LETTER Communicated by Carol Whitney

Broken Symmetries in a Location-Invariant Word Recognition Network

Thomas Hannagan
thom.hannagan@gmail.com
Frédéric Dandurand
frederic.dandurand@gmail.com
Jonathan Grainger
Jonathan.Grainger@univ-provence.fr
Laboratoire de Psychologie Cognitive, CNRS, Aix-Marseille University, 13331 Marseille, France

We studied the feedforward network proposed by Dandurand et al. (2010), which maps location-specific letter inputs to location-invariant word outputs, probing the hidden layer to determine the nature of the code. Hidden patterns for words were densely distributed, and *K*-means clustering on single letter patterns produced evidence that the network had formed semi-location-invariant letter representations during training. The possible confound with superseding bigram representations was ruled out, and linear regressions showed that any word pattern was well approximated by a linear combination of its constituent letter patterns. Emulating this code using overlapping holographic representations (Plate, 1995) uncovered a surprisingly acute and useful correspondence with the network, stemming from a broken symmetry in the connection weight matrix and related to the group-invariance theorem (Minsky & Papert, 1969). These results also explain how the network can reproduce relative and transposition priming effects found in humans.

1 Introduction

Reading is one of the most complex skills that children have to master. In languages that use alphabetical orthographies, such as French and English, the heart of the reading process is orthographic processing: our ability to encode the identities and positions of individual letters in the letter strings that form words. Most current models of visual word recognition assume the existence of some form of location-invariant word-centered prelexical orthographic code—that is, a scheme that codes for the identity of individual letters or combinations of letters smaller than the whole word and that defines the position of letters in the word while abstracting away from their location relative to eye fixation. Therefore, much contemporary research on visual word recognition aims to specify the precise nature of this

location-invariant, word-centered code and how it is activated by retinotopic feature information. Theorizing in this area is subject to the computational constraints imposed by the transformation of a retinotopic to a location-invariant code, as well as the constraints imposed by empirical findings reflecting the flexible nature of the location-invariant code (see Grainger, 2008, for a review).

In light of this, some psychological models have postulated that the shift from a location-specific retinotopic orthographic code to a location-invariant orthographic code is achieved by coding for combinations of letters in the correct order for both contiguous and noncontiguous letter sequences. For example, in the models of Grainger and van Heuven (2003) and Whitney (2001), so-called open bigrams code two-letter combinations in a position-independent yet ordered, but not necessarily contiguous, fashion. As a result, the word WITH, for example, is composed of the following open bigrams: WI, WT, WH, IT, IH, and TH. In this study, we examine whether such letter combination representations are discovered by a trained connectionist network.

One prior study has investigated the learning of location-independent orthographic representations in a connectionist model. Shillcock and Monaghan (2001) trained a feedforward network to map location-specific letters into a location-independent representation of the same letters. For example, the network would learn to associate patterns WITH##, #WITH#, and ##WITH (in which # represents blanks) to the common output WITH coded as a given letter identity at each of four possible positions (slot-coding). Shillcock and Monaghan modeled visual hemifields by splitting the input string at its center, sending these split inputs to two independent processing streams. Model splitting accounted for edge effects—the superiority effect of first and last letters of words in reading—in the sense that network error was lower for exterior letters in the split model but not in a nonsplit model.

In an adaptation of Shillcock and Monaghan's modelling strategy, Dandurand, Grainger, and Dufau (2010) (hereafter DGD) asked whether a feedforward network trained to map location-specific letter identities onto location-invariant word representations would exhibit flexible and location-invariant orthographic coding. The DGD network, shown in Figure 1, successfully simulated two benchmark phenomena observed in skilled readers: transposed-letter priming—the finding that primes obtained by transposing two letters in the target give more facilitation than those obtained by replacing two letters (Perea & Lupker, 2003a, 2003b; Schoonbaert & Grainger, 2004)—and relative-position priming—primes obtained by removing or inserting some letters in the target while maintaining relative letter order are more effective than unrelated control primes (Grainger, Granier, Farioli, Van Assche, & Van Heuven, 2006; Van Assche & Grainger, 2006)—suggesting that the network might have learned to code for contiguous and noncontiguous letter combinations.

