A Developmental Neural Model of Visual Word Perception

RICHARD M. GOLDEN

Brown University
Psychology Department, Box 1853
Providence, RI 02912

A neurally plausible model of how the process of visually perceiving a letter in the context of a word is learned, and how such processing occurs in adults is proposed. The model consists of a collection of abstract letter feature detector neurons and their interconnections. The model also includes a learning rule that specifies how these interconnections evolve with experience. The interconnections between neurons can be interpreted as representing the spatially redundant, sequentially redundant, and typographic information in letter string displays. Anderson, Silverstein, Ritz, and Jones's (1977) "Brain-State-in-a-Box" (BSB) neural mechanism is then used to implement the proposed model. The resulting system makes explicit qualitative predictions using both letter recognition accuracy and reaction time as dependent measures. In particular, the model offers an integrated explanation of some experiments involving manipulations of orthographic regularity, masking, case alternations, and experience with words. The similarities and differences between the model and models proposed by Adams (1979) and McClelland and Rumelhart (1981) are also discussed.

INTRODUCTION

Recently, a number of rather novel information processing models of perception and cognition have been proposed. These models are based upon the assumption that design constraints derived from neurophysiological considerations may provide useful insights about certain psychological phenomena. Such models are sometimes known as connectionist models (Field-
One particular connectionist model that considers the problem of perceiving letters within the context of words has been suggested by McClelland and Rumelhart (1981).

In the McClelland and Rumelhart model, a set of letter and word nodes are connected together in a pre-specified manner. When a stimulus representing a visual pattern of features is presented to the system, the relevant letter nodes are partially activated. These letter nodes then partially activate word nodes in the system that in turn reinforce the activation of the appropriate letter nodes. Because the model uses bottom-up and top-down processing simultaneously to arrive at a consistent interpretation of the incoming information, the model is often referred to as the Interactive Activation (IA) model.

The IA model has been successfully applied to a number of problems in the “letter-within-word” perception literature (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982). A unique aspect of the IA model is that all processing within the system is based on very simple local computations similar in spirit to the types of computations that might be performed by neurons. These simple local computations then give rise to interesting global phenomena at the network level. Although the IA model successfully explains a number of experimental findings, McClelland and Rumelhart (1981) do not specifically address the developmental issues. In particular, what are the origins of the letter and word nodes in the IA model? How did the connections between the nodes within the system originate? Although some partial solutions to these questions have been recently suggested (McClelland, 1985; Rumelhart & Zipser, 1985), the problem of how experience with letters and words might specifically influence the development of the ability to detect orthographic information has not been considered.

In this paper, an alternative model of visual letter in word perception, based upon dynamic principles similar to the IA model but different representational assumptions, is suggested. More importantly, however, this alternative model provides a formal framework for considering how the effects of experience influence the visual perception of letters in the context of words. This new model is called the Letter-in-Word (LW) neural network model. The LW model consists of a collection of position-specific letter feature nodes and a learning rule that describes how correlations can develop between pairs of letter feature nodes. Position-specific letters are represented as activation patterns over the position-specific letter feature nodes. Words are represented as conjunctions of position-specific letter activation patterns. A correlation among two letter feature nodes associated with a given letter position in a word is called a within-letter feature correlation. A correlation among two letter feature nodes located in different letter positions in a word is called a between-letter feature correlation. An activation pattern over the letter feature nodes is “learned” by the system through the simultaneous perturbation of both between-letter and within-letter feature
correlations. An error-correction property associated with the learning algorithm is also assumed. Thus, atypical letter feature correlations are learned faster than typical letter feature correlations (Figure 1).

Sources of Orthographic Information in Visual Stimuli
Experimental evidence obtained through studies of how people visually perceive letters in the context of words has been steadily accumulating over the past decade. Such studies have often focussed upon the various types of visual information in a word stimulus that might be used in perceptual processing. In particular, three distinct types of orthographic information in the visual stimulus have been described by a number of researchers.

**THE LW MODEL**

*Figure 1. The LW neural model. Note that the model is specified by groups of letter feature detector neurons and their interconnections. The neurons in each group are dedicated to detecting letter feature information only at a specific letter position in a word. Interconnections within a group (dark lines) are referred to as within-letter feature correlations. Interconnections between groups (light lines) are referred to as between-letter feature correlations. For clarity, only a small subset of the neurons and their interconnections are shown.*
Spatial Redundancy Information. Spatial redundancy information is one particular type of orthographic information. The spatial redundancy of a letter is simply the frequency of occurrence of that letter at a given letter position within a word. The spatial redundancy of a word is calculated by simply summing the spatial redundancy scores of all letters in a word. Mason (1975) (also see Massaro, Venezky, & Taylor, 1979; McClelland & Johnston, 1977) have found that letters presented in the context of letter strings with low spatial redundancy ratings are perceived less efficiently than letters presented in the context of letter strings with high spatial redundancy ratings. Moreover, because spatial redundancy information is an extremely good predictor of performance in such letter perception tasks, this type of information may be quite important for the visual letter-within-word perception process.

Sequential Redundancy Information. Sequential redundancy information is defined as the likelihood of the occurrence of a letter at a specific letter position within a word, given information about the occurrence and location of other letters in the word. The sequential redundancy hypothesis is appealing because letters within orthographically regular strings of unrelated letters (pseudowords) are perceived more efficiently than letters within orthographically irregular strings of unrelated letters (nonwords) (see for example, Baron & Thurston, 1973; Reicher, 1969). Moreover these effects have been observed to be independent of the effects of spatial redundancy information (Massaro et al., 1979). On the other hand, both McClelland and Johnston (1977) and Johnston (1978) in a series of careful studies were unable to obtain any direct evidence supporting the presence of sequential redundancy information in letter-within-word perception. These studies therefore suggest that sequential redundancy information plays only a minor role in the visual word perception process.

Transgraphemic Information. The third type of information in a visual letter string stimulus is sometimes referred to as transgraphemic information. A transgraphemic feature is defined as a constellation of simple contour features that specifies a useful property of visual patterns representing letter strings. Typically, experiments involving mixed-case stimuli tend to support the hypothesis that transgraphemic information is an important aspect of visual letter-within-word perception. In particular, letters within same-case words (e.g., THAT) are perceived more efficiently than letters within mixed-case words (e.g., ThAt) (Adams, 1979; McClelland, 1976; Taylor, Miller, & Juola, 1977). Note that in this case both the spatial and sequential redundancy information in the stimulus have been held constant, but the actual pattern of visual information has been disrupted.
Development of Orthographic Information in the LW Model

The LW Model Learning Rule. Learning in the LW model is based upon the following two-step procedure. First, when a training stimulus is presented to the model, the model does a partial categorization of the training stimulus using the current set of letter feature correlations. An error-correction learning rule is then used to perturb the set of stored correlations using feature correlations in the training stimulus. The magnitude of the perturbation is assumed to be proportional to the difference between the model's interpretation of the training stimulus and the actual value of the training stimulus.

The LW Model Training Procedure. Evidence from the educational literature (for a review see Durr & Pikulski, 1981) indicates that, before children are taught to visually perceive letters in the context of words, the ability to visually discriminate individual letters is at least partially attained. Similarly, learning in the LW model begins by first training the system with a set of single letters, and then training the system with a set of word stimuli. Because letters possess only within-letter feature correlations, the model acquires knowledge of only within-letter feature correlations during the initial stages of learning. Then later, during the word learning stage, a set of between-letter feature correlations is also acquired by the system since words possess both within-letter and between-letter feature correlations.

Developmental Consequences of the Learning Rule and Training Procedure. If such a learning procedure is followed, then the magnitude of the within-letter feature correlations will always be greater than the magnitude of the between-letter feature correlations for the following reason. During the letter learning stage, the magnitude of error the model makes in categorizing letter strings decreases as the time period of letter learning is increased. The smaller error rate in conjunction with the error-correction learning rule then causes the feature correlations to be updated more slowly. Since letters are taught to the model before words are presented, the system acquires a set of large within-letter feature correlations during letter learning and a set of relatively small between-letter feature correlations during word learning. Also note that the ratio of the magnitude of the within-letter feature correlations to the between-letter feature correlations can be indirectly adjusted by varying the ratio of the letter learning time period to the word learning time period.

