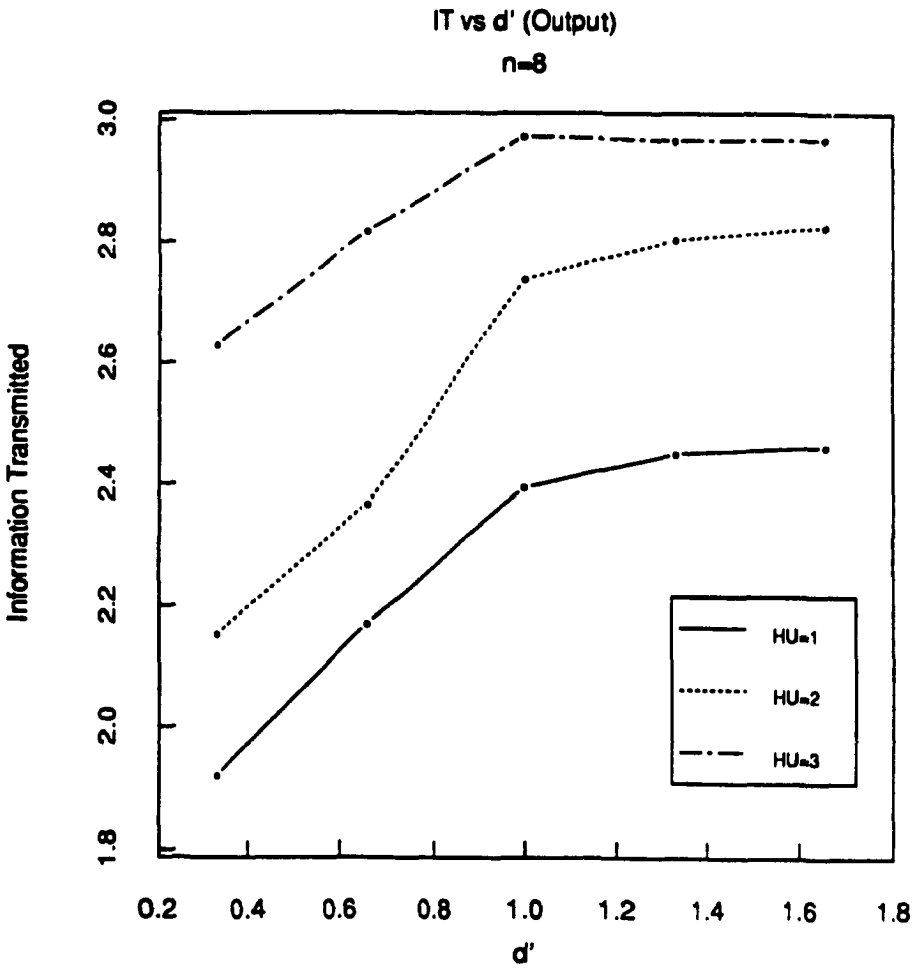

Figure 4.16b

Log-log plots for MSE

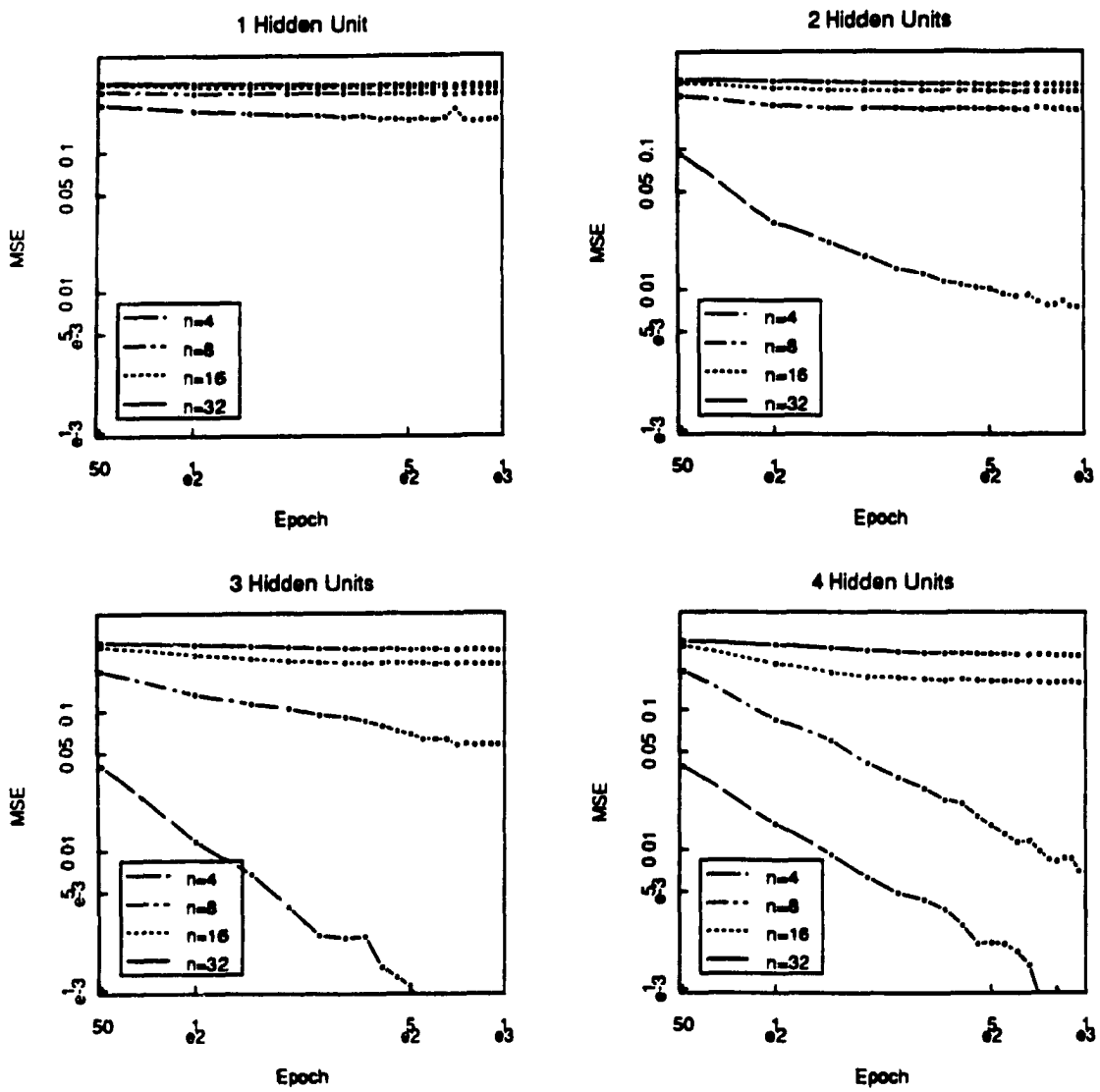

Figure 4.17

Log-log plots for IWT

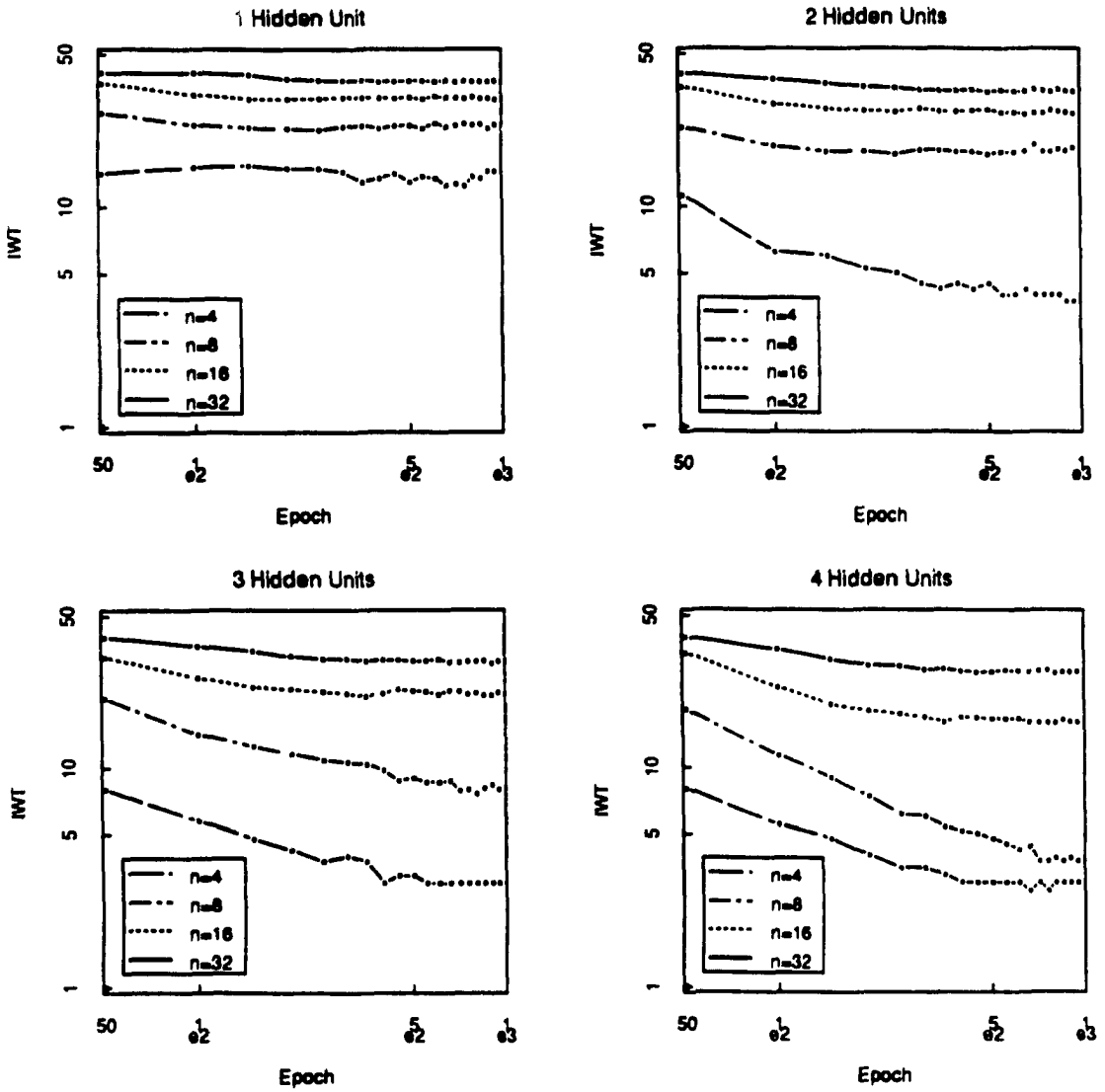

Figure 4.18

Log-log plots for WTA

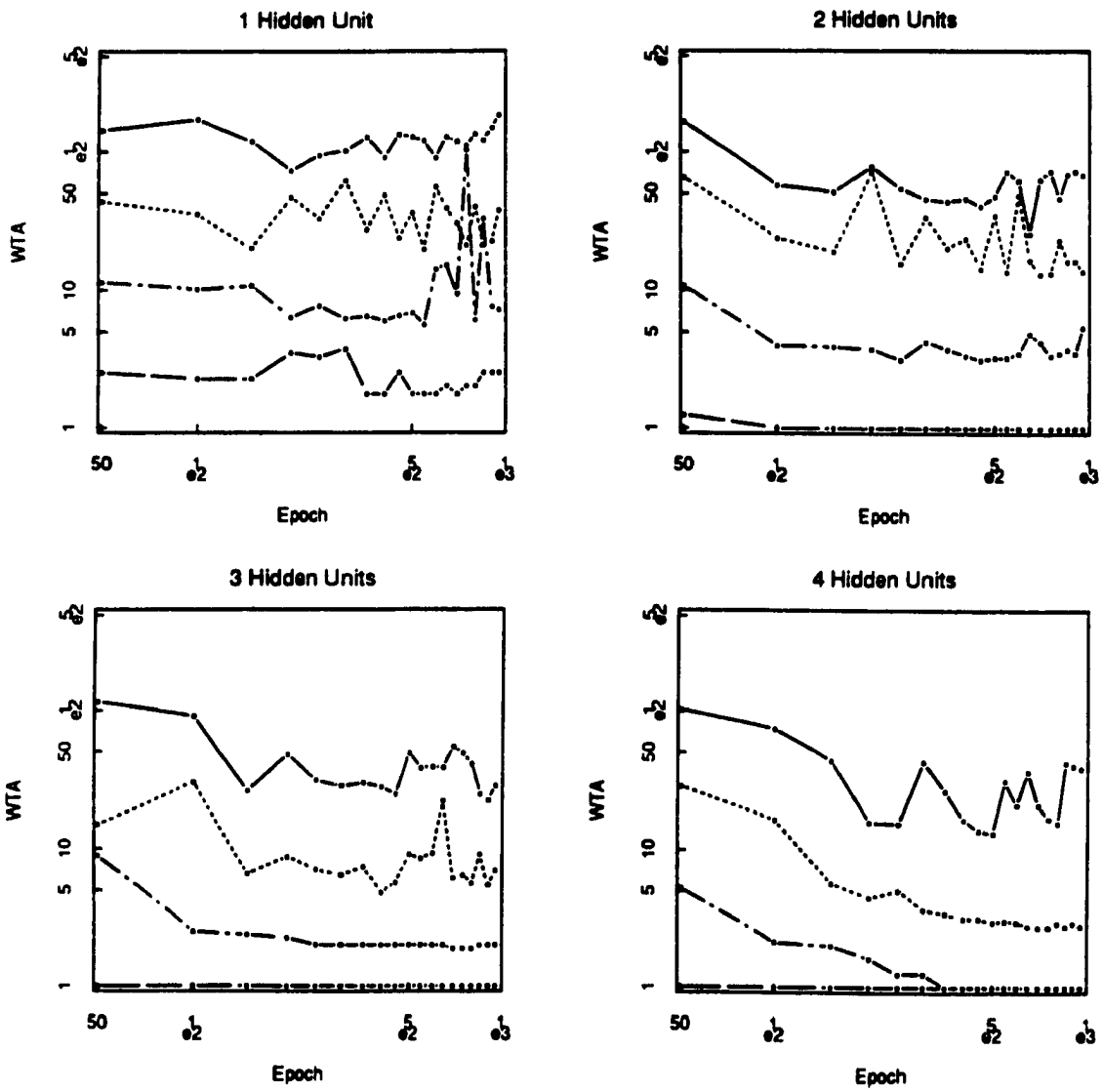

Figure 4.19

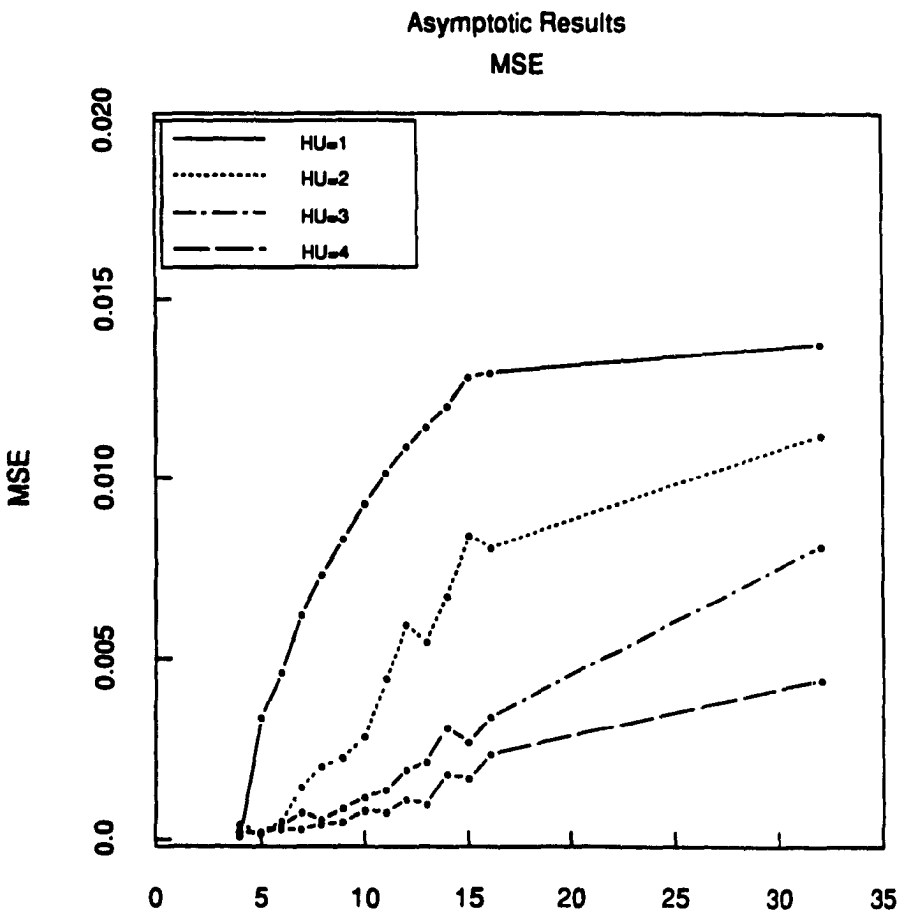