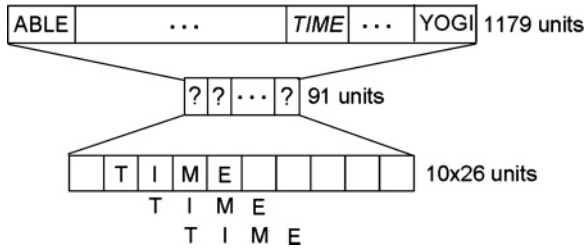

Figure 1: The feedforward network of Dandurand et al. (2010). The network learns to map location-specific letter strings in the input layer to location-invariant word units in the output layer through a hidden layer of 91 units. The training base contains 1179 English four-letter words (one for each unit in the ouput layer). Inputs consist of slot-coded representations presented at one of 10 possible locations.

Research building on Minsky and Papert's group invariance theorem (Minsky & Papert, 1969) has established some sufficient conditions on a feedforward network's connectivity that guarantee location invariance or any other kind of symmetries (Shawe-Taylor, 1993). Learning algorithms such as Tanprop (Simard, Victorri, Le Cun, & Denker, 1992) can enforce these sufficient conditions, and Hinton (1987) showed that backpropagation alone could achieve satisfactory location invariance. However, like all other work on location invariance in backpropagation networks (Shillcock & Monaghan, 2001; Dandurand et al., 2010), Hinton's study aimed at characterizing network performances rather than network knowledge. On the other hand, studies that actually probed for network knowledge did not deal with location-invariant tasks (Shultz & Elman, 1994; Plate, Bert, Grace, & Band, 2000).

Hence, the question of how location invariance is achieved in backpropagation networks has been unanswered until now. Addressing this question in the domain of visual word recognition might contribute to its broader understanding: Does the code for invariant visual words build on letters, bigrams, or any kind of letter combination knowledge? Or did the network settle on an altogether different way to represent letter strings independent from their locations? Does it represent words any differently when letter order particularly matters, such as in the case of anagrams?

After presenting DGD, we report a series of analyses designed to shed some light on these questions. Section 2 deals with the density and overlap of word representations. Sections 3 and 4 use standard clustering and regression analyses to ask whether the network has learned letter or bigram knowledge, respectively. Building on these results, we proceed to emulate the code with holographic reduced representations (Plate, 1995), and test

this emulation against the network. Finally the implications for theories of visual word recognition and invariant pattern recognition are discussed.

1.1 The DGD Network. The DGD network, illustrated in Figure 1, is a standard multilayer feedforward perceptron network with one hidden layer. The input, hidden, and output layers consist of 260, 91, and 1179 units, respectively. The input and output layers use localist coding: each input unit stands for a letter at one of 10 possible locations, whereas each output unit stands for a word in the McClelland and Rumelhart (1981) database. The network was trained using backpropagation with momentum. Presenting a letter string at a given location to the network means switching the corresponding input units to one while the others remain at zero, and word recognition was granted for an input word when activation in the corresponding output unit rised above 0.99. Training proceeded until a sufficiently low target error (SSE) level was reached.

It is known that multilayer perceptron networks such as DGD can approximate any input-output mapping (Hornik, Stinchcombe, & White, 1989). To do so, they use their intermediate layers as a proxy where input patterns get translated in a more adequate format, which is then used to compute the output. Hence, in order to understand the network, a good starting point is to probe its hidden space. We thus begin by analyzing the density and overlap of word representations, two important dimensions on which feedforward networks are known to vary (Plaut & McClelland, 2010).

2 Density and Overlap of Word Representations

In this first analysis, we ask how units activate for any given word representation (the density of the code) and how much activity patterns resemble one another (the overlap of the code).

To calculate densities, all 1179 words in the base were presented centrally (at locations 4–7 e.g., %%%TIME%%%) to the network. The resulting activity patterns were plotted in the histogram of Figure 2 (left), which also shows the density of the anagram subgroup for comparison. Distances were computed for 130 anagrams and 130 nonanagrams (the anagram group featured only one word from each set of anagrams) and are presented in Figure 2 (right). Rather than using cosine as a similarity measure, we used the Euclidean distances between hidden patterns on the grounds that this does not suppress information about vector lengths, which might be of significance to the network.[1] Note that because hidden vectors have dimension 91 and take values between 0 and 1, the maximum possible distance between two patterns is 9.54 (the square root of 91).