Correspondence Between Feature Correlations and Orthographic Information. In the LW model, the between-letter feature correlations correspond to the transgraphemic information in visually displayed letter strings. More-
over, certain subsets of the between-letter feature correlations may be considered to correspond to the sequential redundancy information in the matrix. Finally, since the within-letter feature correlations are sensitive to specific orderings of letters and are ambivalent to the transgraphemnic information in a word stimulus, the within-letter feature correlations are defined as the spatial redundancy information in the matrix. Thus, the learning rule and the training procedure implicitly produce a system that is highly dependent upon spatial redundancy information, moderately dependent upon transgraphemnic feature information, and weakly dependent upon sequential redundancy information. This “hierarchy” of dependencies is an important psychological modelling assumption of the LW model since it predicts the presence of effects of spatial redundancy and transgraphemnic information, and the absence of effects of sequential redundancy information.

**Neural Modelling Assumptions**

A particular characteristic of processing in visual cortex is the presence of retinotopic mappings (Cowey, 1981). That is, evidence that the original spatial distribution of information in the visual world is preserved to some extent in the representation of that information by neural activation patterns. Such observations tend to support the hypothesis that a visual word stimulus may be represented as a pattern of neural activity over a set of position-specific letter feature neurons: the ith neuron in the neuronal set responding to a particular letter feature at a particular letter position in a word stimulus. Note that such a representation assumes prior processing that solves problems such as word position and word size invariance. Rumelhart and McClelland (1982) have noted that such problems also arise with their IA model and have suggested a “smearing” transformation as a possible solution. Such a solution may also be applicable to the LW neural model.

The information processing and learning assumptions in the LW model also have a neurophysiological basis. The position-specific letter feature neurons process information by computing weighted sums of the firing rates of the other neurons in the system. Moreover, the weights a given neuron assigns to incoming information evolve according to a Hebbian-like learning rule. Such a rule states that when two neurons are simultaneously active, a change occurs in the nervous system such that the two neurons become more correlated in their discharges. The validity of these modelling assumptions are considered in greater detail by Anderson et al. (1977), Kohonen (1984), and Anderson and Silverstein (1978).

**Integrating the Model with Other Subsystems**

How might this model be integrated with other subsystems that are necessary for word recognition? Kawamoto (1985) has proposed a system consisting of four groups of neurons. The first group of neurons in Kawamoto’s model process the semantic properties of a word stimulus, the second group of neurons process the syntactic properties, the third group process the
phonetic properties, and the final group process the graphemic properties of a word stimulus. In a manner similar to the model proposed here, knowledge in Kawamoto's system is represented through the use of correlational synapses between and within these four subsystems. The LW model may be considered to be the graphemic subsystem of Kawamoto's model.

**COMPUTER SIMULATION METHODOLOGY**

**Specific Modelling Assumptions**

*Representational Assumptions.* To implement the LW model, Anderson et al.'s (1977) Brain-State-in-a-Box (BSB) model was used. In the BSB model a pattern of neural activation is represented by a state vector: the ith element of the state vector corresponding to the difference between the firing rate of the ith neuron in the system and the ith neuron's spontaneous firing rate (Anderson et al., 1977). Using the BSB state vector formalism, a word vector in the LW model may be represented as the concatenation of four letter subvectors. Each letter subvector, in turn, can be represented as a concatenation of letter feature subvectors. More specifically, since the dimensionality of a word vector was equal to 112, a unique 28-dimensional letter subvector was assigned to each of the upper case and lower case forms of the nine most frequent letters of the English alphabet. Each of these letter subvectors was constructed in a principled manner from an extension of an abstract letter feature set originally proposed by Gibson (1969, p. 88). Although the psychological validity of this letter feature set is uncertain, the Gibson feature set was useful for the purposes of simulation. Figure 2 illustrates the state vector encoding procedure. Appendices 1 and 2 provide additional details of the letter and word stimulus sets used in the computer simulations.

*Information Processing Using the BSB Model.* In addition to representing incoming information as a state vector, the BSB model represents knowledge with a matrix of real-valued synaptic weights. Information processing in the BSB model amounts to transforming the incoming state vector in such a manner so as to increase the familiarity between that vector and the matrix of synaptic weights. Neurophysiologically, this transformation may be interpreted as increasing the amount of neural activity in the system using positive feedback until all of the neurons have obtained their maximum or minimum firing rates (Anderson et al., 1977).

More formally, the explicit transformation is given by the following equation:

$$X(K + 1) = S(AX(K) + X(K)) \quad (1)$$

where $X(K)$ is the neural activation pattern at discrete time interval $K$, $A$ is a matrix of synaptic weights, and the $S$ function sets all vector elements whose
Figure 2. An example illustrating the basic concepts used to represent words as state vectors. A word vector is a concatenation of letter subvectors. A letter subvector is a concatenation of letter feature subvectors. In this example, the word vector is 12-dimensional, a letter subvector is 3-dimensional, and the letter feature subvector is a real number. The example illustrates in detail how a letter, located in the first letter position of a word and possessing a horizontal line segment but no vertical or diagonal line segments is, represented in the word vector.

values are above some maximum firing rate deviation equal to that maximum firing rate deviation, and all vector elements whose values are below some minimum firing rate deviation equal to that minimum firing rate deviation.

Equation 1 describes how the state vector evolves at any given time interval in the BSB model. For example, the input state vector $X(0)$ may be inserted into Equation 1 to obtain the transformed state vector $X(1)$. The vector $X(1)$ is then inserted into Equation 1 to obtain $X(2)$. These iterations continue until $X(K) = X(K + 1)$. At this point, usually all of the elements of the system state vector have obtained their maximum or minimum values. When all elements of the state vector are saturated, the vector is referred to
as a hypercube corner. If the input stimulus vector travels to the appropriate hypercube corner, then the stimulus is defined to have been properly categorized. The number of iterations required to arrive at a hypercube corner is taken as the system's reaction time. Finally, note that because the eigenvalues of the matrix tend to be large and positive, the trajectory of the state vector is always directed “outward” towards the corners of the hypercube. (More detailed descriptions of the BSB model may be found in Anderson [1983], Anderson and Mozer [1981], and Anderson et al. [1977].)

The familiarity of a state vector with the set of matrix weights is now precisely defined by the following function. Following Smolensky (1983), this function will be referred to as the *harmony* function. The harmony function for the BSB model is:

\[
E(X) = \sum_{i,j} a_{ij} x_i x_j = X^T A X
\]

(2)

where \(a_{ij}\) is the \(ij\)th element of the \(A\) matrix in Equation 1, and \(x_i\) is the \(i\)th element of the system state vector \(X\).

Assuming the vector \(X\) is a hypercube corner, note that each term of the harmony function measures how closely a given synaptic weight in the matrix matches two specific elements of the corner vector. Thus, larger values of the harmony function suggest a given vector is more familiar to the matrix, while smaller values of the harmony function suggest a given state vector is less familiar. Golden (1986) has formally demonstrated that the BSB model transforms the input vector so as to make the vector more harmonious with the synaptic weights of the matrix where harmonious is given a precise meaning using Equation 2.

**Learning in the LW Model.** The learning rule used in the LW model is very simple. A “scaled-down” version of a training vector is cycled \(K\) times through the BSB model algorithm (Equation 1) to obtain a vector \(Z\). The BSB model's response (the \(Z\) vector) is then compared with the original training vector through a subtraction, and the resulting vector is used to update the matrix weights using a Hebbian (see Anderson et al., 1977) learning rule (Figure 3). The \(i\)th on-diagonal matrix element represents the correlation of the firing rate of the \(i\)th neuron in the system with itself. Thus, to prevent the on-diagonal elements of the matrix from growing without bound, the on-diagonal elements were not updated (see Anderson et al., 1977).