Figure 4.20

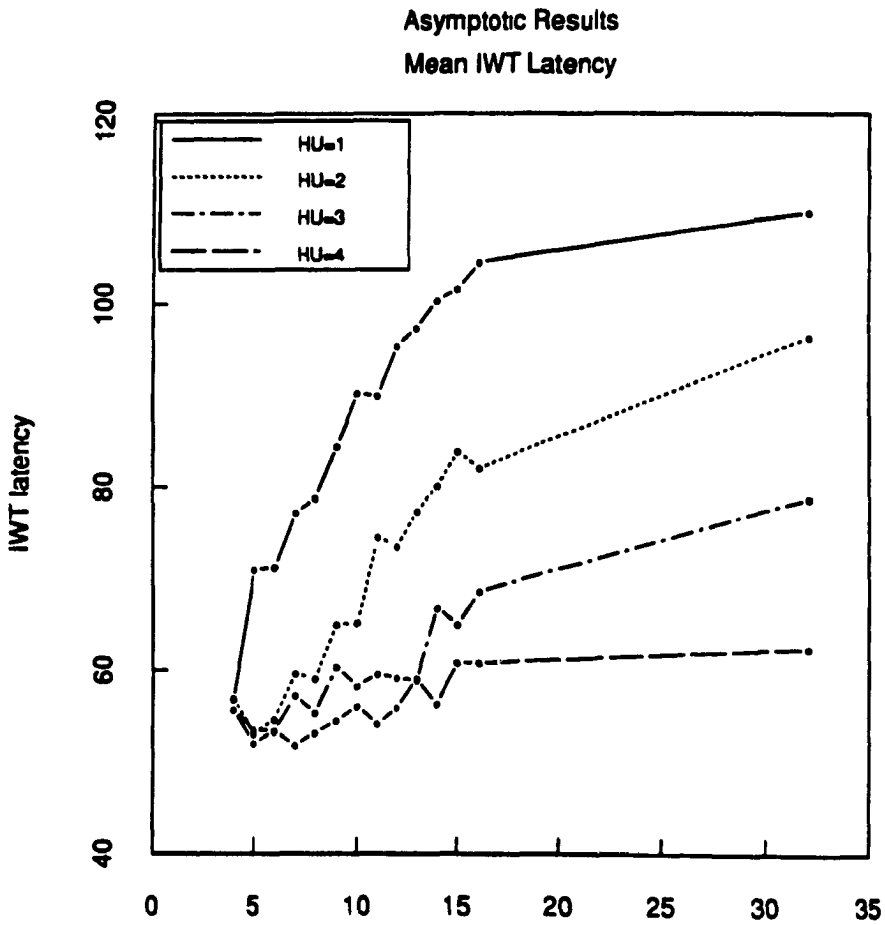

Figure 4.21

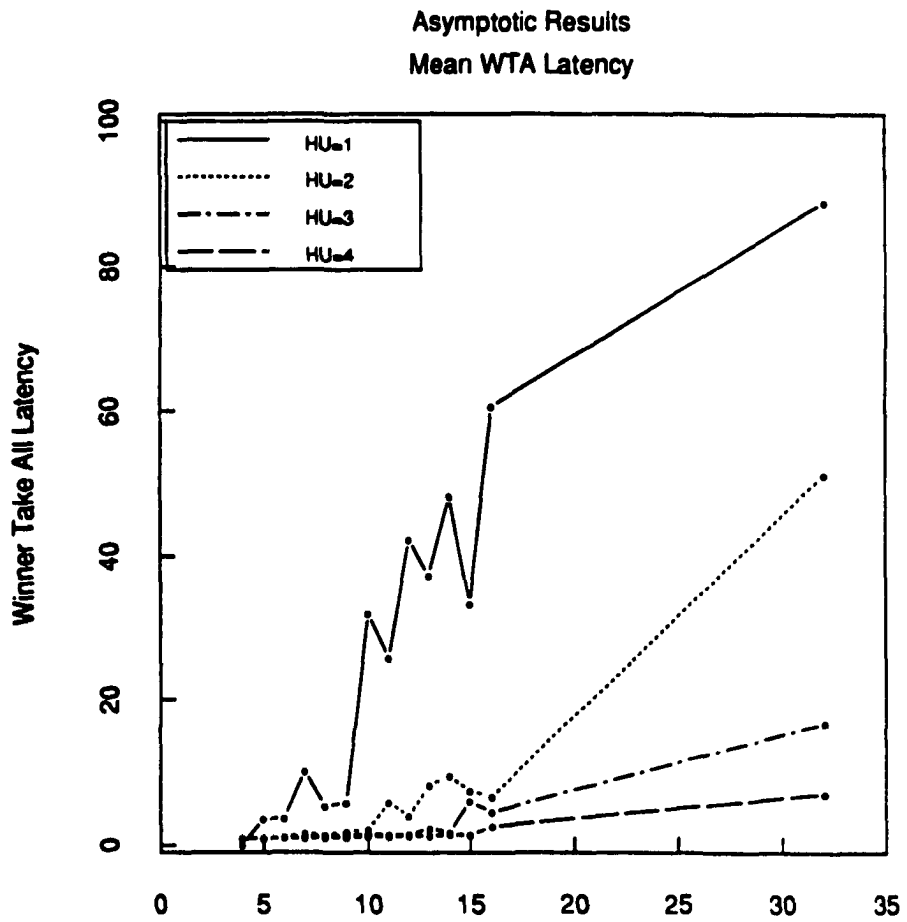

Figure 4.22

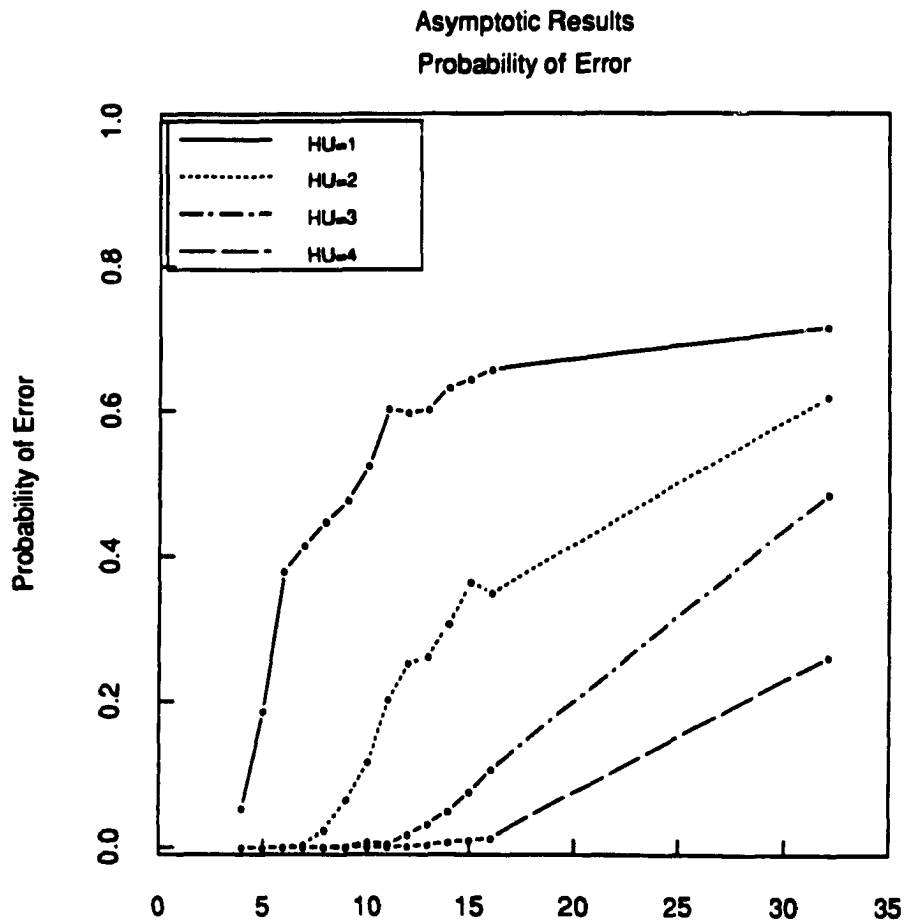

Figure 4.23

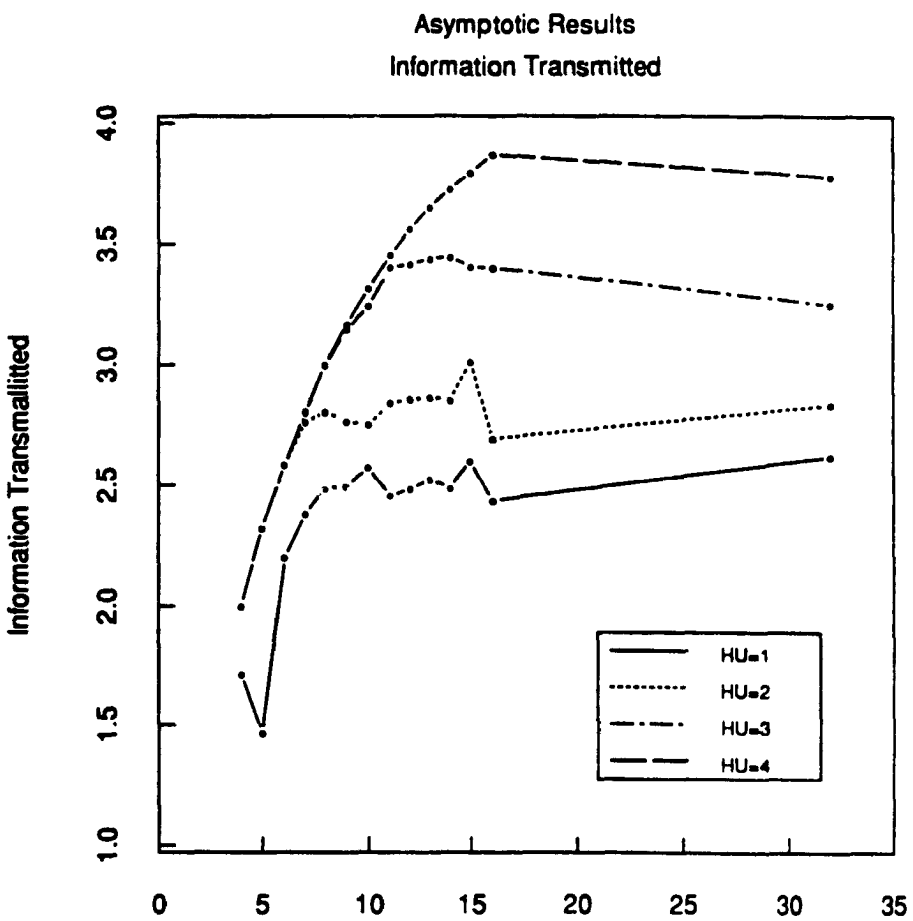

Figure 4.24

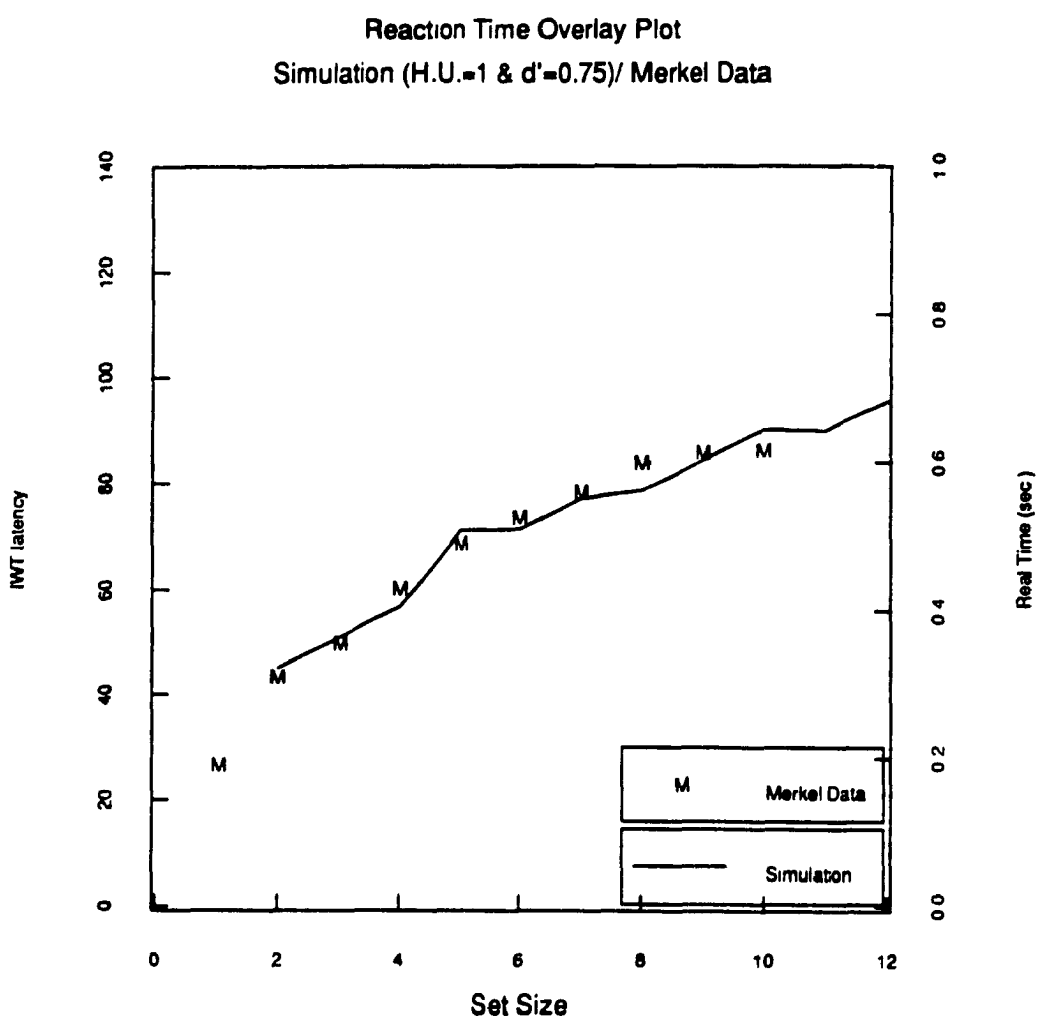

Figure 4.25

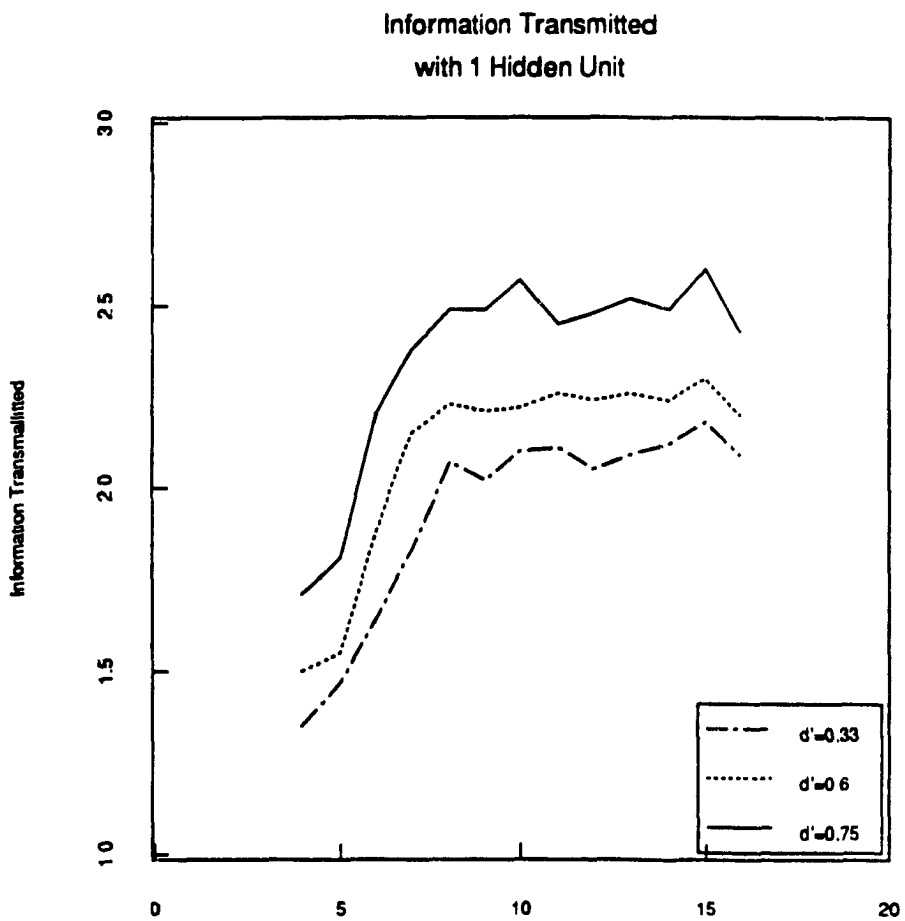