[1]The Euclidean distance between vectors x and y is given by $\|x - y\|_2 = [(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2]^{\frac{1}{2}}$.

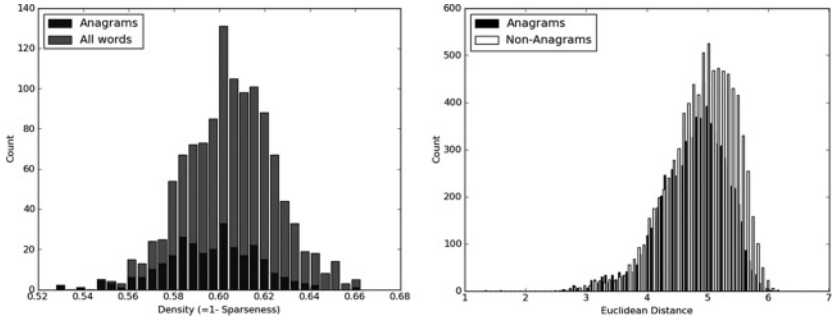

Figure 2: (Left) Density distributions for anagrams (black) and for all words (gray). (Right) Distance distributions for anagrams (black) and nonanagrams (white).

We found that words were densely distributed (mean = 0.604), anagrams slightly less so (mean = 0.596). Nonanagrams had the average overlap expected from independent and identically distributed (i.i.d.) gaussian vectors (mean = 4.86), and anagrams overlapped slightly less (mean = 4.79). Extending the overlap analysis to all words in the DGD lexicon, a population of 694,431 distances (not shown), we found a distance distribution similar to the nonanagram group. This distribution was unimodal, with a mean of 4.88, a low standard deviation of 0.53, and a negative skew (−0.77). Consequently the median was slightly higher than the mean, at 4.95. The bulk of distances (94%) were superior to 4.

2.1 Conclusions from the Preliminary Analysis. These basic statistics show that word patterns are densely distributed and follow a gaussian law centered close to the middle of the activity range. Patterns are not confined in the same region of hidden space, and their average distance deviates only slightly from a gaussian distribution. The density distribution of anagrams is more widespread, and they also overlap less. All in all, this suggests that words can be distinguished between efficiently on the basis of their hidden patterns only, hence minimizing the role played by hidden-to-output connections and supporting our choice to focus on the input-to-hidden set of weights.

Because we must analyze dense and overlapping representations of words with many exemplars, a natural choice is to use cluster analysis. Our underlying assumption will be that if patterns cluster along a given dimension, this means the network has learned something (has some knowledge) about this dimension. Also because it appears that anagrams are treated differently from other words, we will systematically compare anagrams to nonanagrams.

Table 1: *k*-Mean Clustering on Hidden Activation Patterns Evoked by Single Letter Inputs.

Letter	Cluster Size	Majority	Alpha Error	Beta Error	Error	Fitness	Letter Frequency
A	9	100%	0	1	1	.395	9.05%
B	10	100	0	0	0	.476	2.65
C	9	100	0	1	1	.442	2.97
D	10	100	0	0	0	.431	4.13
E	10	100	0	0	0	.395	11.20
F	10	100	0	0	0	.362	2.16
G	10	100	0	0	0	.402	2.16
H	10	100	0	0	0	.564	3.14
I	9	100	0	1	1	.493	5.45
J	7	100	0	3	3	.633	0.38
K	10	100	0	0	0	.466	2.82
L	10	100	0	0	0	.456	7.78
M	10	100	0	0	0	.414	3.12
N	10	100	0	0	0	.410	5.07
O	10	100	0	0	0	.370	7.23
P	10	100	0	0	0	.440	3.77
Q	21	48	11	0	11	.179	0.04
R	10	100	0	0	0	.429	5.81
S	10	100	0	0	0	.362	5.68
T	10	100	0	0	0	.439	5.75
U	8	100	0	2	2	.469	3.63
V	9	100	0	1	1	.387	1.00
W	10	100	0	0	0	.388	2.50
X	9	100	0	1	1	.409	0.32
Y	10	100	0	0	0	.466	1.72
Z	9	100	0	1	1	.456	0.47

Notes: The majority column shows the number of exemplars of the best-represented letter in the cluster. An exemplar that was assigned to cluster Y, when it should have been assigned to cluster X, counts as a beta error for X and an alpha error for Y. The global error score is the sum of alpha and beta errors. The reported letter frequencies are collapsed over all position in-string.