More formally a hypercube corner training vector, \(C_i\), is randomly selected from the set of training stimuli. The vector \(C_i\) is then normalized, perturbed with random noise, and passed through Equation (1) exactly \(K\) times to obtain a new vector \(X_i(K)\). Let \(Z_i = X_i(K)\). The matrix of synaptic weights is then updated using the following learning algorithm:

\[
A_{\text{new}} = A_{\text{old}} + \gamma (C_i - Z_i) (C_i - Z_i)^T
\]

(3)
Figure 3. The learning rule used to update the letter feature correlations. The training vector, $C$, is multiplied by a constant $\beta$ ($0 < \beta < 1$) and noise is added to the vector. The resulting vector, $X(0)$, is cycled $K$ times through the BSB model to obtain the vector $X(K)$. The difference between $X(K)$ and $C$ is then computed and used to update the matrix weights with a Hebbian learning rule.

where $\gamma$ is the learning constant and $[B]$ indicates that the main diagonal elements of the matrix $B$ should all be set equal to zero. The initial value of the $A$ matrix is a matrix of zeroes. Related learning rules that have been successfully used in the BSB model are discussed in detail by Anderson et al. (1977) and Anderson and Mozer (1981).

Consider now the following cost function that directly measures the harmony of the set of training vectors with the set of matrix weights:

$$J(A) = \sum C_i^T A C_i$$

(4)

where $C_i$ is the $i$th hypercube corner vector in the set of training stimuli, $A$ is
the BSB model matrix, and the summation is taken over the set of training vectors.

Empirical studies of $J(A)$ as the learning rule modifies the weights of the $A$ matrix indicate that the learning algorithm is increasing the value of the cost function (Equation 4) for the simulation parameter values reported here. Or in other words, the matrix weights are modified at each learning trial so as to increase the harmony of the matrix with the set of training vectors. This basic idea of constructing an energy landscape in some principled manner, and then using that landscape during the recall process has been suggested by Ackley, Hinton, and Sejnowski (1985). Appendix 3 describes some explicit conditions that indicate when the learning rule (Equation 3) must maximize $J(A)$.

**Computer Simulation Details of the Training Procedure**

Different computer subjects were trained using a set of 196 four-letter words coded as state vectors. Half of the vector set consisted of the upper case forms of the 98 word stimuli (see Appendix 2), the other half of the vector set consisted of the lower case forms of the 98 word stimuli. The individuality of each subject was determined by a unique random number seed that specified the learning sequence order and the random perturbations of the training stimuli.

During the training period, each computer subject learned state vectors whose elements had been corrupted by independent and identically distributed Gaussian noise (see Equations 1 and 3). The noise was added only to the non-zero elements of the position-specific letter training vectors. This noise represented the natural variation of stimuli in the environment plus the internal noise of the nervous system. The standard deviation of the noise/neuron was 0.02 units which was roughly 10% of the signal amplitude/neuron (0.19 units). The maximum deviation of a neuron's firing rate from its spontaneous firing rate was 1.0 units. The minimum deviation was -1.0 units. The value of the training constraint $\gamma$ was always 0.005.

The simulation subjects were initially trained for 1500 learning presentations using state vectors representing randomly selected position-specific letters. Thus, because only vectors representing four-letter words are considered here, three fourths of the elements of each position-specific letter training vector were all set equal to zero. The subjects were then tested using the reaction time and/or letter recognition accuracy testing procedures that will shortly be described. For the next 200 learning presentations, subjects learned stimuli selected at random from the set of upper case and lower case word stimulus vectors and were tested again. The subjects were then trained for an additional 800 learning presentations with the word stimuli, and tested a final time.
The Reaction Time Testing Procedure
In the testing phase, each of 1176 test stimuli was presented in sequence to the BSB model. Each stimulus vector was cycled through Equation 1 until each stimulus was categorized. Only reaction times for stimulus vectors that had been correctly categorized were recorded. A correctly categorized stimulus was defined as a hypercube corner whose four-letter subvectors were all correctly categorized. This particular dependent variable was selected because of its similarity to reaction time measures used in same/different matching tasks for the “same” trials, and letter string search tasks when the target letter is not present.

The Letter Recognition Accuracy Testing Procedure
The reaction time testing procedure examines the behavior of the system under normal operating conditions. To compare the pattern of letter recognition errors produced by the model with the pattern of errors produced by people, a special testing procedure was devised to force the system to make letter recognition errors. The format of the testing procedure was modelled after Reicher’s (1969) visual forced-choice paradigm. In his procedure, a letter string is presented for a brief time period. The letter string is then immediately replaced with a masking stimulus and two letters. One letter appears above one of the letter positions where the original stimulus had been presented, while the other letter appears below that letter position. The subject is then asked to decide which of two letters was presented. The intensity of the display, the stimulus and mask onset times, and various other factors are usually adjusted such that the subject responds correctly about 60% to 90% of the time.

In the specific procedure implemented here, independent and identically distributed Gaussian noise was added to each element of the initial state vector. The signal amplitude per neuron was also reduced. After the first few iterations of the BSB algorithm (Equation 1), a mask vector composed of four Xs was added to the system state vector. The system was not trained upon position-specific X vectors during the letter training stage. The addition of the mask and noise were specifically intended to make the system produce letter recognition errors. The magnitude of the signal vector, the magnitude of the mask vector, the magnitude of the noise vector, and the number of iterations before the mask was applied were selected by the experimenter to avoid floor and ceiling effects in pilot studies. The dependent variable in this procedure was the average percentage of letters correctly categorized at a specific letter position averaging over all four-letter positions.

Finally, in all of the following computer simulation experiments, the treatment effects were always significant (typically, $p < .0001$). Thus, only cell means are reported.
EXPERIMENT 1: ORTHOGRAPHY AND MIXED-CASE EFFECTS (REACTION TIME DATA)

Nine computer subjects were trained using the learning procedure and tested using the reaction time testing procedure. The test stimuli consisted of a set of 1176 same-case and mixed-case stimuli generated from the upper case and lower case forms of the set of 98 words, 98 pseudowords, and 98 nonwords. To generate a mixed-case stimulus, a same-case stimulus was altered by assigning the second and fourth letters of the stimulus to their alternative case values. As described in Appendix 2, an unrelated letter string was classified as a pseudoword or a nonword based upon its spatial redundancy rating. The spatial redundancy ratings were assigned on the basis of an analysis of the word stimuli learned by the model. Table 1 shows the spatial redundancy rating distribution for words, pseudowords, and nonwords.

Simulation Results and Discussion

Juola, Schadler, Chabot, and McCaughey (1978) found that second graders, fourth graders, and college students searched for letters within words faster than nonwords, while no effects of orthographic redundancy were observed in kindergarteners. Moreover, trends in the data indicate that, while for college students, letters within words were perceived faster than letters within pseudowords, the reaction times in the word and pseudoword conditions for the second and fourth graders were essentially identical.

The results of the computer simulations also demonstrate these developmental phenomena (Figure 4). First, as the number of learning presentations increases, the overall reaction time of the system for word, pseudoword, and nonword stimuli decreases. Second, reaction times decrease very rapidly during the first 200 learning trials, and then decrease at a much slower rate as the system obtains additional experience with word stimuli. And third, reaction times for words were 1.551 iterations faster than nonwords at 2500 learning presentations, while at 1700 learning presentations the reaction time difference between words and nonwords was only 1.302 iterations. At 2500 learning presentations, pseudowords were categorized in 0.449 iterations more than words and 1.102 iterations fewer than nonwords. At 1700 learning presentations, pseudowords were categorized in 0.411 iterations fewer than nonwords.

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>62.0</td>
<td>14.0</td>
</tr>
<tr>
<td>Pseudowords</td>
<td>65.7</td>
<td>6.9</td>
</tr>
<tr>
<td>Nonwords</td>
<td>33.4</td>
<td>5.7</td>
</tr>
</tbody>
</table>
Figure 4. Reaction time plotted as a function of case type, orthography, and learning trials for Experiment 1. Solid lines indicate same-case stimuli. Dashed lines indicate mixed-case stimuli.
more than words and 0.891 iterations fewer than nonwords. Thus, as experience with words increased, the word-pseudoword, pseudoword-nonword, and word-nonword reaction time differences also increased in magnitude.

Adult reaction time studies of the effects of orthographic redundancy and alternating case are also compatible with the computer simulation results. For adult subjects, letters in words are perceived faster than letters in nonwords (Mason, 1975; Taylor et al., 1977), and letters in pseudowords are perceived slower than words, yet faster than nonwords (Taylor et al., 1977). In addition, for mixed-case stimuli, words are recognized faster than pseudowords which are recognized faster than nonwords, and the reaction time advantage for same-case over mixed-case stimuli decreases as stimuli become less orthographically regular (Taylor et al., 1977; Pollatsek, Well, & Schindler, 1975).