Figure 4.26

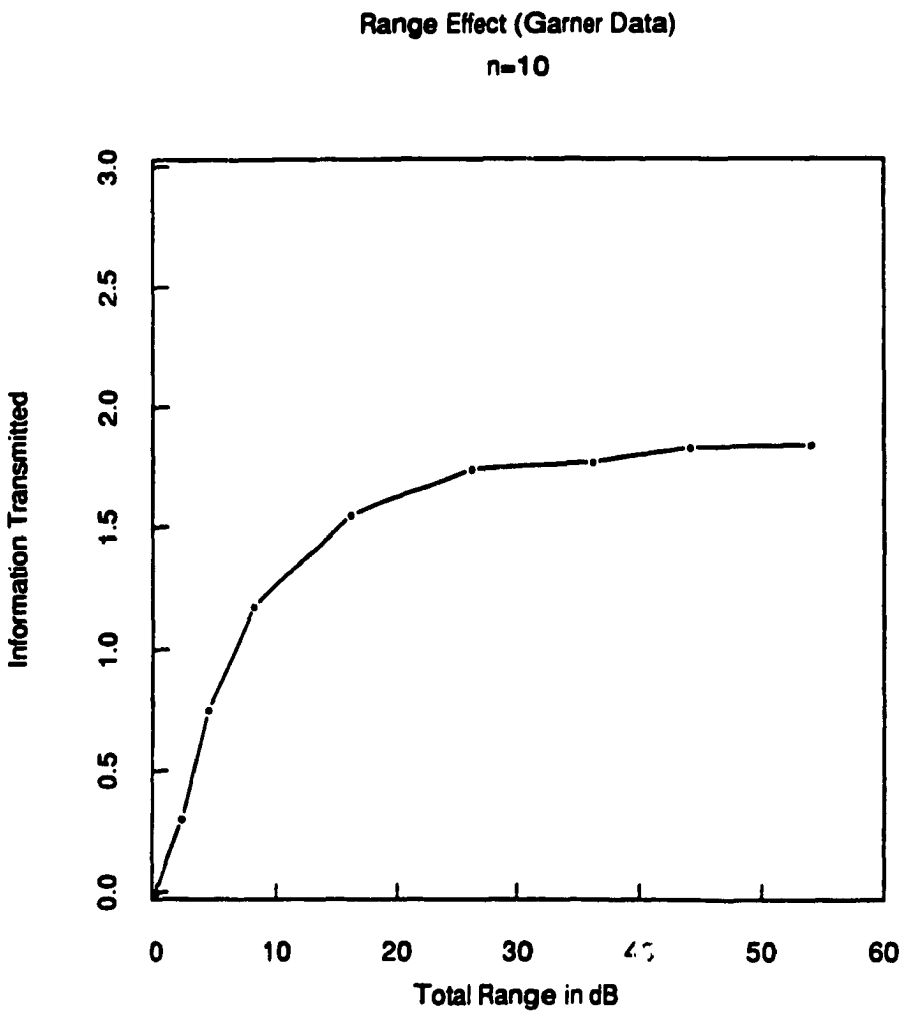

Figure 4.27

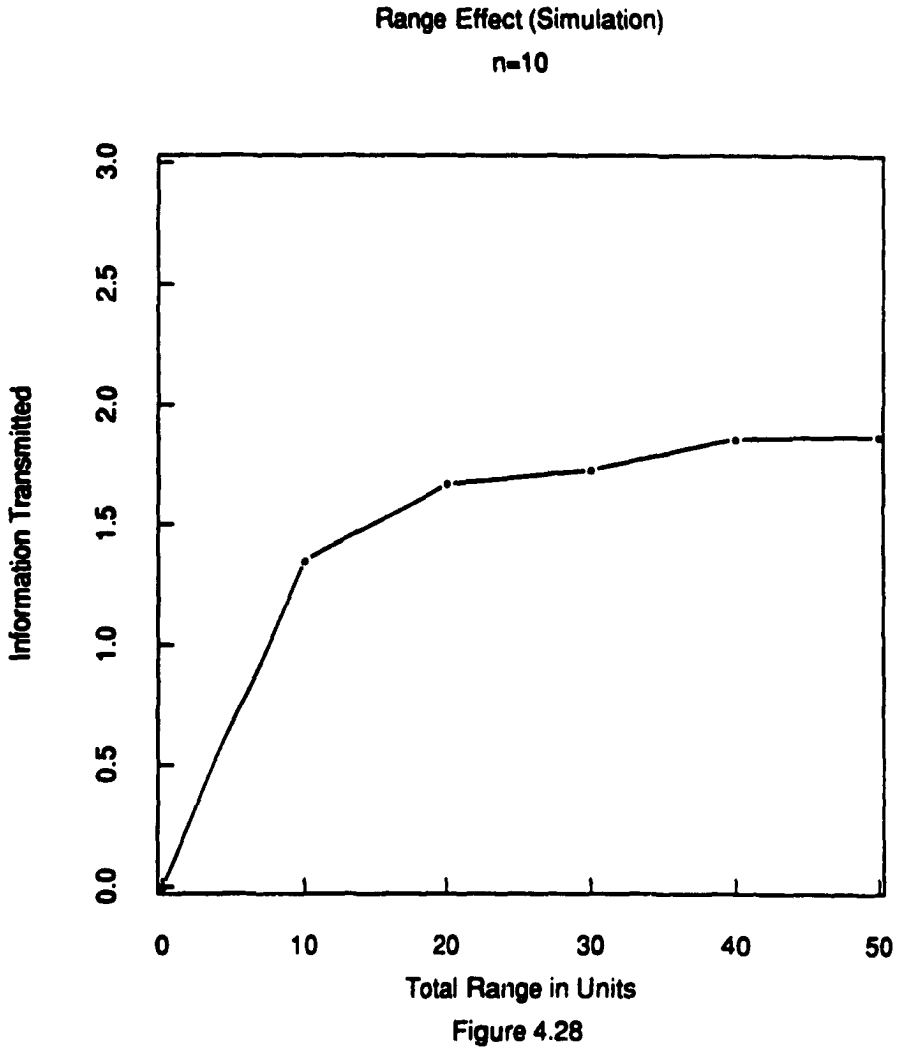

Figure 4.28

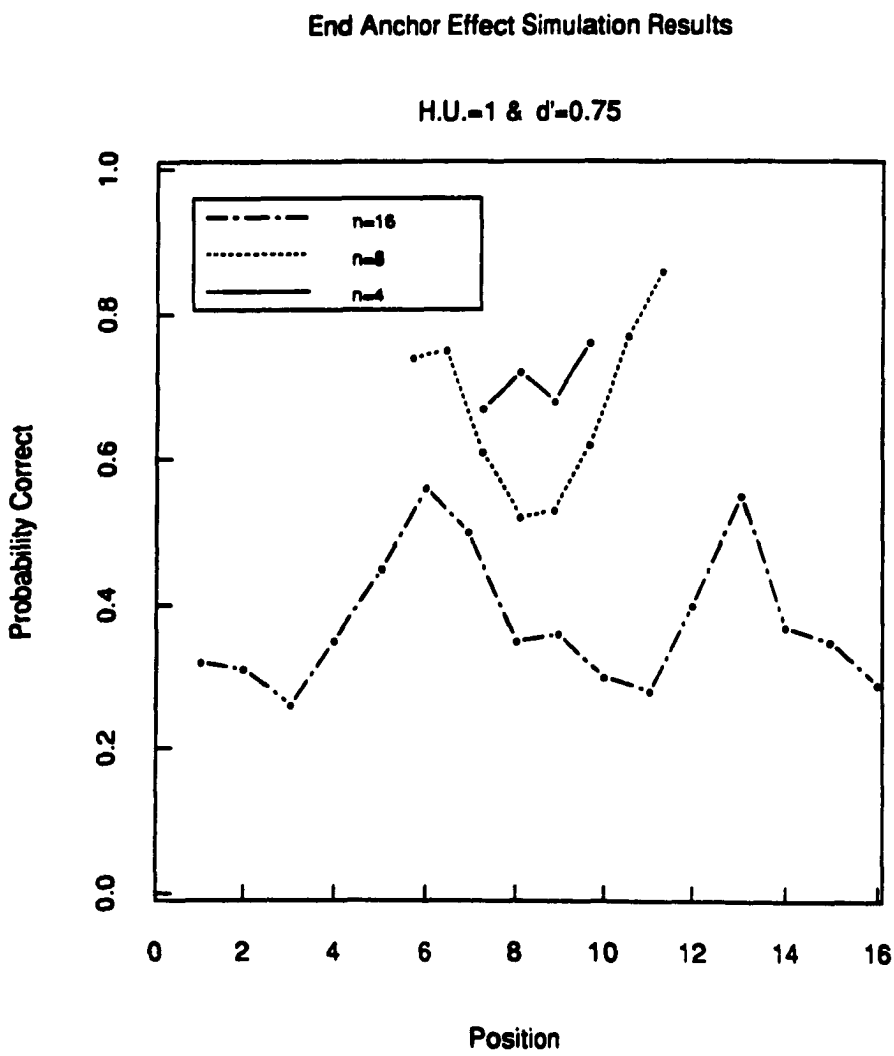

Figure 4.29

Correlation Plot IWT/MSE With 1 Hidden Unit

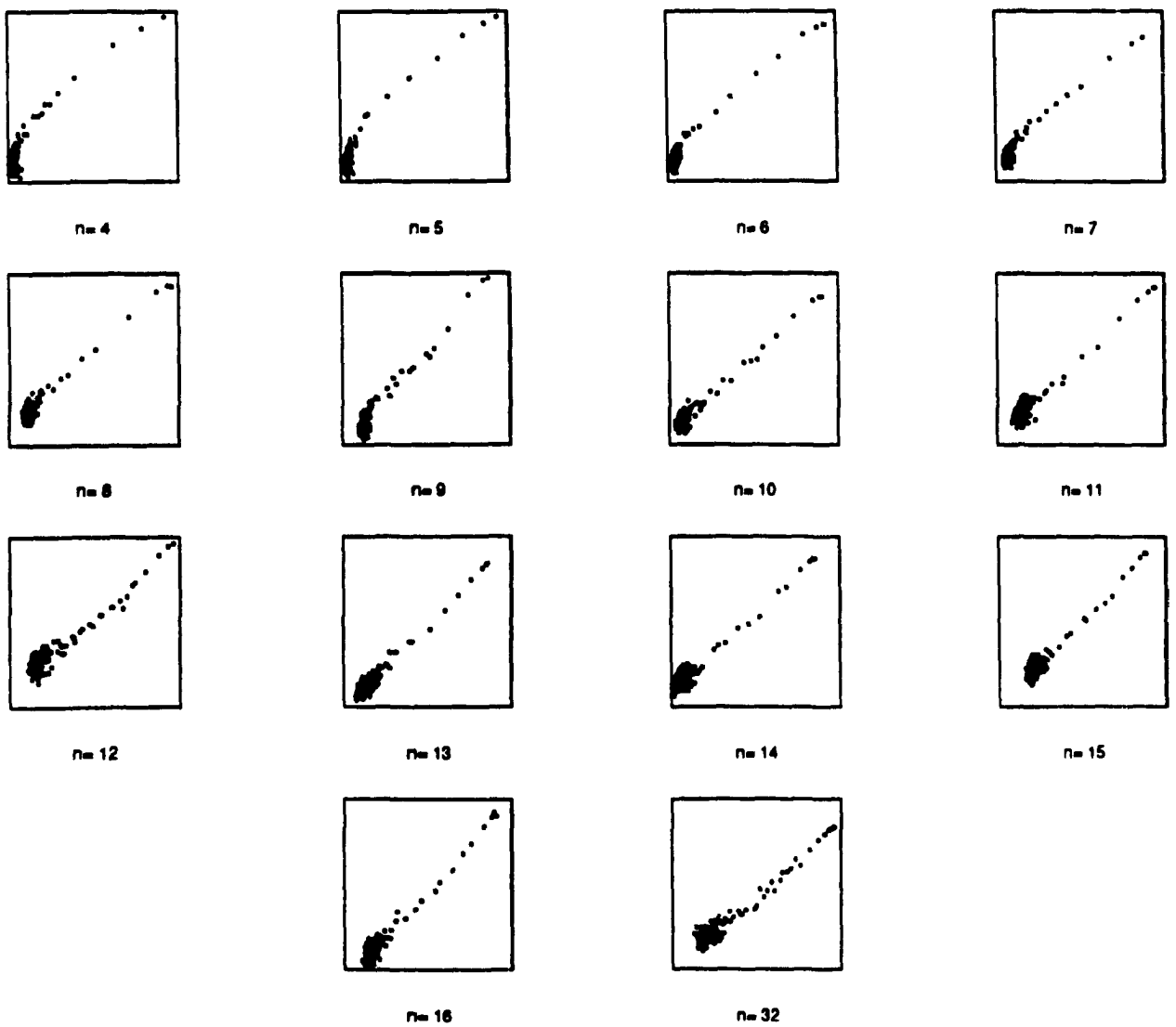

Figure 4.30

Correlation Plot IWT/MSE With 2 Hidden Units