3 Letter Analysis

Having established that word representations are densely distributed with average overlap, we proceed to ask whether in order to represent words, the network uses some knowledge pertaining to single letters. To do so, we exposed the network to single letter inputs at each of the 10 possible locations and collected the resulting hidden-layer activities. We then performed a *k*-mean clustering procedure with 26 clusters on these 260 activation patterns.

Results are presented in Table 1. We found a one-to-one mapping between clusters and letters, in the sense that the *k*-mean algorithm uncovered 26 clusters, each showing a majority of exemplars from the same letter.

Therefore, we will simply describe clusters by giving their sizes as well as the extent of the majority achieved by its dominant letter. We also define a cluster as exact if it has size 10 and a majority of 100%.

Clusters had an average size of 10 items (SD = 2.36) and an average letter majority of 98%. The most ambiguous cluster was cluster Q, with a majority of only 48% but 11 misclassified exemplars (3 J exemplars and 8 exemplars coming from separate clusters, either in first or last position). There were 17 exact clusters, 3 of them being vowels, so that the clustering success for vowels was 50%, and 75% for consonants.

We also computed silhouette fitness scores for all clusters (see the appendix). Fitness and errors were anticorrelated ($R = -0.47$)—exact clusters having better fitness, and conversely. Indeed cluster Q, which had the most errors, showed the worst fitness score ($f(Q) = 0.179$). However, the anticorrelation was not perfect, as exemplified by cluster J ($f(J) = 0.633$), which achieved the best fitness score but the second worst in terms of errors. This situation appears mostly to reflect the high sensitivity of fitness scores to alpha errors. Indeed low-fitness cluster Q produced 11 alpha errors, whereas high-fitness cluster J had none.

Comparing fitness and error scores to letter frequencies in the DGD database (last column in Table 1), we found that letter frequency anticorrelated weakly with errors ($\rho = -0.36$) and not with fitness ($\rho = -0.05$). These correlations did not improve much when letter position frequencies were considered or when one further distinguished between types of errors. We also found that anagrams were related with errors: looking for letters that were never seen in anagrams (j, q, x, and z), or remained at the same position in these anagrams (y), we found that these coincided precisely with the most poorly defined clusters in terms of errors.

The existence of a natural classification, a one-to-one mapping between letters and clusters, is also apparent from simple inspection of the distances between all 260 letter exemplars, shown in Figure 3 (left). As we can see in this letter distance matrix, patterns evoked by a given letter input are similar to one another regardless of the position of this input (white diagonal squares), and not so much similar to any other exemplars (dark gray nondiagonal squares). The light gray lines and columns for letters Q and J outline the proximity of these letters to all others, explaining the classification errors made in their respective clusters.

Figure 3 (right) focuses on the distances between patterns produced by the same letter seen at different locations. These distances are collapsed for all letters, producing a symmetrical 10×10 location distance matrix. This figure reveals two facts about hidden patterns: both proximity and centrality of locations make patterns more similar. Indeed, letter inputs presented at outer positions evoke hidden patterns that are more segregated than those coming from inner input positions. This impact of centrality is visible in the lighter gray regions around central locations. At the same time, distances between patterns increase as the gap between their