The reaction times for the same-case and mixed-case stimuli of the computer simulations after 2500 learning presentations are shown in Figure 4. For mixed-case stimuli, words are categorized 0.268 iterations faster than pseudowords which are categorized 1.122 iterations faster than nonwords. In addition, the reaction time difference between mixed-case and same-case stimuli was 0.503 iterations for words, 0.141 iterations for pseudowords, and 0.181 iterations for nonwords. Thus, the simulation results qualitatively agree with the experimental data in all respects with the exception that the alternating case effect for pseudowords was smaller in magnitude than the alternating case for nonwords.

EXPERIMENT 2: ORTHOGRAPHY AND MIXED-CASE EFFECTS (ACCURACY DATA)

In Experiment 2, the nine computer subjects used in Experiment 1 were tested with the same set of 1176 stimuli using the letter recognition accuracy testing procedure. The masking stimulus was added to the state vector at the beginning of the fourth iteration of the BSB algorithm. The signal amplitude per neuron was reduced to 0.95 units, and the noise-per-neuron standard deviation was 0.05 units. The magnitude of the masking stimulus was equal to unity.

Simulation Results and Discussion

The development of the ability to distinguish between pseudowords and nonwords develops very rapidly in children, usually during the first 2 years of grade school (Lefton & Spragins, 1974; Lefton, Spragins, & Byrnes, 1973; Rosinski & Wheeler, 1972). First grade children recognize letters in pseudowords as accurately as they recognize letters in nonwords, while third graders, fifth graders, and adults recognize letters in pseudowords more accurately.
than letters in nonwords (Lefton & Spragins, 1974). Inspection of the types of categorization errors made by the model, as the system's experience with word vectors was increased, reveals a developmental error pattern similar to that obtained in children (Figure 5). At 1500 learning presentations, letters in pseudowords and nonwords were perceived with equal accuracy because the model, at that time period, had not acquired any knowledge about words. At 1700 learning presentations, the magnitude of the pseudoword-nonword effect was 9.94%, while at 2500 learning presentations, the magnitude of the pseudoword-nonword effect was 10.8%.

The model also accounts for the basic patterns of letter recognition errors observed in adult subjects. Adults recognize letters in words more accurately than letters in nonwords (see e.g., Adams, 1979; Johnston, 1978; McClelland, 1976; Reicher, 1969), and perceive letters in pseudowords more accurately than letters in nonwords (see e.g., Adams, 1979; McClelland, 1976). Moreover, for mixed-case stimuli, letters in words are perceived more accurately than letters in pseudowords which are, in turn, perceived more accurately than letters in nonwords (Adams, 1979; McClelland, 1976). In addition, for words and pseudowords, same-case stimuli are more accurately perceived than mixed-case stimuli (Adams, 1979; McClelland, 1976), while the same-case/mixed-case advantage is not apparent for nonwords (McClelland, 1976).

Consider now the performance of the model after 2500 learning presentations. For both same-case and mixed-case stimuli, letters in words were categorized more accurately than letters in pseudowords, which were categorized more accurately than letters in nonwords. Furthermore, the relative magnitudes of the same-case over mixed-case effect for words, pseudowords, and nonwords were 6.08%, 3.85%, and 2.65% respectively. Thus, the letter recognition error patterns characteristic of adult subjects were quite similar to the data obtained from computer simulations of the model with the exception of the same-case over mixed-case advantage observed for the non-word condition.

GENERAL DISCUSSION OF EXPERIMENTS 1 AND 2

In this and following discussions, the effects of the matrix upon the various types of stimulus vectors presented to the model are qualitatively considered. The basis of this qualitative analysis is the observation that the BSB model transforms a testing stimulus such that the resulting state vector is more harmonious with the set of matrix weights. Stimuli that are more harmonious with the set of matrix weights tend to be categorized more quickly and more accurately. The harmony existing between a vector and a matrix is given a rigorous definition in Equation 2.
Figure 5. The proportion of correctly categorized letters plotted as a function of case type, orthography, and learning trials for Experiment 2. Solid lines indicate same-case stimuli. Dashed lines indicate mixed-case stimuli.
An Explanation of Orthographic and Case Effects in the LW Model

By definition, a nonword is a state vector that has not been "learned" by the system. Such a state vector can nevertheless be categorized by the BSB model since the within-letter feature correlations are used to categorize each letter of the nonword independently.

The harmony of a nonword stimulus tends to be less than a word stimulus for two reasons. First, the within-letter feature correlations will be less harmonious with nonwords possessing low spatial redundancy ratings. And second, the between-letter feature correlations will be less harmonious with nonwords possessing low transgraphemic feature ratings. The advantage for letters in words over letters in pseudowords is explained in a similar manner. Also note that, within the framework of this model, the superiority of letter recognition for same-case relative to mixed-case stimuli is analogous to the work-pseudoword advantage.

An Explanation of Developmental Effects in the LW Model

The development of the ability of the model to more efficiently extract orthographic information from a word stimulus as the system gains experience with the set of word training vectors is considered formally in Appendix 3. More informally, the learning rule modifies the matrix weights at each learning trial such that the harmony of the matrix with the set of training stimuli tends to increase. The rapid acquisition of the ability to detect orthographic information occurs because letter feature correlations acquired from only a few words are used to categorize many other words possessing those letter feature correlations.

EXPERIMENT 3: EFFECTS OF VISUAL LETTER FRAGMENT MASKING

The model proposed here differs from several other models in the letter in word perception literature (e.g., Adams, 1979; McClelland & Rumelhart, 1981) in that the letter feature, letter, and word levels of the system are highly interactive. This distinction suggests that the proposed model might be able to offer an explanation regarding why the word superiority effect (Baron & Thurstone, 1973; Estes, 1975; Reicher, 1969; Wheeler, 1970) is so dependent upon the nature of the masking stimulus. Johnston and McClelland (1973) found that when a letter fragment masking stimulus was used, subjects perceived letters in words more accurately than letters alone. However, when a clear mask stimulus was used, the word superiority effect was considerably diminished. It is also important to note that, in both the clear and letter fragment mask conditions, the overall performance of subjects was kept fairly constant. This observation suggests that in the letter fragment mask condition, a high quality stimulus is presented for a short period of
time and is then disrupted by a masking stimulus. On the other hand, in the clear mask condition, a degraded stimulus is presented for a much longer time period (see McClelland & Rumelhart, 1981, for a more detailed discussion).

To explore this phenomenon, a set of five computer subjects were trained with letter and word stimuli according to the training procedure previously described. The subjects were tested with the letter recognition accuracy testing procedure after 2500 learning presentations. The test stimuli consisted of the upper case word stimuli used in Experiment 1. Each word stimulus was then used to generate four position-specific letter stimuli: the letter in the kth position of the word stimulus generating a position-specific letter vector with non-zero elements only in the region of the kth subvector. For example, the word stimulus THAT generated letter stimuli TOOO, OHOO, OOAO, and OOOT where a 0 indicates that region of the state vector should be filled with zeros.

A series of pilot studies suggested the following signal to noise ratios and mask onset times. For the letter fragment mask condition, the standard deviation of the added noise/neuron was 0.05 units, the signal amplitude/neuron was 0.133 units, and the mask was added to the system state vector at the beginning of the sixth iteration of the BSB model testing equation (Equation 1). The magnitude of the letter fragment mask was 5.29 units. For the clear mask (i.e., no mask) condition the standard deviation of the added noise/neuron was 0.095 units, and the signal amplitude/neuron was 0.0855 units. These parameters were selected such that the overall performance degradation in the clear mask condition was approximately equal to the overall performance degradation in the letter fragment mask condition. Note that, for the clear mask condition, the variance of the additive noise is larger than the signal power.

Simulation Results and Discussion
Table 2 shows the results of the computer simulations. The overall performance of the system was about 65% in both the letter fragment and clear mask conditions. For the letter fragment mask condition, letters in words were categorized 23.2% more accurately than letters presented alone. For the clear mask condition, the magnitude of the word superiority effect was

<table>
<thead>
<tr>
<th></th>
<th>Words</th>
<th>Letters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter fragment mask</td>
<td>0.69</td>
<td>0.56</td>
</tr>
<tr>
<td>Clear mask</td>
<td>0.70</td>
<td>0.60</td>
</tr>
</tbody>
</table>
only 16.7%. Qualitatively, these simulation results are in agreement with the experimental literature (Johnston & McClelland, 1973).