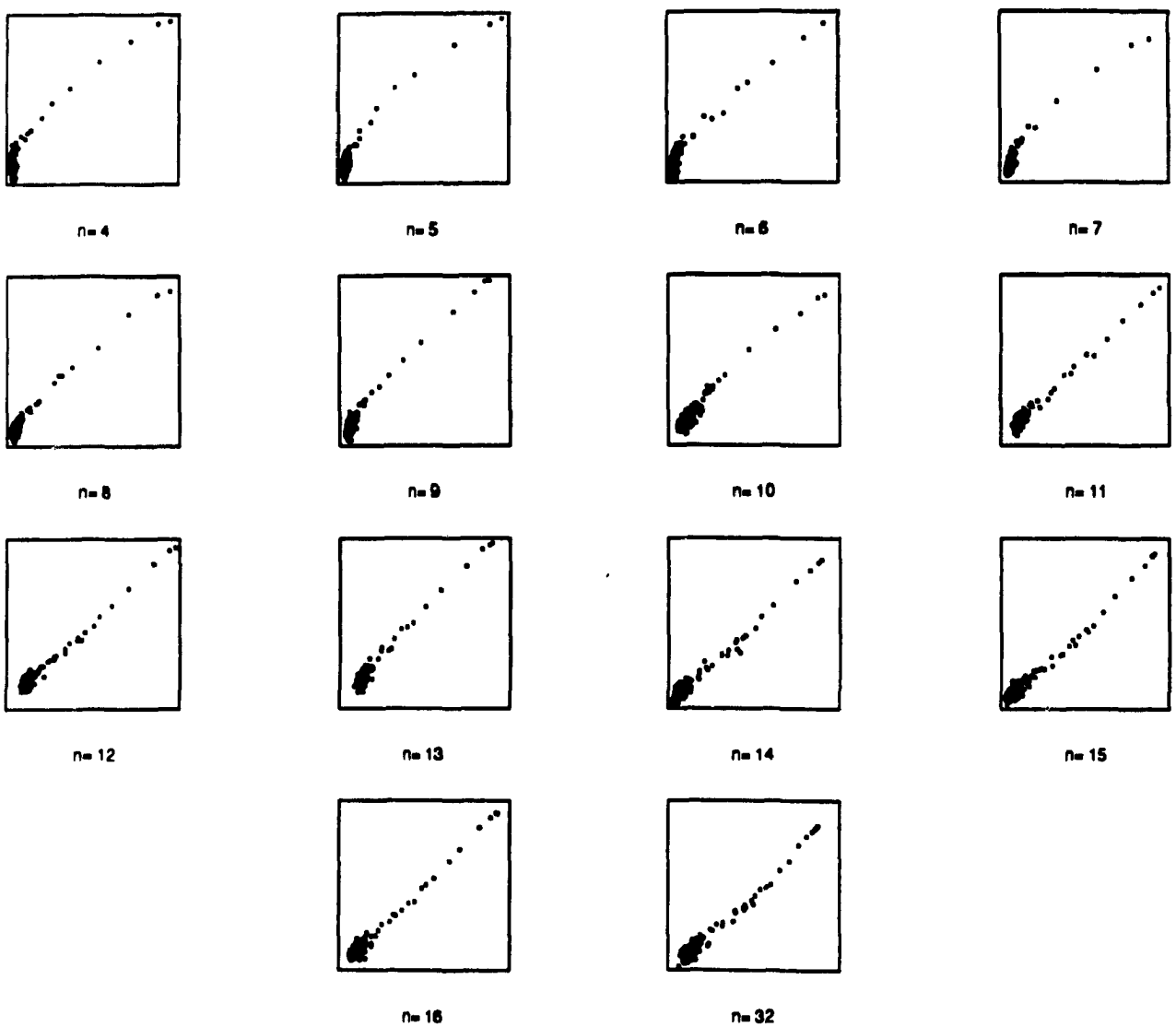

Figure 4.31

IWT Latency Distributions, H.U. = 1, d'=0.75

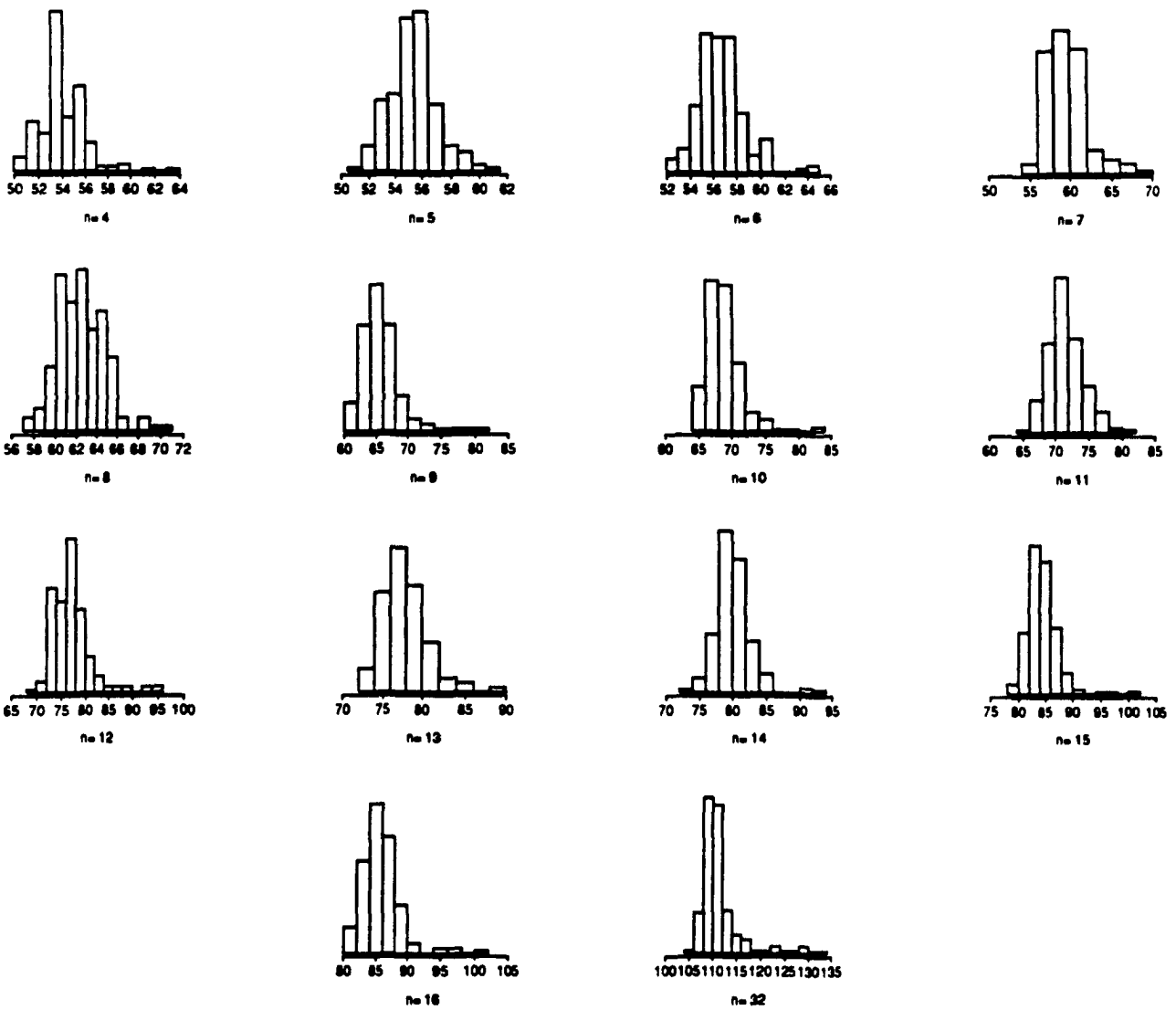

Figure 4.32

IWT Latency Distributions, H.U. = 2, d'=0.75

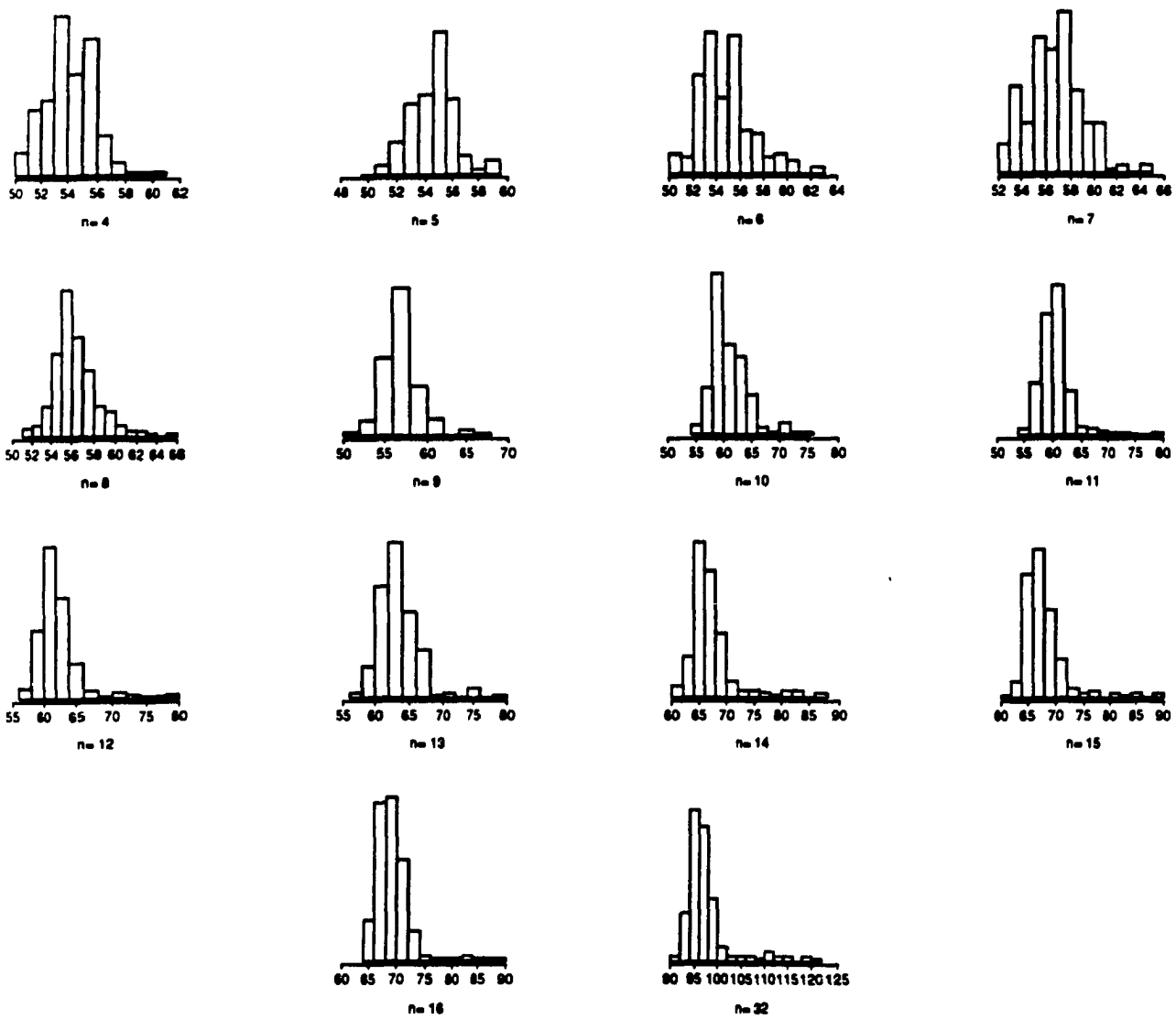

Figure 4.33

QQ-Log Normal Prob. Plot for IWT Distributions (1 H.U.)