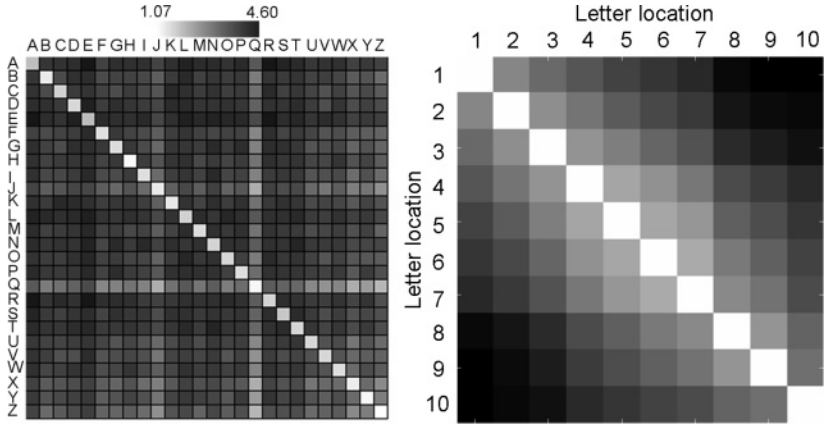

Figure 3: (Left) Distance matrix between letter exemplars across all clusters. (Right) Distances within a letter category, averaged over all categories. Axes show the letter location in the input array. White = small distances. Black = large distances.

corresponding letter locations widens. This is reflected in the light-to-dark gradient as one recedes from the white diagonal. Proximity and centrality factors are thus both visible as gradients of gray going in orthogonal directions (resp. along the first and second diagonals). Moreover, the predominance of proximity over centrality is apparent from the much stronger gradient along the first diagonal.

The centrality effect can be readily attributed to differences in letter exposure during training, as illustrated in Figure 4. Because the input layer is not circularly connected (i.e., it is an array with a first and a last location), during training fewer letter exemplars are seen at the edges than at the center. Letter location frequency is in fact an inverse U-shaped function of location (square-marked curve in Figure 4). It is known that in backpropagation networks, frequent or early seen items come to "catch the weights" (Smith, Cottrell, & Anderson, 2001; Ellis & Lambon-Ralph, 2000), which in our case produces an increase in the average input-to-hidden weights across letters as one moves toward central locations. As a consequence, this inverse U-shaped function is also reflected in the length of letter patterns (the circle-marked curve in Figure 4). When averaged over all letter vectors, inner vectors tend to be longer than outer vectors (the triangle-marked curve in Figure 4).

3.1 Conclusions from Letter Analysis. In summary, this analysis showed that exemplars of the same letter tend to cluster well together regardless of location in the input array. Cluster fitness is correlated with

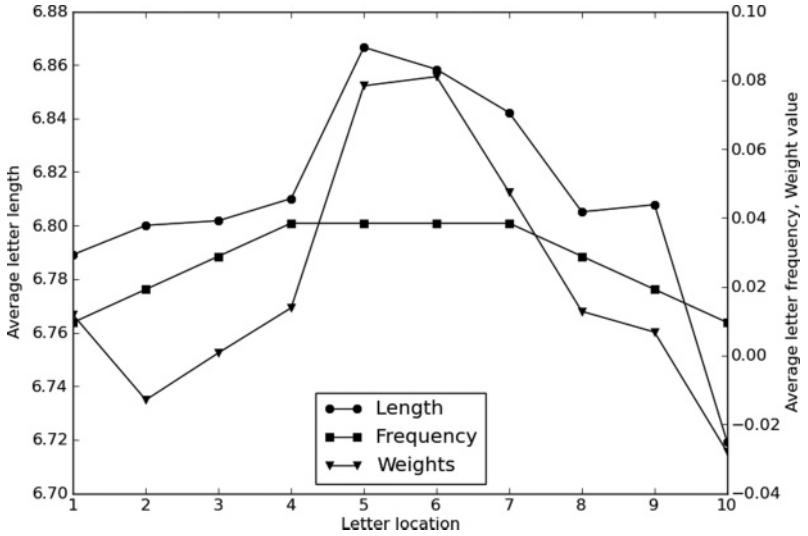

Figure 4: Average lengths of hidden letter vectors (circles), letter location frequencies (squares), and connection weights (triangles) plotted against the location of letter input.

errors, especially beta errors (when the exemplar does not conform to cluster majority). Letters that were never seen in anagrams or remained at the same position in these anagrams are precisely those for which we have cluster errors. Letter clusters have an internal structure: they show both a centrality and a proximity effect. The (relatively weak) centrality effect observed within clusters appears to be related to unequal exposure during training, which has an impact on weight strengths and, consequently, on hidden vector lengths.