In the LW model, the word superiority effect occurs because both the between-letter and within-letter feature correlations help a letter in a word become categorized, while only within-letter feature correlations can aid the model in categorizing single letters. The qualitative differences between the clear mask and letter fragment mask conditions in this experiment are due to the fact that the performance of the model is very robust with respect to test stimuli perturbed with random noise, yet the model is very sensitive to the effects of a letter fragment mask. Moreover, to obtain equal levels of overall performance, the stimulus must be so degraded in the clear mask condition that the system’s sensitivity to the orthographic information in the stimulus becomes diminished. This loss of sensitivity results in a decrease in the magnitude of the word superiority effect.

The reason why the model proposed here is so robust with respect to random noise, but so sensitive to letter fragment masks can be explained as follows. Let a test stimulus vector be perturbed with independent and identically distributed Gaussian noise, and then compute the harmony of the random vector. If the on-diagonal elements of the matrix defining the harmony function are all equal to zero, then the expected value of the random vector’s harmony turns out to be equal to the harmony of the original deterministic test vector. On the other hand, if a letter fragment mask vector is added to the test stimulus vector, the harmony of the perturbed test vector will be quite different from that of the original test stimulus vector. Or in other words, the trajectory of a test stimulus vector perturbed with random noise is not severely affected as it travels toward a hypercube corner since the noise acts equally upon each component of the vector on the average. A letter fragment mask vector, however, is constructed from precisely those vector components that are most influential in determining the trajectory of the state vector through the hypercube.

**EXPERIMENT 4: A DETAILED EXAMINATION OF MIXED-CASE EFFECTS**

Word superiority effect experiments involving mixed-case and same-case stimuli are important because they directly address the issue of whether transgraphemic features should be included in models of visual word perception. If spatial and sequential redundancy were the only types of information used in word perception, then the recognition of letters in same-case words (e.g., THAT) should be as efficient as the recognition of letters in mixed-case words (e.g., ThAt). A number of experimenters have found a same-case over mixed-case advantage for words and pseudowords and a noticeable absence of this effect for nonwords using both reaction time (Pollatsek et al., 1975; Taylor et al., 1977) and letter recognition accuracy.
studies (McClelland, 1976). A notable exception to this consensus was an experiment involving a wide range of font styles and case types by Adams (1979). Adams found that all stimuli (words, pseudowords, and nonwords) were perceived equally well in the same-case relative to the mixed-case condition. She interpreted her findings to suggest that case alternations affect only the processing of individual letters and not the conjunctions of letters.

Adams's results, however, have an alternative interpretation. The suggestion is made here that the use of stimuli comprised of such a wide variety of font styles effectively destroys much of the transgraphemic information. The word–pseudoword and pseudoword–nonword effects in her experiment were presumably due to effects of spatial redundancy. If this hypothesis is correct, then increasing the number of case alternations should tend to decrease the efficiency of letter in word perception. Exactly this hypothesis was tested in a same–different visual comparison task by Taylor et al. (1977). Taylor et al. found that increasing the number of case alternations for words and pseudowords increased reaction time.

To examine the effects of mixed-case stimuli in more detail than Experiments 1 and 2, an additional set of mixed-case stimuli was constructed possessing only a single case alternation (e.g., ThAT). The five computer subjects from Experiment 3 were then tested after 2500 learning presentations with the reaction time testing procedure. The goal of these additional simulations was to explore the relationship between the effects of orthography and the effects of increasing the number of case alternations in words, pseudowords, and nonwords in greater detail.

**Simulation Results and Discussion**

The results of the computer simulations are displayed in Figure 6. Like Taylor et al.'s (1977) study, a case alternation by orthography interaction was obtained. In addition, the simulation shows a sudden increase in reaction time with the first case alternation for words in the same manner as the experimental data (Taylor et al., 1977). The model diverges from the experimental data, however, in predicting for nonwords an increase in reaction time as the number of case alternations are increased. This problem with the model (a problem also observed in Experiments 1 and 2) might be remedied by experimenting with varying the ratio of letter learning trials to word learning trials. Alternatively, an effect of case type upon nonwords might be present (but not easy to detect) in human subjects.

**EXPERIMENT 5: HIGHER ORDER INFORMATION IN THE LW MODEL**

It is generally believed that the perceptual system should make use of "higher order" information during the processing of a visual stimulus. On the other hand, McClelland and Johnston (1977) were unable to find any evidence for
Figure 6. Reaction time plotted as a function of orthography and case alternations for Experiment 4.
position-specific bigram detectors. In this section, the LW model is demonstrated to be very insensitive to the effects of sequential redundancy information. Following these demonstrations, a more sophisticated proposal for thinking about higher order information in word perception is suggested based upon the eigenstructure of the BSB model matrix.

**Absence of Sequential Redundancy Effects**

*Selective Removal of Between-Letter and Within-Letter Correlations.* The model proposed here is predominantly a position-specific letter detector model whose processing is guided by transgraphemic features. The model does not depend upon sequential or bigram information for processing. To illustrate this contention, all of the between-letter feature correlations in the matrix associated with one of the computer subjects in Experiment 3 were set equal to zero. After “surgery” the computer subject was still able to correctly categorize each of the 98 upper case word stimuli. The average reaction time was about nine iterations. The between-letter feature correlations were then replaced and the within-letter feature correlations were removed. Under these conditions, the computer subject was unable to categorize any of the 98 word stimuli in less than 300 iterations.

*Absence of Effects of Position-Specific Bigram Stimuli.* In another experiment, two sets of unrelated letter strings were constructed based upon statistical regularities computed from the set of words actually learned by the system. The first group of unrelated letter strings (the hi-bigram group) possessed a mean position-specific letter frequency count of 42.9 and a mean position-specific bigram frequency count of 5.89. The second group of unrelated letter strings (the lo-bigram group) had a mean position-specific letter frequency count of 42.0 and a position-specific bigram frequency count of 1.29. For purposes of comparison, the set of word stimuli had a mean letter frequency count of 62.0 and a mean bigram count of 11.4. There were 56 stimuli (both upper case and lower case forms) in both bigram groups. The reaction time testing procedure was used to evaluate the performance of the five subjects tested in Experiment 3 using a within-subjects design as usual. For the hi-bigram group, the mean reaction time was 10.93 iterations. For the lo-bigram group, the mean reaction time was 10.81 iterations. Thus, the lo-bigram group was actually slightly faster than the hi-bigram group in processing the stimuli. Note, for purposes of comparison, that the reaction time difference between the pseudoword and nonword conditions in Experiment 1 was about 1.0 iterations, while the corresponding difference in this experiment was 0.12 iterations in the reverse direction.

*An Explanation of the Absence of Sequential Redundancy Effects.* The absence of sequential redundancy effects in the model are due to three fac-
tors. First, the learning algorithm is a highly nonlinear error correction procedure. Frequently used matrix weights are updated less often than rarely used matrix weights. This tends to remove slight statistical frequency differences in the set of training stimuli. Second, the recall algorithm is a nonlinear categorization procedure that also reduces the magnitude of slight frequency differences. And third, the within-letter feature correlations dominate the between-letter feature correlations to such an extent that manipulations affecting only the between-letter feature correlations tend to be "washed-out" by the BSB model recall algorithm (Equation 1).

Perceptual Invariants in the BSB Model
Despite the lack of sequential redundancy effects, however, higher order structural information is present in the BSB model. To see this, rewrite Equation 2 as follows. Assume the A matrix is symmetric.

\[ E(X) = \sum a_{ij} x_i x_j = X^T A X = \sum \lambda_k x^T e_k e_k^T X = \sum \lambda_k (e_k^T X)^2 \]

where \( e_k \) is the kth eigenvector of the A matrix and \( \lambda_k \) is the kth eigenvalue.