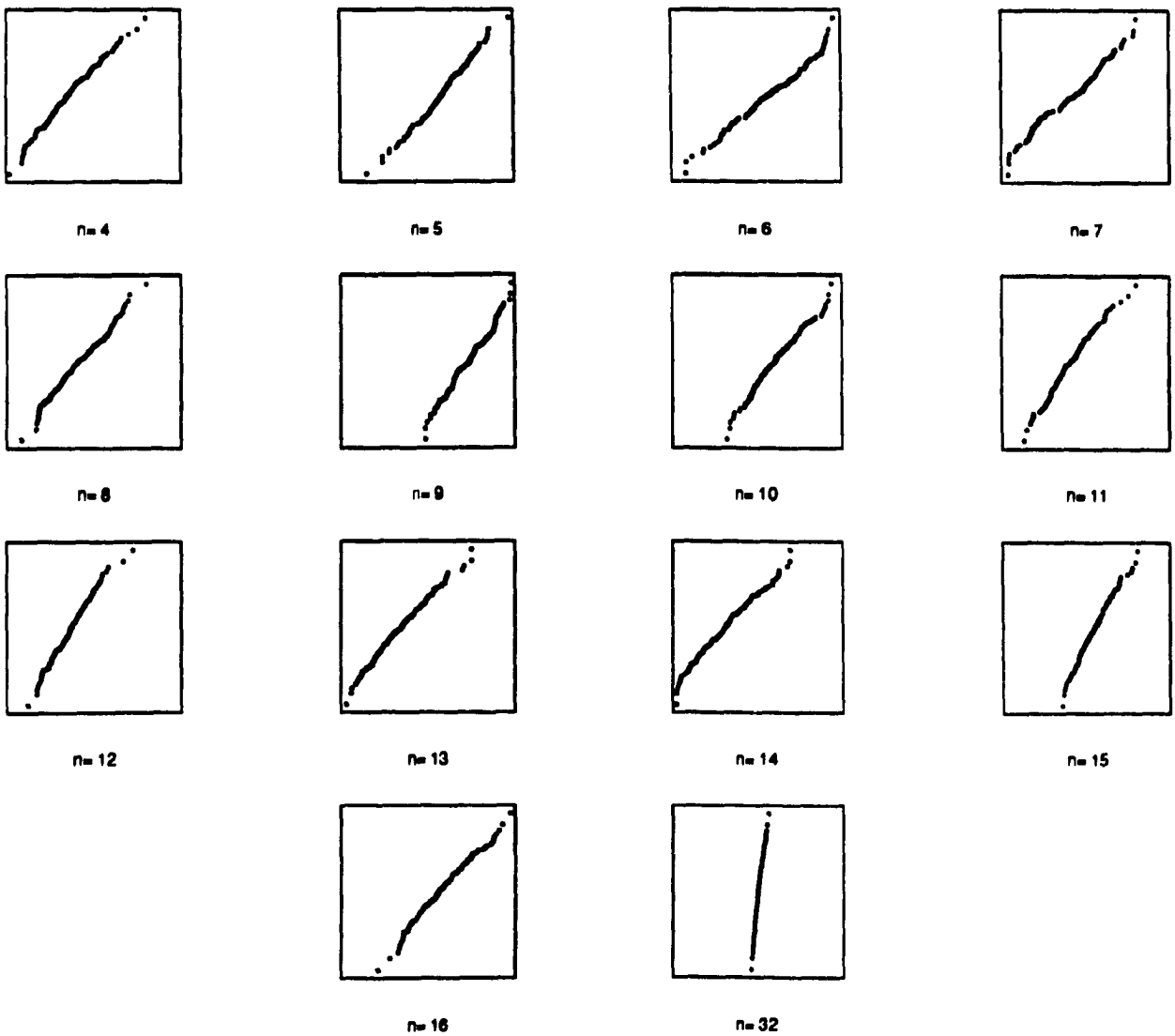

Figure 4.34

QQ-Log Normal Prob. Plot for IWT Distributions (2 H.U.)