These results allow us to conclude that the network has learned about letter identities, although any given letter at two distinct locations can still be distinguished. We will refer to these letter representations as semi-location-invariant. However, it might still be that the network has extracted other regularities from the training base, such as some knowledge about letter combinations. A natural letter combination candidate to be testing is the open bigram (Grainger & Van Heuven, 2003; Whitney & Berndt, 1999), and the next section asks whether some knowledge pertaining to these entities could indeed be present in the network.

4 Bigram Analysis

In this section we conduct three analyses to determine whether the network has encoded information about bigrams: combinations of two

letters that are position independent yet ordered, but not necessarily contiguous.

4.1 Cluster Analysis. We first perform a clustering analysis. If the network has learned bigram knowledge, we would expect the clusters for bigrams that are present in the training base to differ significantly in fitness from those absent from it.

We chose 24 bigrams and divided them along three dimensions: (1) bigrams had either been seen or not during training, (2) they consisted of either repeated or mixed letters, and (3) they either contained consonants only or contained some vowels.

Bigrams in the unseen condition had a frequency of zero in the training corpus, and to limit the risk of confounds, neither had their transpositions been seen (or they had the lowest possible frequency when this could not be helped). The condition with two different vowels was not available because it did not match this criterion. Whenever possible, bigrams in the seen condition were constructed from the unseen condition by replacing one letter so as to minimize structural differences while maximizing frequency differences.[2]

We presented a given bigram with one of three gaps (AB, $A-B$, $A--B$, bigger gaps were proscribed by the four-letter strings in DGD) at each of the possible locations in the input layer (resp. 9, 8, and 7 locations) used in DGD (simulation 2.3.1), and collected the activation patterns evoked in the hidden layer.

We then collapsed all activity patterns into two groups of 12 bigram classes each, making sure that each group had an equal number of vowel and consonants, mixed and repeated, and seen and unseen bigrams. The division in two groups was required to avoid interference across bigrams. In this way, no two bigrams of the same group shared a common letter. Finally, the $(9 + 8 + 7) \times 12 = 288$ hidden patterns corresponding to all bigram exemplars in each group were submitted to a k-means procedure with 12 classes.

4.2 Results. The results of the two k-mean clusterings are reported in Table 2. The first thing to notice is that all clusters were perfectly identified; no errors were made. The fact that bigram clusters were better defined than the letter clusters from the previous analysis is a consequence of the former having half as many clusters as each of the latter. In consequence, Table 2 reports only fitness scores, along with bigram frequencies.

[2] A consequence of maximizing bigram frequency differences between groups is that related bigrams in each conditions had large gaps in letter frequencies. An alternative setup in which letter frequency differences are kept minimal is also possible but reduces the bigram frequency contrast. The setup we described was chosen because analysis 2 showed that letter frequency had little or no impact on cluster fitness scores.

Table 2: K-Mean Cluster Analysis Results for Bigram Groups 1 and 2.

	Unseen			Seen		
	Repeated	Fitness	Frequency	Mixed	Fitness	Frequency
Bigram group 1						
Vowels	UU	0.556	0.0	AK	0.575	40
Vowels	YY	0.616	0.0	ES	0.526	43
Vowels	II	0.589	0.0	OT	0.531	46
No vowels	JJ	0.570	0.0	DR	0.522	16
No vowels	VV	0.511	0.0	HN	0.640	18
No vowels	WW	0.545	0.0	PL	0.585	24
	Seen			Unseen		
	Repeated	Fitness	Frequency	Mixed	Fitness	Frequency
Bigram group 2						
Vowels	OO	0.536	42	UV	0.428	0.0
Vowels	AA	0.498	12	IX	0.576	0.0
Vowels	EE	0.525	51	YQ	0.568	0.0
No vowels	LL	0.555	44	DZ	0.509	0.0
No vowels	SS	0.508	18	HJ	0.585	0.0
No vowels	TT	0.567	13	PF	0.517	0.0

As Table 2 suggests, fitness scores were similar in all conditions. Indeed, an analysis of variance did not reveal any main effect of training at the 0.05 level ($F(1, 12) = 0.0,\ p = 0.99$), nor did it show an effect of presence of vowel ($F(1, 12) = 0.35,\ p = 0.56$), or of repetition ($F(1, 12) = 0.0,\ p = 0.33$). Furthermore, none of the interactions was significant at the 0.05 level. Finally, frequency did not correlate with fitness scores ($R = -0.03$), and the correlation did not much improve when it was restricted to seen bigrams ($R = -0.11$)

Within bigram clusters, patterns had the same organization in terms of proximity previously found for letters: exemplars presented at close locations produced similar patterns. There was no sign of any influence of gap between constituent letters, which is compatible with the noncontiguous characteristic of bigrams.