Equation 5 demonstrates that the harmony function may be alternatively interpreted as directly measuring how compatible a given state vector is with the A matrix's eigenstructure. Anderson and Mozer (1981) (also see Anderson et al., 1977) note that the eigenvectors with the largest eigenvalues of the matrix in the BSB model have many of the desirable properties associated with what Gibson (1969) and other theorists have referred to as perceptual invariants. Like perceptual invariants the eigenvectors are typically unique and independent. Moreover, Equation 5 shows that the BSB model is a system that enhances the components of a stimulus vector associated with the eigenvectors possessing large eigenvalues.

EXPERIMENT 6: WORD BIAS EFFECTS IN LETTER FEATURE PERCEPTION

A critical assumption of the model proposed here, that differentiates this model from many other models in the letter-in-word perception literature, is that the orthographic information and visual information that specify a word stimulus are allowed to freely interact. Massaro (1979) has attempted to directly study this critical assumption. In his experiment, strings of four letters were presented to subjects followed by a masking stimulus. The subjects' task was to identify a particular target letter in the letter string. Moreover, the target letter was an ambiguous stimulus that could be interpreted as either one of two letters. The magnitude of the visual ambiguity was also varied. Massaro (1979) found that the other letters in the letter string could be selected to bias the interpretation of the ambiguous target letter.
To illustrate how the model can account for the effects of orthographic bias upon the perception of letter feature information, a special set of stimulus vectors was constructed. Referring to Appendix 1, note that if four elements of the letter subvectors that represent E and T are all set equal to zero, that the resulting letter subvector is ambiguous. The coding details of the resulting ambiguous letter subvector E/T are provided in Appendix 1. A set of 15 word stimuli that possessed the letter T in the final letter position, and a set of 15 word stimuli that possessed the letter E in the final letter position were then selected from the set of upper case words learned by the model. The final letter of each of these word stimuli was then replaced with the ambiguous letter subvector E/T. For purposes of control, a set of 30 nonword stimuli with E/T in the final letter position were also constructed.

All 60 stimuli were then presented to the five computer subjects that were tested in Experiment 3. The letter recognition accuracy testing paradigm was used. The input signal to noise power ratio was identical to that used in Experiment 3 for the letter fragment mask condition. The magnitude of the letter fragment mask, however, was equal to unity. Analysis of the simulation data revealed that the model always managed to properly reconstruct either a T or an E in the final position of the letter string. Thus, only the proportion of times that the simulation reconstructed an E in the final letter position of the stimulus will be reported.

For nonwords (control condition), the model interpreted E/T as E 13% of the time. For word stimuli that biased the model to reconstruct a T in the final letter position, the letter subvector E/T was interpreted as an E only 6% of the time. For word stimuli that biased the model to reconstruct an E in the final letter position, the letter E was reconstructed 24% of the time. Massaro (1979) observed similar qualitative effects in his experiment.

The behavior of the model may again be readily understood in terms of the within-letter and between-letter feature correlations. The between-letter feature correlations tend to create a bias in the ambiguous region of the letter subvector E/T in the initial stages of processing. The within-letter feature correlations and between-letter feature correlations then use this bias to reconstruct the missing region of the ambiguous letter subvector.

RELATED MODELS OF WORD PERCEPTION

A good review of additional theories of letter-within-word perception may be found in Henderson (1982). Paap, Newsome, McDonald, and Schvaneveldt (1982) also have proposed an interesting formal model of word perception that uses an empirically derived set of correlations to represent the information in a word stimulus. To emphasize more effectively certain unique aspects of the model proposed here, some comparisons of the Letter-in-Word neural model will be made only with Adams’s (1979) and McClelland and Rumelhart’s (1981) theories of letter-in-word perception.
Adams's Model of Letter-Within-Word Perception

A Description of Adams's Model. Adams (1979) has proposed that associations between letter detector units are formed by a Hebbian learning assumption. As words are presented to her model, pairs of letter detectors that are simultaneously active become bonded together. More specifically, Adams (1979) suggested that the association strength between a pair of letter units could be estimated from a table of bigram frequencies. When a word is presented to Adams's model, the component letter units of the word are directly activated and indirectly receive activation from one another via pair-wise letter unit associations. In contrast to this behavior, when a nonword is presented to her model, only the component letter units of the nonword are directly activated since the pair-wise letter unit associations are very weak. Adams's model is interesting because it parsimoniously explains the word-nonword effect and makes a specific claim regarding how the ability to visually perceive words develops with experience. More specifically, she suggests that the associational strength between pairs of letter units is proportional to the frequency of occurrence of that pair of letter units.

A Comparison of Adams's Model to the LW Model at the Psychological Level. The psychological assumptions underlying Adams's model of letter-in-word perception are very similar to the psychological assumptions of the LW model. In both models, learning is assumed to proceed through the extraction of regularities from an environment, and effects of orthography are considered a result of differential association strengths. The two models differ, however, in several important ways.

First, the functional units in Adams's model are position-independent letter units, while the functional units in the LW model are position-specific letter feature units. The position-specific code in the LW model is used to explain effects of spatial redundancy. In Adams's model, effects of spatial redundancy are assumed to arise from inter-letter association strengths. Massaro et al. (1979) in a very careful study, however, demonstrated independent effects of both spatial redundancy and sequential/transgraphemic redundancy thus casting doubt on Adams's interpretation of the origins of spatial redundancy effects. Second, her letter units are case-independent, while the BSB model does not identify the same letter in two different fonts as the same letter. Thus, her model cannot explain why same-case stimuli are perceived more efficiently than mixed-case stimuli as the stimuli are made more orthographically regular. Third, Adams's model assumes that the processing of orthographic information is independent of the visual letter feature information in a word. Thus, if Adams's model is assumed, then the interaction between orthographic and visual information obtained by Massaro (1979) presumably is due to the response bias stage of processing. Fourth, her model is based upon position-independent letter bigram infor-
mation, which is notably absent in visual word perception tasks (McClelland & Johnston, 1977). The LW model essentially uses only transgraphemic and spatial redundancy information. And finally, although Adams offers an explanation regarding how letter feature correlations develop, her model does not make any statements regarding how the system's letter units initially originated.

The Interactive Activation Model of Word Perception

A Description of the IA Model. The IA model consists of a set of feature detection units or nodes, a set of letter nodes, and a set of word nodes. Each letter node for a specific letter position within a word has a bi-directional excitatory connection with all words possessing that letter in that letter position, and a bi-directional inhibitory connection with the remaining word nodes in the system. In addition, every word node inhibits all other word nodes and every letter node inhibits all other letter nodes through an additional set of inhibitory connections. This within-level or lateral inhibition helps the system to arrive at consistent interpretations by preventing more than one word node from becoming too active at any given time. Each node is also associated with a specific number or activation level that varies continuously between a minimum and maximum level according to a sigmoidal (S-shaped) function. This sigmoidal function will sometimes be referred to as the IA model's saturating nonlinearity. At each time interval, each node simultaneously computes a weighted sum of the activation levels of the other nodes in the system and the connection strengths from those nodes. The weighted sum is called the net input to the ith node. The net input, current activity level, resting level, and decay rate of the ith node are then used to update the activity level of the ith node.

When a word is presented to the model, the letter feature nodes become initially activated. The letter feature nodes then activate the letter nodes which can activate word nodes. Partially activated word nodes then excite their component letter nodes at the letter level of the system while inhibiting the remaining letter nodes. These interactions between the letter and word nodes continue until eventually the system's state settles upon a consistent interpretation of the input at the word and/or letter levels of the system.

Implementing the IA Model Using the BSB Model Formalism. The qualitative dynamic behaviors of the IA model and the BSB model are very closely related. To examine these similarities more closely, arrange the letter and word nodes of the IA model in any order within a high dimensional state vector. Also ignore the IA modelling assumption that the activation levels of the active nodes in the system tend to return to their resting levels. The connection strengths between all the letter and word nodes in the IA model may then be arranged using a BSB model synaptic weight matrix. For exam-
ple, let the word THAT be the first element of the IA model’s state vector and let the symbols T1, H2, A3, and T4 be the second, third, fourth, and fifth components of the state vector. (The notation T4 indicates the letter T in the fourth letter position.) Now consider the IA modelling assumption that states when a word node is activated, the component letters of that word node are also activated. This modelling assumption would be implemented by assigning an equal positive weight to the elements of the matrix that represent the correlations between THAT and T1, THAT and H2, THAT and A3, and THAT and T4. In a similar manner, between-level inhibitory and within-level inhibitory connections may be represented in the matrix using negative valued weights. Thus, by selecting the weights of the corresponding BSB model matrix appropriately, the pre-wired hierarchical structure of the IA model may be realized.