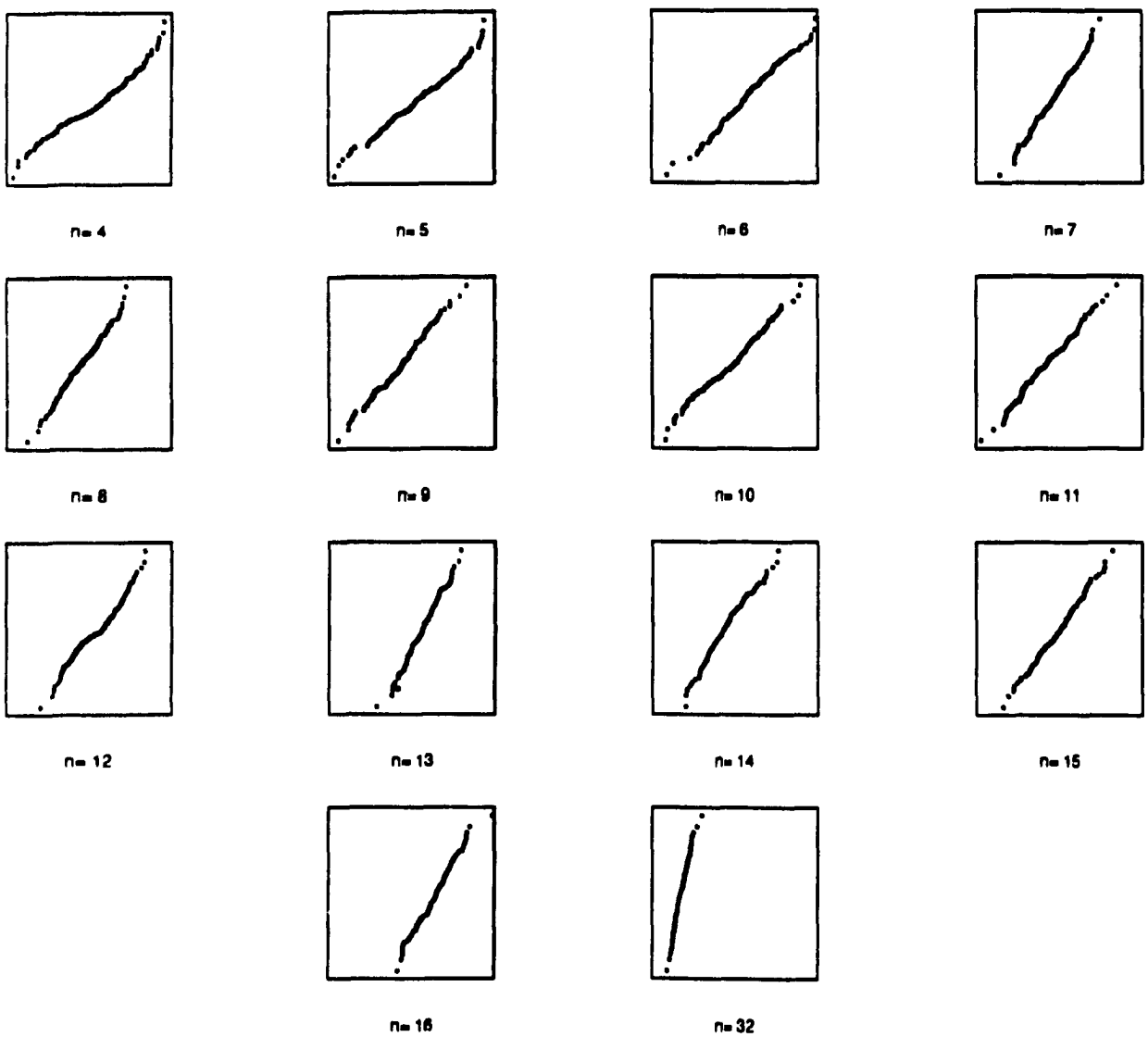

Figure 4.35

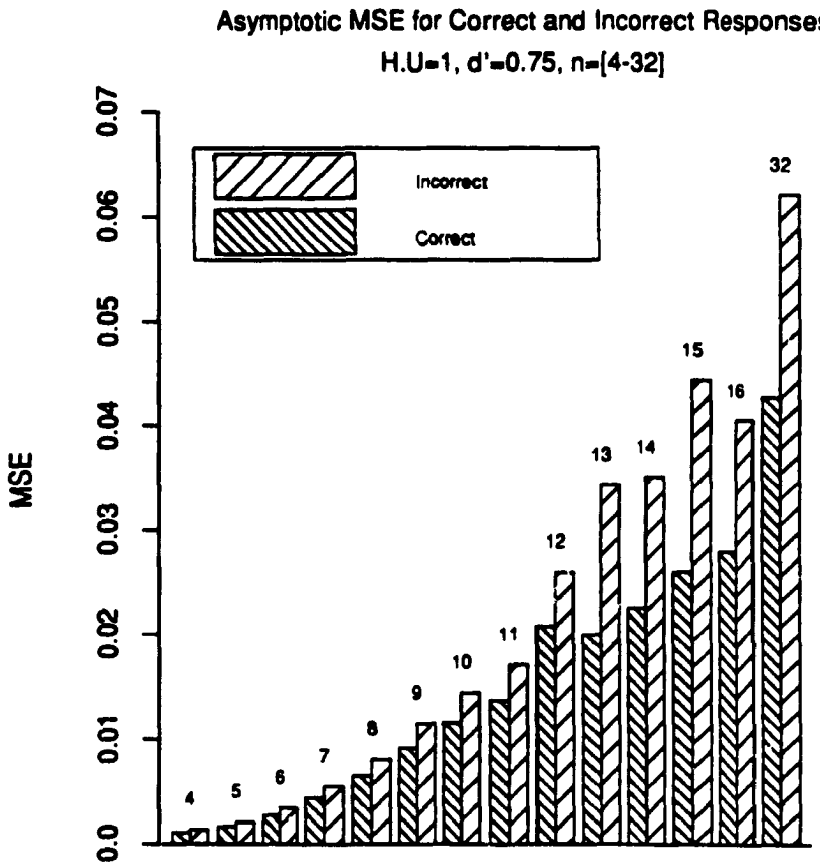

Figure 4.36

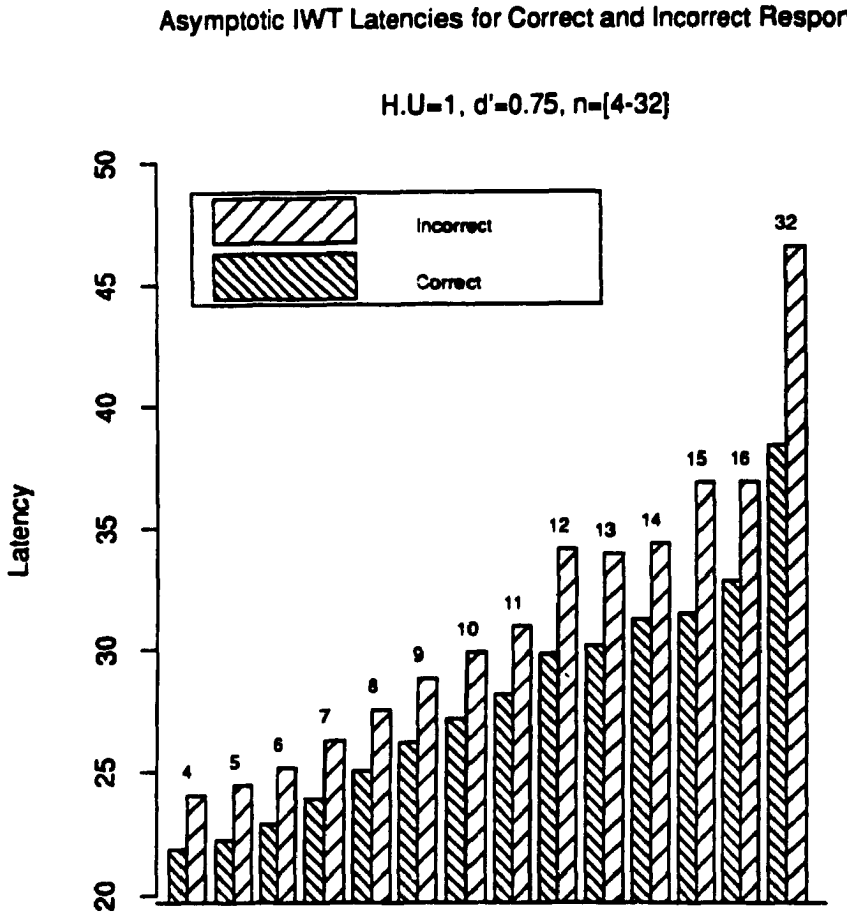

Figure 4.37

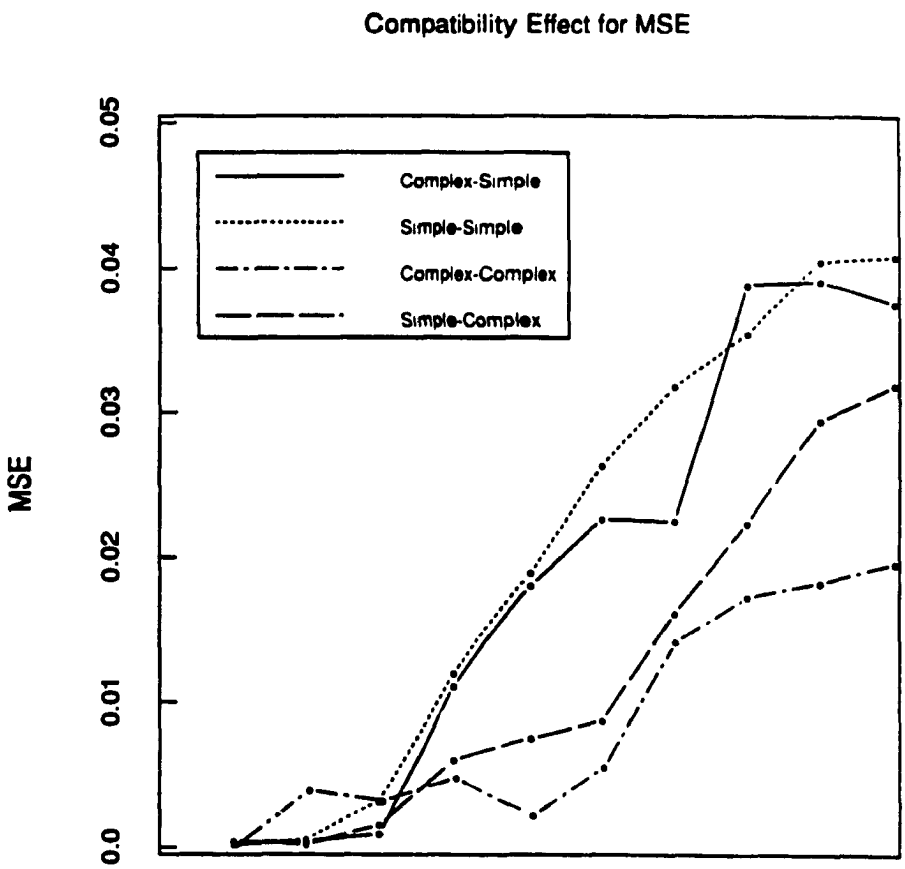

Figure 4.38

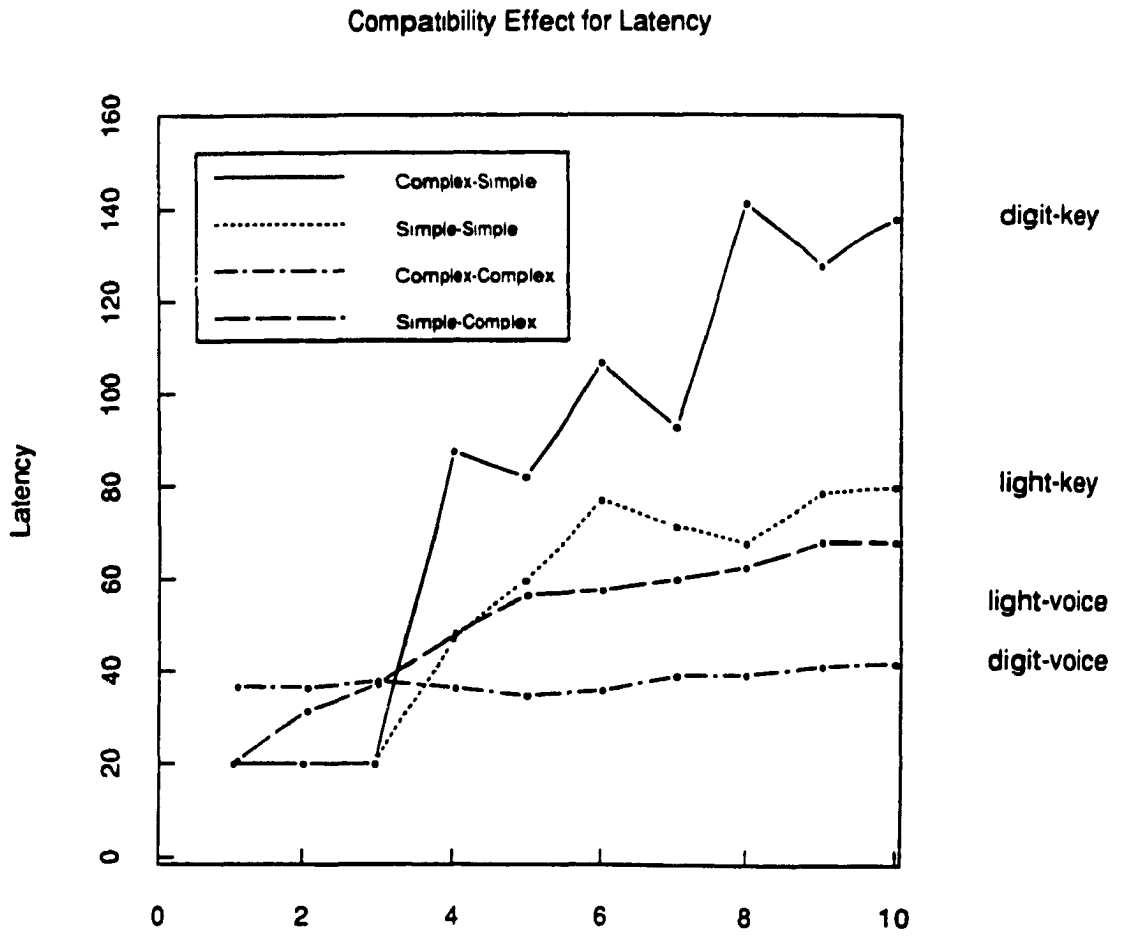

Figure 4.39

The 16 Two-Dimensional Stimuli

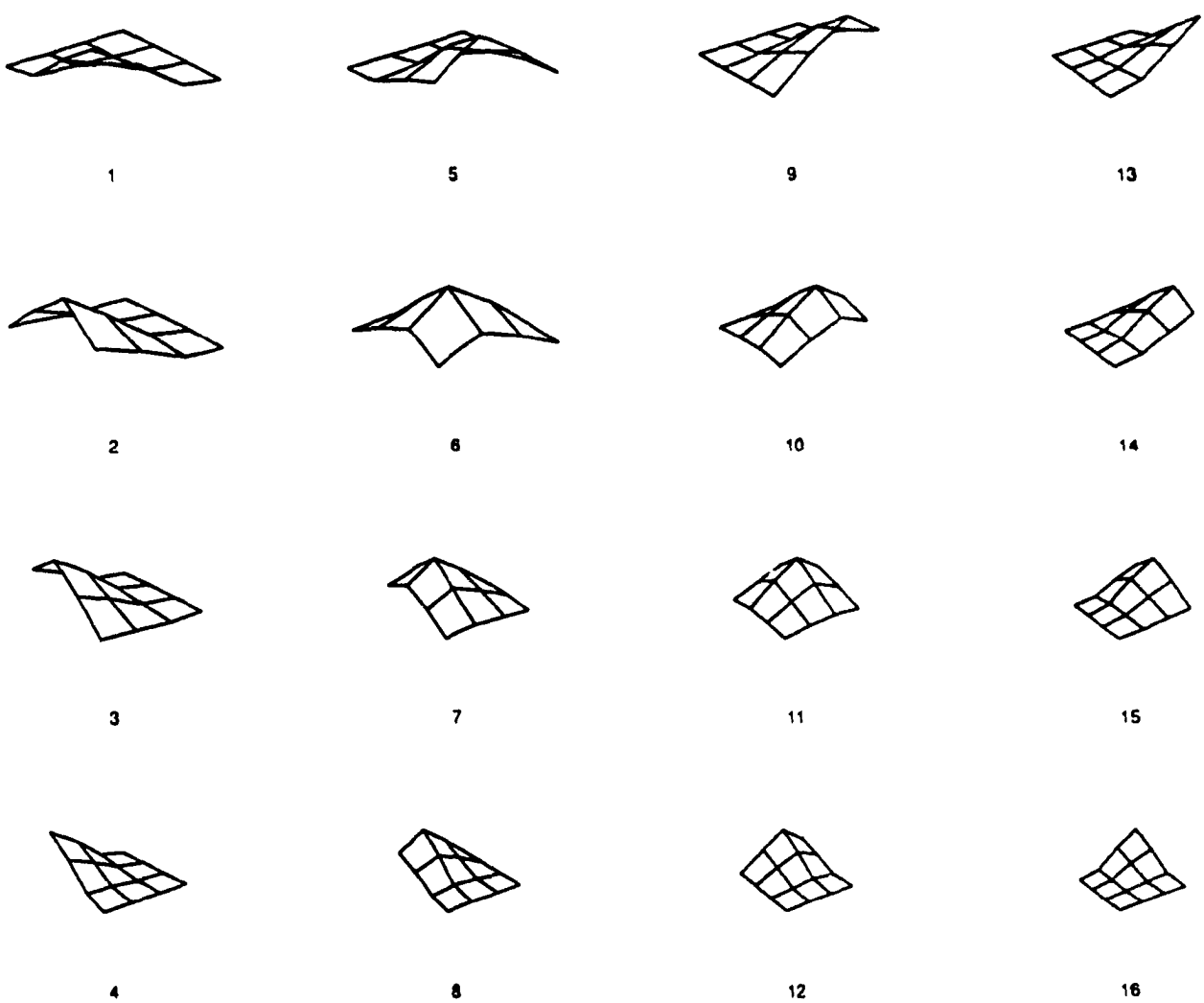

Figure 4.40

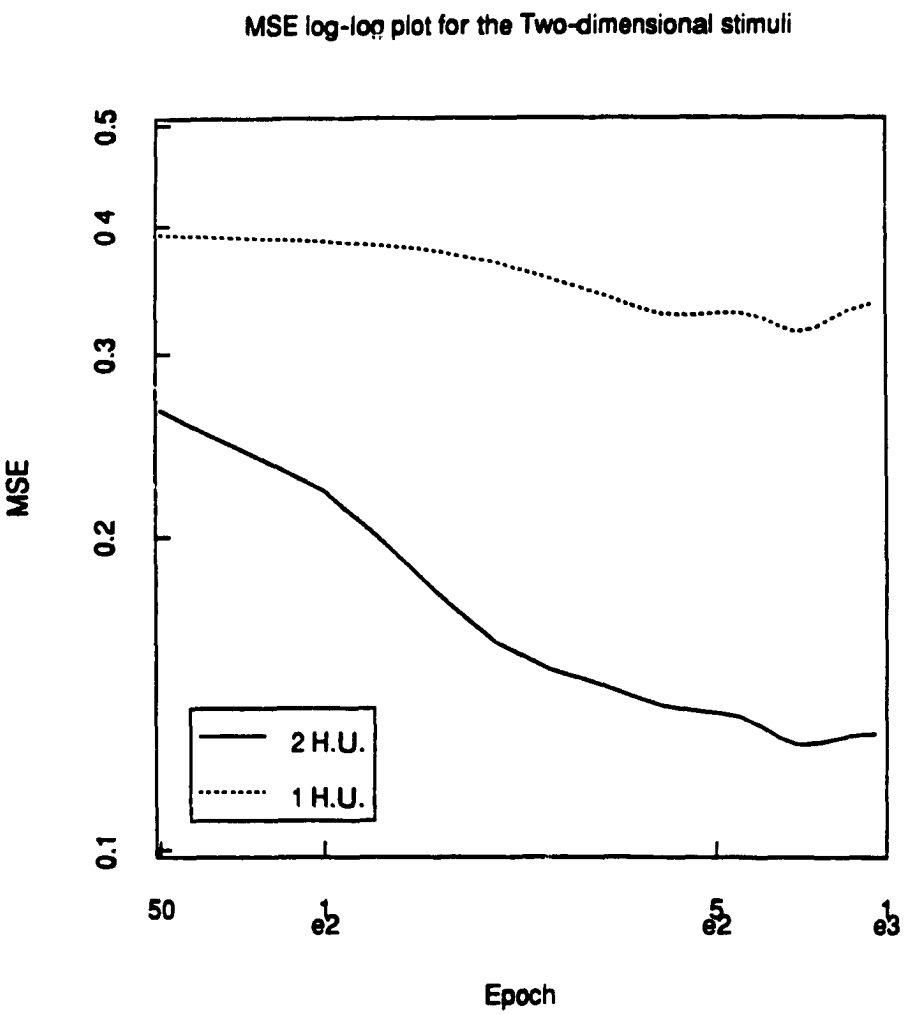

Figure 4.41

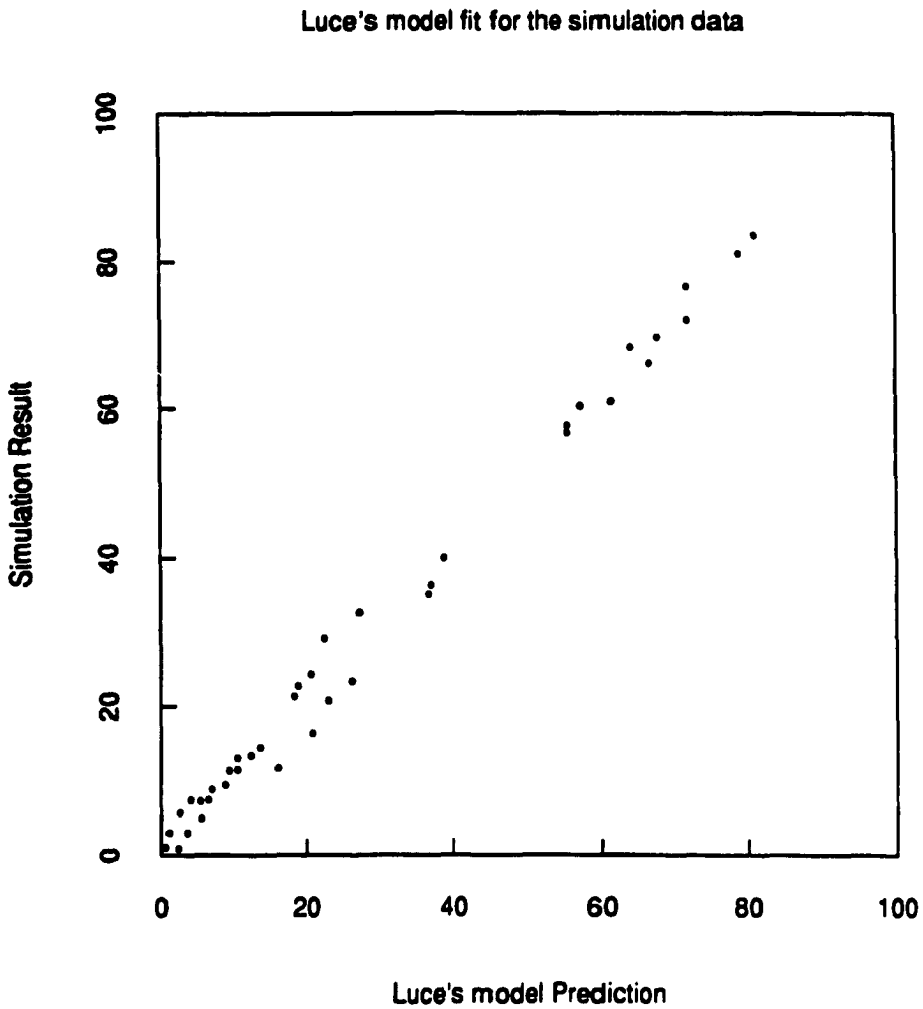

Figure 4.42

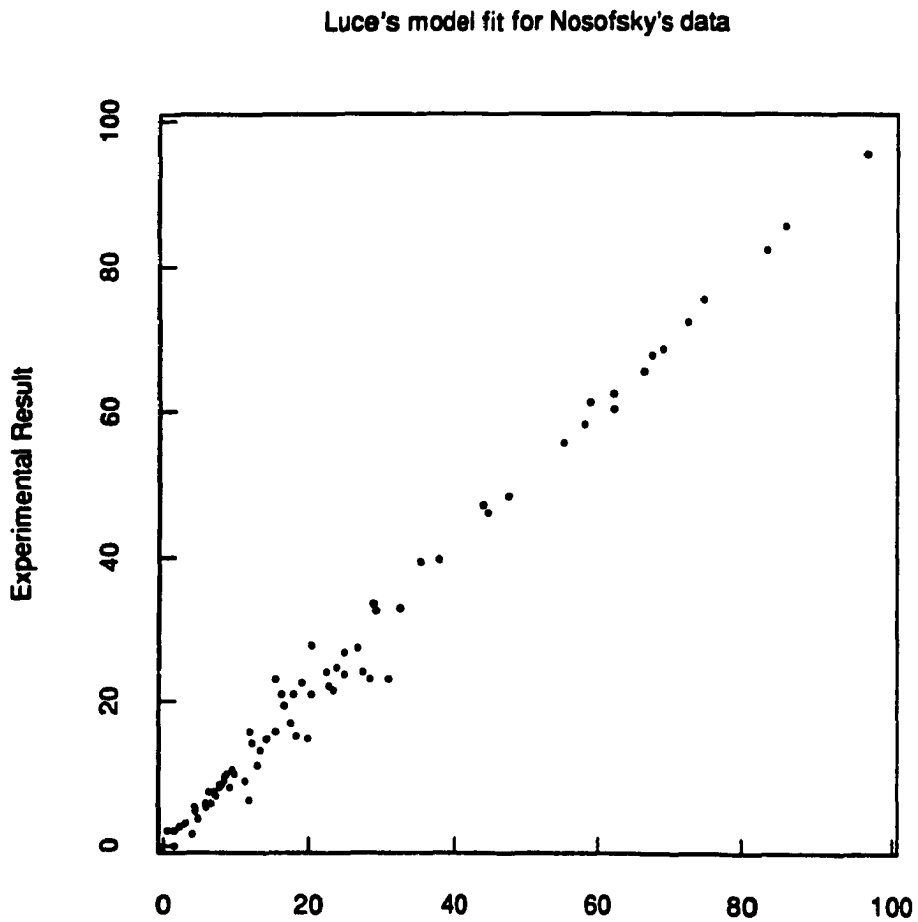

Figure 4.43

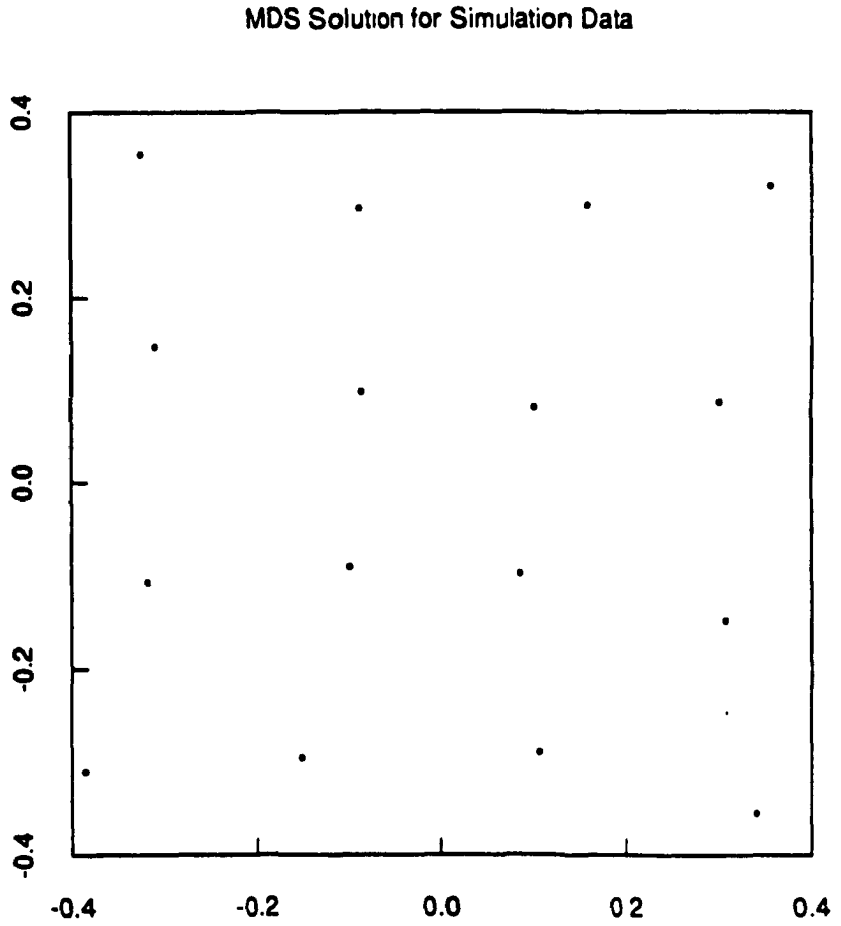