In sum, this analysis failed to find any clear and direct evidence of bigram knowledge. Although bigrams clustered together just as letters did and showed a similar internal organization, there was no difference between seen and unseen bigrams in terms of fitness scores. We can think of two conflicting explanations for this result. One possibility is that the network could have extracted bigram information from the training environment and might generalize from this to unseen bigrams.[3] Alternatively, the network might not have any bigram knowledge, and the observed

[3]We thank one anonymous reviewer for pointing this out to us.

bigram clusters would be simple combinations of letter clusters that reflect knowledge about individual letters. In order to try and decide between these two alternatives, we proceed to a second analysis.

4.3 Bigram Index. Many different properties are associated with the notion of a bigram: it must activate for ordered combinations of letter pairs that can possibly be noncontiguous and may occur at different locations.[4] Although these properties may vary across schemes, all agree in asserting that a given bigram entity should be more active when the input letters are shifted by one location than when they are reversed altogether.[5] When representations are densely distributed, the prediction is that a bigram pattern should be more distant to the pattern elicited by its transposed version than by its shifted version. In fact, because the resulting distances are graded measures and because this property is the only one that is genuinely common to all bigram schemes, it provides a good basis to quantify the extent to which the network has learned bigram representations.

To any bigram, we can associate two quantities S and T, which may be interpreted as the sensitivity to order and location, respectively. We define S in the following way:

$$S_{ab} = \frac{d(a_5 b_6, a_6 b_7)}{d(a_5 b_6, x_6 y_7)}.$$

Hence S is the distance between the hidden patterns obtained when bigram AB is presented centrally (positions 5 and 6) and when it is shifted by one location (positions 6 and 7), divided by the distance to a different shifted bigram XY.

Likewise, T is defined as

$$T_{ab} = \frac{d(a_5 b_6, a_6 b_5)}{d(a_5 b_6, x_5 y_6)}.$$

T is the distance between hidden patterns for centrally presented and transposed (positions 6 and 5) bigrams AB, divided by the distance to a different bigram pattern XY. S and T indices were computed based on the Euclidean distance and cosine similarities[6] for 50 discriminant bigrams ($D > 0$) and 50 nondiscriminant bigrams ($D = 0$) matched for frequency.[7]

[4]With the exception of overlapping open bigrams (Grainger et al., 2006), where unordered letter pairs are activated to a lesser degree.

[5]We thank Carol Whitney for suggesting this to us as a test.

[6]The cosine distance between X and Y was obtained by $1 - \text{cosine}(X, Y)$, which was always positive for the exemplars considered.

[7]We used stratified samples in five frequency ranges of 10 bigrams each.

When $S = T = 1$, the network sees disordered or displaced input bigrams as different entities altogether: all have orthogonal codes. On the contrary if $S = T = 0$, the network sees disordered or displaced input bigrams as the same entity. Bigrams' defining characteristic can thus be written as $0 \leq S < T \leq 1$. When $S = 0$ and $T = 1$ the network recognizes shifted inputs as the same entity, but disordered bigrams as completely different entities. This is the distributed analog to the localist bigrams used, for example, in Grainger and Van Heuven (2003) and Whitney (2008), and means that the network has learned location-invariant bigram knowledge. In contrast, when $S = 1$ and $T = 0$, the network has learned to recognize unordered letter pairs seen at a given location and thus encodes no information about relative position or order of letters. Such representations violate the defining characteristic of bigrams.

Let us finally define discriminant bigrams as those that are useful to distinguish between anagrams. Given a set of anagrams (e.g., ["vein," "vine"]), a discriminant bigram ("ne") is one that is present in at least one anagram from the set (here in "vine") but not in all of them (not in "vein").