The equations governing the dynamic behavior of the IA model, when represented using matrix algebra, are therefore quite similar to the testing dynamic equations of the BSB model. Both systems cycle their respective state vectors through a connectivity matrix, and obtain useful categorization behaviors using a saturating nonlinearity. Only two significant qualitative differences exist with respect to the mechanisms involved in both models. The activation levels of the nodes in the IA model tend to return to their resting levels, and the saturating nonlinearity used in the IA model is considerably more sophisticated than the BSB model’s nonlinearity.

Comparison of the IA Model and the LW Model at the Psychological Level. Although the mechanisms underlying the dynamic behavior of the BSB and IA models are very similar, the models differ in their “psychological interpretation” of how letter and word stimuli are represented over a given set of nodes (or neurons). In the LW model, the entire activation pattern represents the incoming visual stimulus. In the IA model, the incoming visual stimulus only activates the letter nodes within the system state vector. When a word is presented under normal operating conditions in the LW model, a completely specified state vector is presented to the system that is then merely amplified. In the IA model, the initial state vector will only have the letter nodes activated. Consequently, when a word is presented under normal operating conditions in the IA model, only a partially specified state vector is presented to the system and the model must “reconstruct” the word node regions of the state vector. The incoming visual stimulus in the LW model, on the other hand, partially activates all of the nodes in the system.

It should be noted at this point that the hierarchical vector coding scheme employed in the IA model is powerful enough to permit the reconstruction of completely missing letters in words. On the other hand, the use of only pair-wise letter feature correlations to represent word stimuli in the LW model coupled with the weak between-letter feature correlation contribution, tends to make the LW model very poor at this task.
Two specific points must be made regarding this computational limitation of the LW model. First, this problem arises because of the non-hierarchical coding scheme in the LW model and is not an intrinsic problem with the BSB neural network mechanism. Furthermore, it is presently unclear to what extent people, on the basis of visual information alone, can accurately reconstruct an entire letter stimulus. In fact, the conspicuous absence of sequential redundancy effects in the experimental literature (Johnston, 1978; McClelland & Johnston, 1977) suggests that this computational limitation of the LW model in visual letter-in-word perception may also be shared by people.

The second major difference between the LW and the IA models is that feedback from the orthographic information level to the letter feature level is not permitted in the IA model, but is essential to the LW model. Massaro (1979) has attempted to test experimentally this critical assumption. In his experiment, Massaro was unable to find any direct evidence for an interaction between orthography and perceptual discrimination accuracy. Massaro interpreted his results as evidence for discounting models like the LW model that allow feedback from the orthographic information levels to the letter feature levels. There are some problems, however, with this interpretation. The LW model, for example, assumes a priori the existence of an abstract position-specific letter feature coding. If perceptual discrimination accuracy effects occur at a level of processing prior to the construction of this code, then the lack of an interaction between perceptual discrimination accuracy and orthography is not damaging to the validity of the LW model. Moreover, Massaro did find a highly consistent interaction between orthographic and letter feature information when the dependent variable was the subject's interpretation of the stimulus. Massaro interpreted this latter effect as being associated with the response bias stage of processing. A more parsimonious explanation, however, might assume that this latter effect arose from interactions between the letter feature and orthographic levels in the LW model as demonstrated in Experiment 6.

The third difference between the LW and the IA model is that the IA model assumes (like Adams's model) that the individual letter units are case independent. Models that assume a case independent coding cannot explain why a font type by orthography interaction has been observed (McClelland, 1976; Pollatsek et al., 1975; Taylor et al., 1977). Instead, such models are predicated upon the assumption that the case type by orthography interaction in these studies were obtained because the words, pseudowords, and nonwords stimuli were not tested at comparable levels of performance (Adams, 1979).

It is worthwhile at this point to mention a testable prediction of the LW model that is related to the assumption of case independence. In the LW model, alternating case effects are obtained from exactly the same type of mechanism as orthographic effects. Thus, the development of the effects of alternating case should parallel the development of the ability to detect ortho-
graphic information (see Experiments 1 and 2). As experience with words increases, the LW model predicts that the advantage of letters within same-case stimuli, relative to mixed-case stimuli, should increase at a fast rate in the initial stages of development and more slowly in the later stages. A theory suggesting that the effects of alternating case are located at the “level of single letter discriminability” (Adams, 1979, p. 154; also see McClelland and Rumelhart, 1981), would not necessarily be discredited if this developmental effect were not observed.

The final difference between the LW and the IA models, however, is that the LW model is a learning model. The IA model is initialized by providing the system with a set of letter and word nodes and a set of connection weights. The LW model is initialized by providing the system with a set of independent position-specific letter feature nodes, a learning rule, and an environment. Parameter values in the LW model are therefore a function of how the model is educated. For example, in the IA model, the magnitude of the word superiority effect could be indirectly increased by adjusting the word-to-letter level inhibition parameter in the IA model. Unfortunately, however, such a manipulation is impossible in human subjects. In the LW model, however, the magnitude of the word superiority effect can be manipulated indirectly by varying the ratio of the number of letter learning trials to the number of word learning trials. This latter manipulation is possible, at least in principle, from a psychological perspective.

**SUMMARY**

A developmental neural model of visual letter-within-word perception has been proposed. The model was derived from some commonly accepted principles of information processing within the nervous system, yet the evaluation of the model was based primarily upon the system’s ability to explain a variety of psychological phenomena.

In this particular model, a word is represented as a pattern of neural activity over a set of position-specific feature neurons. This pattern of neural activity is then amplified, using positive feedback, until all neurons within the neural network have saturated. Spatial redundancy information plays a critical role in the amplification process. The contribution of transgraphemic information is smaller in magnitude, but still exerts a detectable influence. The model learns a particular neural activation pattern representing a word by modifying a set of pair-wise correlations that indicate how the set of position-specific letter feature neurons are allowed to interact.

The model was then evaluated with respect to the experimental literature. The LW model provides a rationale for the importance of spatial redundancy and transgraphemic feature effects in visual word recognition. The LW model also explains why sequential redundancy effects have not been particularly easy to find, and how the ability to detect orthographic information develops with experience. In addition, the model makes explicit...
accuracy and reaction time predictions about a variety of experimental phenomena.

Both the LW and the IA models offer a unique perspective upon the nature of the processes involved in the visual perception of letters in the context of words. The detailed application of the IA model to a wide range of experimental phenomena seems to suggest that the basic dynamic principles of the IA model may be fundamentally correct. The LW model, using similar dynamic principles, suggests how experience with word stimuli can affect performance in a variety of visual word perception tasks.

APPENDIX 1

VECTOR ENCODINGS OF LETTER STIMULI

The following table indicates the assignment of specific letter subvectors to letters. The assignment of a vector coding to a letter was based upon an extension of Gibson's (1969, p. 88) abstract letter feature set. The absence or presence of a letter feature is indicated by a two-dimensional subvector that may take on either the values \((-1, -1)\) or \((1,1)\), respectively. An additional pair of letter features were introduced at the end of each letter subvector to indicate the relative size of that letter. For example, the first four elements of the letter subvector representing A are given by \((+1, +1, -1, -1)\) indicating the presence of a horizontal but the absence of a vertical line segment. These four elements are concisely represented in hexadecimal notation with the number C by treating negative vector elements as zeros and positive vector elements as ones. The letter subvector encodings using hexadecimal notation are provided below. The symbol * indicates a set of four vector elements that are all equal to zero (see Experiment 6).

- E F003F3F e C0CF030
- T F00333F t F03F0FF
- A CF033CF a 00C3030
- O 00C303F o 00C3000
- N 33000CF n 30300C0
- R 33C30CF r 303F0F0
- I 30003CF i 30003F0
- S 000CC0F s 000CC00
- H F0033CF h 30300CF
- X 0F0330F E/T F003*3F

APPENDIX 2

LETTER STRING STIMULI SELECTION

Word, pseudoword, and nonword letter strings were used as stimulus vectors in the following experiments. The word stimuli were selected based
upon moderate frequency of occurrence in the English language, and were constructed using only the nine most frequent letters in the English alphabet. The word stimuli were then scrambled, and the scrambled letter strings ranked using a spatial redundancy table obtained by analyzing the original set of word stimuli (see Mason, 1975, for additional details).