Figure 4.44

MDS-Choice Model Configuation for Nosofsky's Data

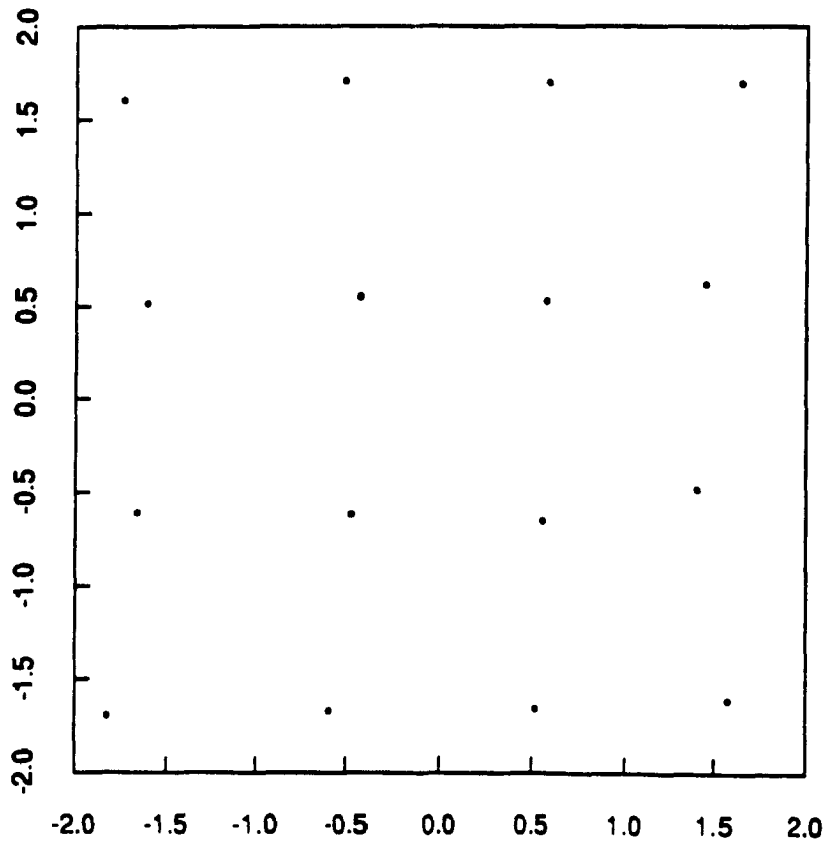

Figure 4.45

Single Hidden Unit Representation
d'=0.75

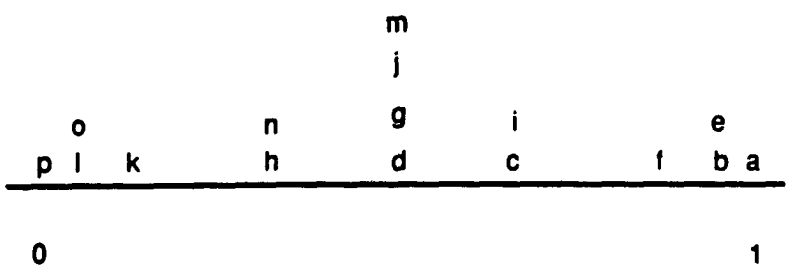

Figure 4.46

Internal Representation for the Two-dimensional Stimuli

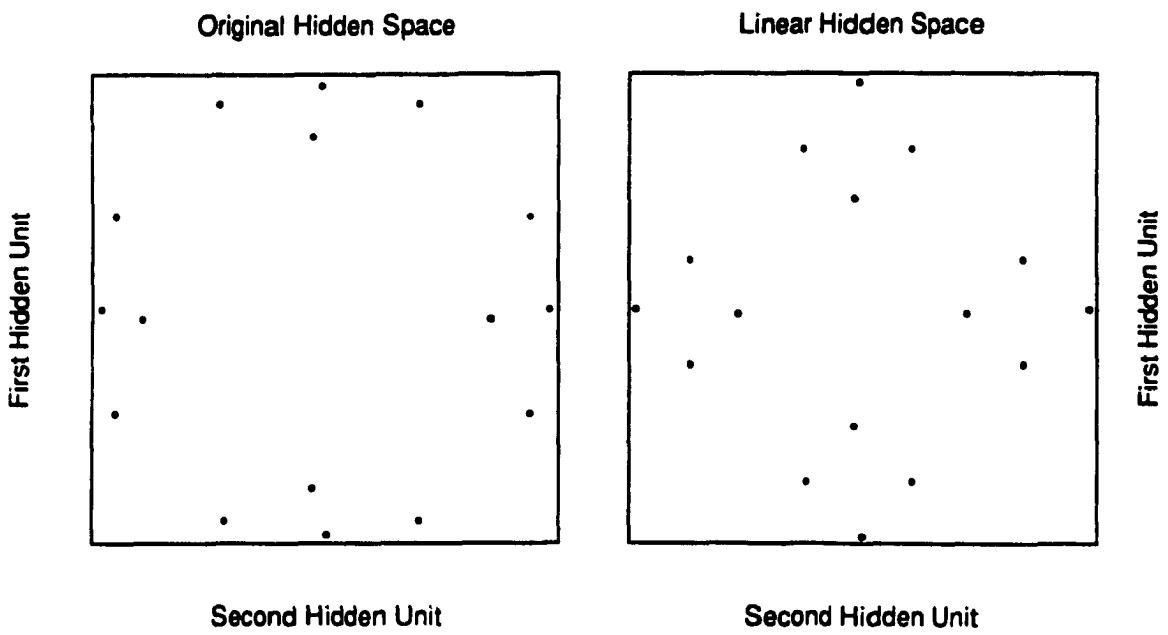

Figure 4.47

A schematic view of the network

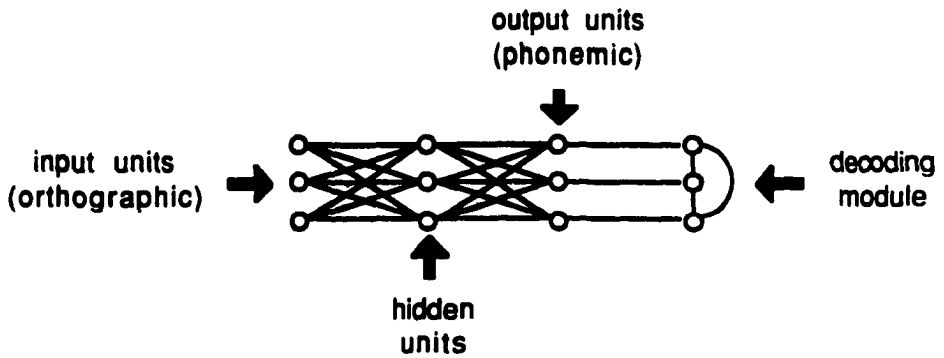

Figure 5.1

Length of the state vector as a function of iterations in the decoding module after presentation of the word "AISLE"