**Word Stimuli** (ordered row-wise by decreasing frequency):

THAT THIS INTO THAN THEN HERE AREA SEEN RATE SOON NEAR EAST SORT RESI HEAR HAIR SENT NOTE TEST ONES SHOT NONE KISS HEAI THIN ROSE NINE TINE RAIN ARTS SITE SETS NOSE ONTO TREE SEAT HERO FEAR ASIA HANS IRON ANNE FASE HATE RARE EARS OHIO HOST SEES HORN ROOT SONS TONS NOON STAR TORN HITS TIRE NEAT RENT NEST TENT TOES THEE EARN HERS SINS HIRE TIES TORE HATS NEON SHOE ROAR TROT ROSS TEAR SEAS SORE HINT HOOT HOSE IONS THOR TOSS TRIO SANE ANNA ANTS HEIR OATS RENO RIOT STIR TART OATH SITS TEEN

**Pseudoword Stimuli** (ordered row-wise by decreasing spatial redundancy):

TENE TERE TETS TORS SENE TOSE TEOS TEIS SESE SOND SENS SEST TETN RETS TASE TERN SOST THNE TONR TEAN TSEN TOSN SOET SARE TEHE SETN TOEN NETS TISE TESR REAT ROES TOOR OENS AETS HORN SOSN TOER THOS NORT SOTR NEES NOES ROIT NOES RETN HAST TERA SONE RAAE NOOT SONO HERS EONT SOSR DORS NEET TNOE SAET TROE AEES NOET HEOR TATR REAN HAET TNOS AOST HEER SOER REON SIST NARE EORT SOHT TRIE TRIE TRAT RAST HIST HOTO TEAH TEHN SETA TOSH TNET SNAE HTNE RAET HERA HETA TOOH TAHT TAER SROE HASN SESA SERH

**Nonword Stimuli** (ordered row-wise by decreasing spatial redundancy):

IRES AHSN OHNR TSRA OISN OTSS EASR SRTA SNTA OTA3 ESSE ITSE IHTN INOT TROI OAH T SNEO ERET EOSH ORIT ESAE NSET HTRO HREO HTNO HSTO TSHI RAH ESST ERTN STRA ANAN ESAT HTTA EHSR ISST HTNI EHOH RTTO OHTO ANEN ERAN RNEO EHER ONEN ISET HSEO ERON STRI EATH ONNO OTNR ETTN IHER ESTN INEN ARER OTSN ENNO ATIR AISI HISI EHRK EHRK OKRI ONSI IKNO OSIK AKIR UIEN ASEN EIRH ISSN IRTO ENNA OTNO OSNO ARNI ORRA EHRI ENRA ENTA OTER ITNO OSEF RTOI OHOI NROI ITER ONTI ETRA ESTA OIHO ESSA OTAH ASAI

**APPENDIX 3**

AN ANALYSIS OF THE LEARNING RULE

The learning algorithm described in the text is rewritten here for convenience:

\[ A_{\text{new}} = A_{\text{old}} + \gamma ([C_i - Z_i] (C_i - Z_i)^T) \]  

where \( A_{\text{new}} \) is the updated matrix, \( A_{\text{old}} \) is the original matrix, \( \gamma \) is the learning constant, \( Z_i \) is the system state vector after iterating \( K \) times through the
BSB model, and \( C_i \) is both a training vector and hypercube corner. The square brackets indicate that \([B]\) is equal to the \( B \) matrix with the exception that the on-diagonal elements of \([B]\) are all identically equal to zero.

The conditions under which (1) maximizes the following cost function,

\[
J(A) = \sum_j C_j^\top A C_j
\]

where \( C_j \) is the  \( j \)th hypercube corner that is to be taught to the system, are now considered.

**Analysis**

Pre-multiply and post-multiply Equation 1 by the quantity \( C_j \), to obtain:

\[
C_j^\top A_{\text{new}} C_j = C_j^\top A_{\text{old}} C_j + \gamma C_j^\top [(C_i - Z_i) (C_i - Z_i)^\top] C_j
\]

Now both sides of Equation 3 are summed over all members of the training set yielding:

\[
A = J(U) - J(\&) = \sum_j (C_j^\top (C_i - Z_i) (C_i - Z_i)^\top) C_j
\]

If the on-diagonal elements are not removed during learning, then the summation in Equation 4 is over a set of non-negative terms. Thus, in this case, \( J(A) \) must increase in value after each learning trial.

To study the behavior of the learning algorithm when the on-diagonal elements are set to zero, Equation 4 is rewritten as follows:

\[
A = \gamma (C_i - Z_i)^\top [Q] (C_i - Z_i)
\]

Now let the magnitude of \( Z_i \) be equal to \( Z \). Let \( \Phi \) be the angle \( Z_i \) makes with \( C_i \). Let \( e \) be equal to the unit vector that is parallel to \( C_i \). Then we can write:

\[
Z_i = Z(\cos \Phi) e + Z(\sin \Phi) r
\]

where \( r \) is a unit vector orthogonal to \( e \).

Let \( C \) be the magnitude of \( C_i \) and let \( b = Z/C \). Now put Equation 7 into Equation 5 to obtain:

\[
A = \gamma C^\top (1 - b(\cos \Phi))^2 c^\top [Q] c
- 2\gamma C^\top (b(\sin \Phi))(1 - b(\cos \Phi)) r^\top [Q] r
+ \gamma C^\top b(\sin \Phi)^2 r^\top [Q] r
\]
Let $\lambda_{\min}$ be the absolute value of the smallest eigenvalue of $[Q]$. Let $q$ be the magnitude of $[Q]c$. We then have:

$$
\begin{align*}
\Delta &\geq \gamma c^T (1 - b(\cos \Phi))^i c^T [Q]c \\
&\quad - \gamma c^T b q (2 |\sin \Psi| - b |\sin 2 \Phi|) \\
&\quad - \gamma c^T b^2 (\sin \Phi)^i \lambda_{\min}
\end{align*}
$$

(8)

A new variable called $\tau$ is now formed by dividing the positive term in Equation 8 by the sum of the absolute values of the three negative terms.

$$
\tau = \frac{(1 - b(\cos \Phi))^i c^T [Q]c}{b^2 (2 |\sin \Phi| + \lambda_{\min} |\sin \Phi|^i) + 2b |\sin \Phi|}
$$

Whenever $\tau > 1$, $J(A)$ must increase after the matrix is updated. Also note that if $\Phi = 0$, then Equation 1 reduces to a gradient descent algorithm that minimizes $J(A)$. To see this, note that the gradient of Equation 2 with respect to $A$ is simply $\Sigma C_i C_i^T$. This suggests that the learning algorithm may be interpreted as a suboptimal gradient descent algorithm. Also note that the magnitude of a weight change is directly proportional to the square of the distance between $Z_i$ and $C_i$. This latter observation provides some insight into the origins of the error-correction properties of the algorithm.

Now since $c^T [Q]c / q \equiv 1$ and since $q > \lambda_{\min}$, an approximate expression for $\tau$ may be found that is independent of the properties of the stimulus set. Using both of these approximations, the following expression is obtained:

$$
\tau = \frac{(1 - b(\cos \Phi))^i}{b^2 |\sin 2 \Phi| + 2b |\sin \Phi|}
$$

(9)

The function in Equation 9 has been carefully studied. For $|\Phi| > 135$ degrees, $\tau$ is always greater than unity. In addition, for $b < 0.3$, $\tau$ is always greater than unity. For $|\phi| \leq 135$ degrees and $b > 0.3$, the behavior of $\tau$ is more complex since the error-connection properties of the algorithm interact strongly with the gradient descent properties. For example, with $b = 0.4$, the $Z_i$ vector may deviate 25 degrees from $C_i$ and $J(A)$ is guaranteed to increase. On the other hand, for $b = 0.7$, the allowable deviation is less than 3 degrees. For the simulations reported here, $\Phi$ ranges from about 25 degrees in the initial stages of letter learning to about 5 degrees in the final stages of word learning.

REFERENCES