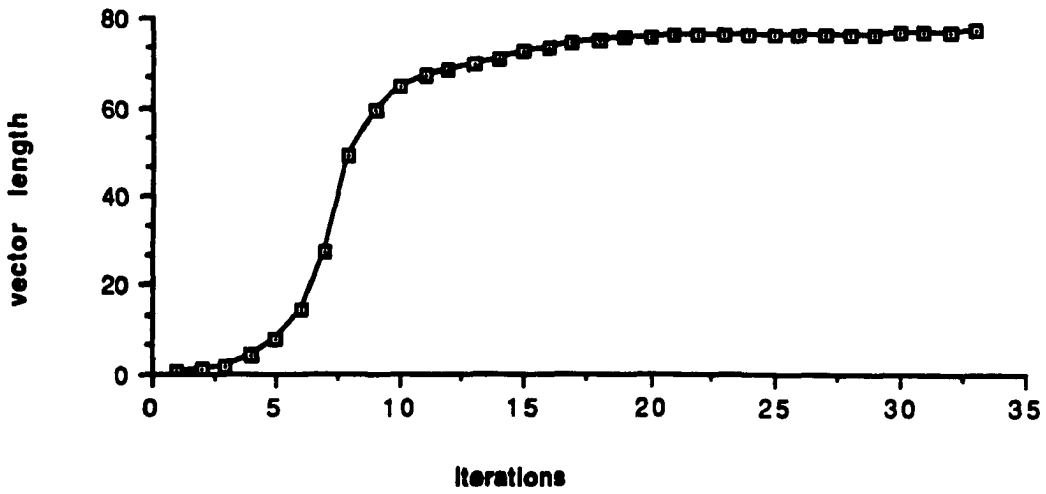

Figure 5.2

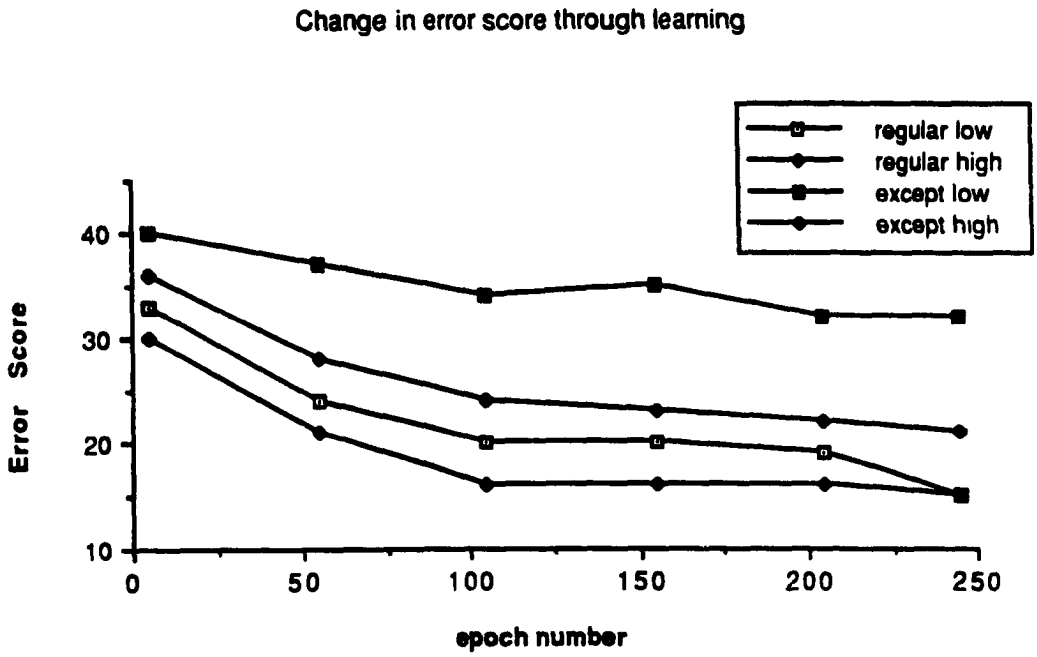

Figure 5.3

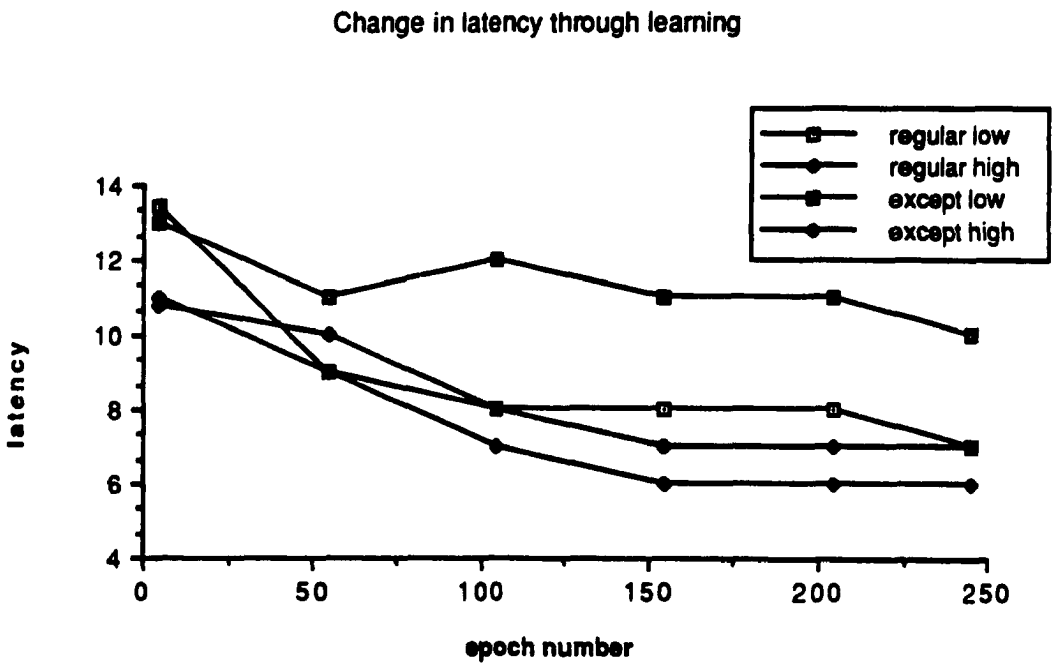

Figure 5.4

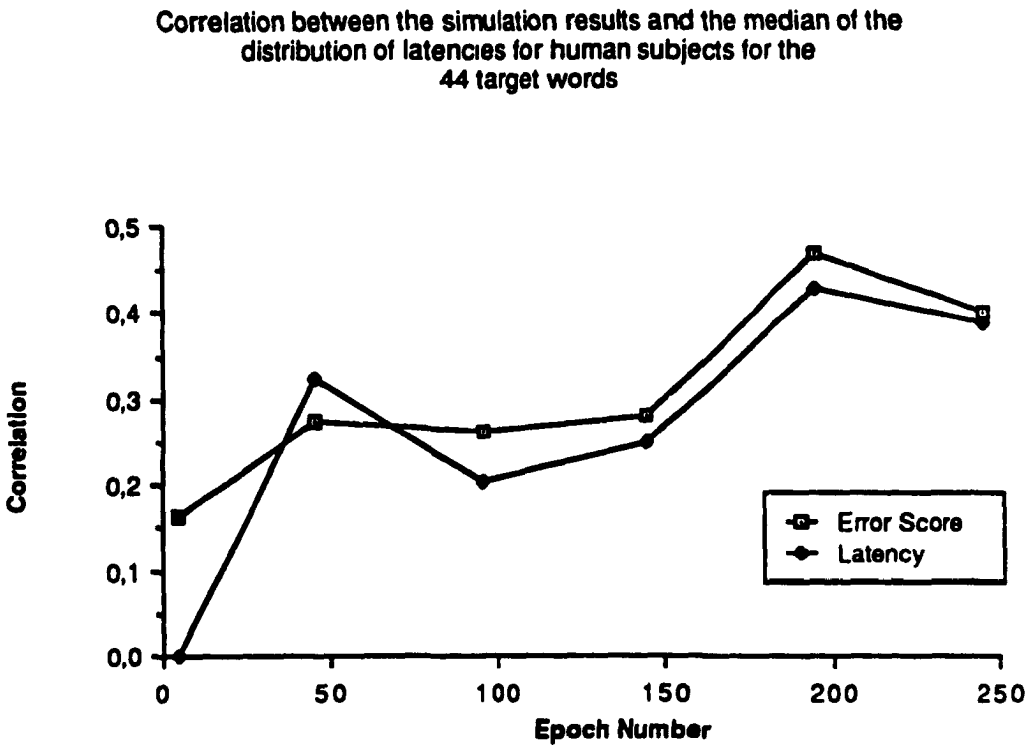

Figure 5.5

n	BP	MV-BP
4	1.971361	2.000000
8	2.173646	2.492983
16	2.453814	3.413347
32	2.689043	4.634162

Information Transmitted
for the BP and MV-BP Algorithms
with n=[4-32]

Table 4.1

n	BP	MV-BP
4	0.003500	0.000000
8	0.301500	0.259682
16	0.731500	0.579807
32	0.933562	0.839017

Probability of Error
for the BP and MV-BP Algorithms
with n=[4-32]

Table 4.2

OO-Condition, n=4

500	0	0	0
6	493	1	0
0	0	500	0
0	0	0	500

OG-Condition, n=4

365	135	0	0
32	468	0	0
0	0	500	0
0	0	35	365

GO-Condition, n=4

498	2	0	0
0	414	84	2
1	53	446	0
0	0	18	482

GG-Condition, n=4

500	0	0	0
0	498	2	0
0	9	491	0
0	0	0	500

00-Condition, n=8

497	0	3	0	0	0	0	0
0	499	1	0	0	0	0	0
0	5	495	0	0	0	0	0
0	0	0	493	0	0	7	0
32	1	14	22	422	0	0	9
0	1	0	0	0	497	2	0
0	0	0	6	0	0	493	1
0	3	0	0	41	43	30	383

0G-Condition, n=8

482	15	0	0	3	0	0	0
65	409	26	0	0	0	0	0
0	6	457	36	1	0	0	0
0	0	37	456	6	0	0	1
16	0	0	13	323	44	33	71
11	0	0	0	7	425	57	0
0	0	0	0	8	40	402	50
0	0	0	8	10	0	59	423

GO-Condition, n=8

443	57	0	0	0	0	0	0
27	463	10	0	0	0	0	0
0	2	473	25	0	0	0	0
0	0	28	453	19	0	0	0
0	0	0	8	476	16	0	0
0	0	0	0	25	470	5	0
0	0	0	0	0	7	484	9
0	0	0	0	0	0	65	435

GG-Condition, n=8

490	10	0	0	0	0	0	0
8	468	24	0	0	0	0	0
0	8	476	16	0	0	0	0
0	0	35	435	30	0	0	0
0	0	0	24	459	17	0	0
0	0	0	0	39	437	24	0
0	0	0	0	0	7	473	20
1	0	0	0	0	0	4	496

Stimulus/reponse matrices
Table 4.3

00-Condition, n=16

500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	28	133	11	17	45	64	27	155	20	0
20	288	0	0	0	0	0	0	36	0	0	0	0	0	0	176
27	119	165	0	0	0	0	0	0	0	0	0	0	88	35	87
30	38	51	144	0	0	0	0	15	24	22	25	20	60	13	58
22	29	45	70	166	0	0	0	21	23	13	15	1	30	16	49
15	14	31	27	76	135	0	0	27	31	12	26	0	22	26	58
16	1	0	4	54	58	219	0	16	4	7	28	0	22	37	34
0	0	0	0	0	0	0	0	500	0	0	0	0	0	0	0
20	1	0	1	40	42	68	210	58	6	5	10	1	6	31	1
37	12	16	11	37	13	0	0	0	0	373	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	500	0	0	0	0
23	0	0	1	71	9	3	80	2	192	40	36	0	38	5	0
10	16	21	10	17	111	0	17	3	61	30	27	170	0	8	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	500	0
12	0	0	0	0	0	0	9	11	147	16	53	44	179	29	0

0G-Condition, n=16

309	120	0	0	21	14	1	0	0	0	0	0	34	0	0	0
83	296	30	88	0	0	1	0	0	0	0	0	0	0	0	2
0	284	15	120	51	26	1	2	0	0	0	0	0	0	0	1
0	4	14	150	241	19	16	0	0	0	0	0	0	0	37	19
17	12	2	20	65	65	134	49	20	0	0	23	0	0	49	3
7	4	0	103	58	79	151	90	1	0	0	0	0	0	7	0
0	0	33	25	1	13	55	367	4	1	0	0	0	0	0	0
0	0	1	17	0	0	0	365	117	0	0	0	0	0	0	0
0	0	0	0	0	53	61	70	285	32	0	0	0	0	0	0
0	0	0	0	5	15	0	0	61	310	102	7	0	0	0	0
0	0	0	0	0	1	0	0	1	99	324	75	0	0	0	0
0	0	0	0	0	17	0	0	0	1	76	346	56	3	1	0
0	0	0	1	4	0	0	0	0	0	0	81	360	54	1	0
0	0	0	0	1	0	0	0	0	0	0	0	111	256	132	0
0	0	0	0	0	0	0	0	0	0	0	0	4	20	474	2
7	12	1	2	0	0	0	0	0	0	0	0	0	3	34	441

G0-Condition, n=16

332	2	9	2	0	89	26	3	0	23	0	0	0	15	0	0
0	0	0	137	0	107	0	0	3	0	0	57	58	18	13	106
26	225	0	129	63	11	0	0	0	0	16	0	21	8	0	0
22	2	0	0	274	2	7	2	0	0	0	0	0	22	13	158
30	110	0	0	0	288	0	21	5	0	0	26	0	0	18	3
295	68	0	97	0	0	0	14	0	0	0	2	0	0	14	10
0	0	3	0	29	361	0	19	0	17	14	0	42	0	0	15
0	22	19	24	3	6	15	37	6	0	0	0	0	0	219	148
0	0	1	1	0	1	4	123	327	42	0	0	0	0	2	0
14	8	8	18	16	5	0	23	62	313	33	0	0	0	0	0
0	12	0	21	17	16	0	0	0	2	430	3	0	0	0	0
19	0	20	27	0	39	9	0	0	0	22	349	15	0	0	0
19	0	0	24	23	28	0	0	0	0	0	2	398	6	0	0
9	0	0	0	9	12	27	1	0	0	0	0	23	405	14	1
0	0	5	0	0	0	17	1	4	0	0	0	0	107	361	5
58	0	41	0	43	34	62	62	54	3	0	0	0	6	118	19

GG-Condition, n=16

239	39	102	60	33	6	0	0	1	0	0	2	0	1	9	7
4	108	327	26	2	0	0	0	0	0	0	0	0	0	33	0
0	35	421	42	0	0	0	0	0	0	0	0	0	0	2	0
1	4	258	207	24	6	0	0	0	0	0	0	0	0	0	0
2	4	22	229	134	55	52	2	0	0	0	0	0	0	0	0
0	0	0	44	124	82	209	39	2	0	0	0	0	0	0	0
0	0	0	1	11	51	254	181	2	0	0	0	0	0	0	0
0	0	0	0	0	4	82	349	65	0	0	0	0	0	0	0
0	0	0	0	0	0	0	90	332	69	8	1	0	0	0	0
0	0	0	0	0	0	0	4	111	247	134	4	0	0	0	0
0	0	0	0	0	0	0	0	2	84	314	100	0	0	0	0
0	0	0	0	0	0	0	0	0	1	91	362	43	3	0	0
0	0	0	0	0	0	0	0	0	0	1	141	277	73	8	0
0	0	0	0	0	0	0	0	0	0	0	1	86	239	174	0
1	1	0	0	0	0	0	0	0	0	0	0	7	60	431	0
3	13	41	11	1	2	0	0	0	0	0	0	3	3	33	390

Stimulus/reponse matrices
Table 4.3 (continued)

n	OO	OG	GO	GG
8	2.68	2.58	2.21	2.41
16	2.60	2.57	2.32	2.58

Information Transmitted for the four filter conditions

Table 4.4

64	0	0	0	22	14	0	0	0	0	0	0	0	0	0	0
43	0	0	0	15	42	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	76	0	0	0	6	16	0	0	0	0	0
0	0	0	0	0	43	9	0	0	0	47	0	0	0	1	0
42	40	0	0	0	18	0	0	0	0	0	0	0	0	0	0
13	13	0	0	10	63	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	55	6	0	0	0	39	0	0	0	0	0
0	0	0	0	0	16	9	0	0	0	72	0	0	0	0	3
0	0	0	0	0	68	22	0	0	0	10	0	0	0	0	0
0	0	0	0	0	46	18	0	0	0	36	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	79	2	0	0	0	18
0	0	0	0	0	0	0	0	0	0	50	0	0	0	1	49
0	0	0	0	0	42	14	0	0	0	44	0	0	0	0	0
0	0	0	0	0	11	0	0	0	3	83	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	36	3	0	0	3	58
0	0	0	0	0	0	0	0	0	0	6	0	0	0	4	90

Simulation Stimulus/Response Matrix with 1 Hidden Unit

Table 4.5

72	7	0	0	16	5	0	0	0	0	0	0	0	0	0	0
61	26	0	0	6	7	0	0	0	0	0	0	0	0	0	0
0	0	34	55	0	0	4	7	0	0	0	0	0	0	0	0
0	0	8	78	0	0	4	10	0	0	0	0	0	0	0	0
66	6	0	0	22	3	0	0	2	1	0	0	0	0	0	0
35	24	2	0	26	12	0	0	0	0	1	0	0	0	0	0
0	2	21	34	0	5	16	20	0	0	1	1	0	0	0	0
0	0	3	62	0	0	5	28	0	0	0	2	0	0	0	0
0	0	0	0	4	0	0	0	23	2	0	0	58	13	0	0
0	0	0	0	3	1	0	0	35	11	1	0	26	20	3	0
0	0	0	0	0	0	1	1	0	2	19	21	0	3	14	39
0	0	0	0	0	0	0	6	0	0	9	28	0	0	3	54
0	0	0	0	0	0	0	0	11	3	1	0	74	11	0	0
0	0	0	0	0	0	0	0	5	2	0	0	68	24	1	0
0	0	0	0	0	0	0	0	0	2	3	8	0	5	28	54
0	0	0	0	0	0	0	0	0	0	1	7	0	0	10	82

Simulation Stimulus/Response Probabilities Matrix with 2 Hidden Units

Table 4.6

63	6	0	0	22	4	1	0	2	2	1	0	1	0	0	0
29	31	6	2	12	12	4	1	1	2	1	0	0	1	0	0
5	16	35	18	2	15	14	5	0	1	3	1	0	0	0	0
0	3	20	37	0	4	13	24	0	1	5	5	0	0	1	0
16	4	0	0	38	11	2	0	20	4	0	1	1	0	0	0
8	18	2	1	11	24	7	1	4	19	3	0	0	3	1	0
1	4	9	6	1	9	28	20	0	10	15	7	0	1	1	1
0	0	6	11	0	2	10	45	0	2	14	19	0	0	3	0
2	2	0	0	24	7	1	0	40	12	0	1	12	6	1	0
2	1	1	0	8	22	6	1	12	33	6	2	4	9	3	0
1	1	2	1	0	2	16	17	0	15	24	12	0	5	11	1
0	0	1	3	0	1	4	21	0	1	17	34	0	0	14	9
0	0	0	0	5	3	0	0	25	13	0	0	37	19	2	0
0	0	0	0	2	5	2	1	5	24	7	3	4	41	20	2
0	0	0	1	0	1	2	4	0	6	14	25	0	7	32	21
0	0	0	1	0	1	2	4	0	0	9	18	0	1	21	45

Nosofsky's Stimulus/Response Probabilities Matrix

Table 4.7