4.4 Results. Figure 5 shows that shifting and transposing a bigram was equally disruptive, as apparent from the fact that all bigrams sit more or less on the diagonal. This absence of a difference holds whether bigrams are helpful or not for discriminating anagrams (resp., white and black signs), for both Euclidean (marked using circles, $p = 0.31$ for $D > 0$ and $p = 0.44$ for $D = 0$) and cosine distances (marked using squares, resp., $p = 0.28$ for $D > 0$ and $p = 0.43$ for $D = 0$). The close-to-zero cosine measures (mean = 0.07) indicate that shifted, transposed, and reference bigram vectors are all very much aligned in the same direction in hidden space. The low Euclidean measures (mean = 0.27) indicate that the network assigns largely overlapping codes to inputs that are made of the same letter. Taken together, these results establish that the network has not learned any bigram knowledge.

4.5 Bigram Patterns Regressed on Constituent Letter Patterns. The large overlap between hidden patterns obtained for different bigram transforms suggests that the code for a bigram relies on the codes of its constituent letters. Although the sigmoid activation function in the network ought to imply that this relationship is nonlinear, we should first test for the simplest relationship using linear regressions.

Hidden patterns were produced for 50 discriminant bigrams and 50 nondiscriminant bigrams presented at randomly selected locations and for their constituent letters presented alone at the same locations. Multiple linear regressions were carried out with the bigram pattern as the dependent variable and letter patterns as explanatory variables.

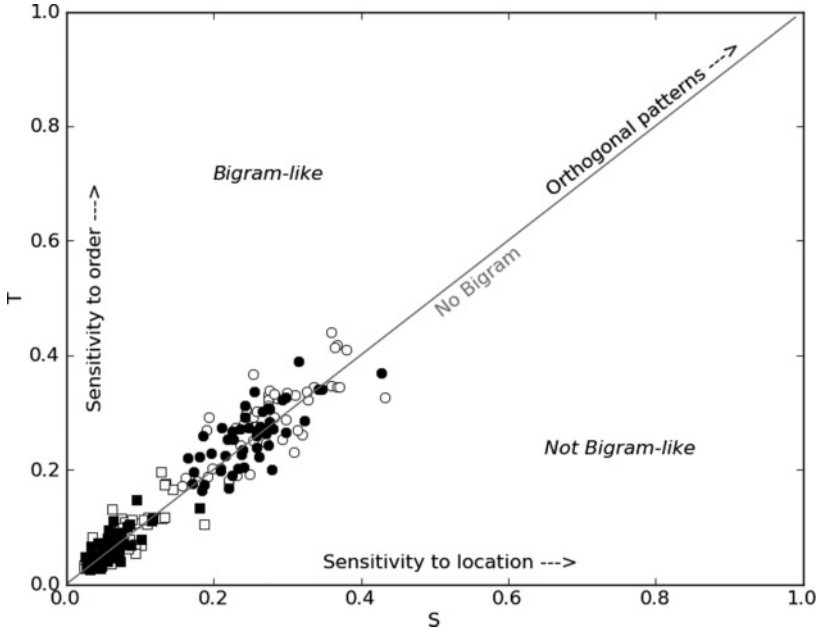

Figure 5: Assessment of the bigram knowledge learned by the network. Discriminant (white) and nondiscriminant (black) bigram representations plotted by their Euclidean (circles) and cosine (squares) coordinates in the $S \times T$ plane (S: horizontal, T: vertical).

Distributions of R^2 regression values for discriminant and nondiscriminant bigrams are shown in Figure 6. All bigrams had high R^2 values, with nondiscriminant bigrams being slightly but significantly higher (mean $= 0.85$ against mean $= 0.83$, $p < 0.01$). These results suggest that a lot of variance in bigram patterns can be explained by letter patterns in a combination that appears to deviate only slightly from linearity.

4.6 Conclusions from Bigram Analysis. In this section, we have conducted three different analyses to assess whether the network has learned bigram knowledge. First, we showed how hidden patterns for seen and unseen bigrams clustered in the same way. Second, we established that the network does not exhibit the defining characteristic of bigram representations, because shifting and transposing a bigram was equally disruptive, even for discriminant bigrams. Finally, we demonstrated that bigram patterns can be very well approximated by a linear combination of their constituent letter patterns. In light of these analyses, our conclusion is that the network does not use a bigram code in order to recognize visual words, but rather that the code is based on letter knowledge. This leaves open the question of how

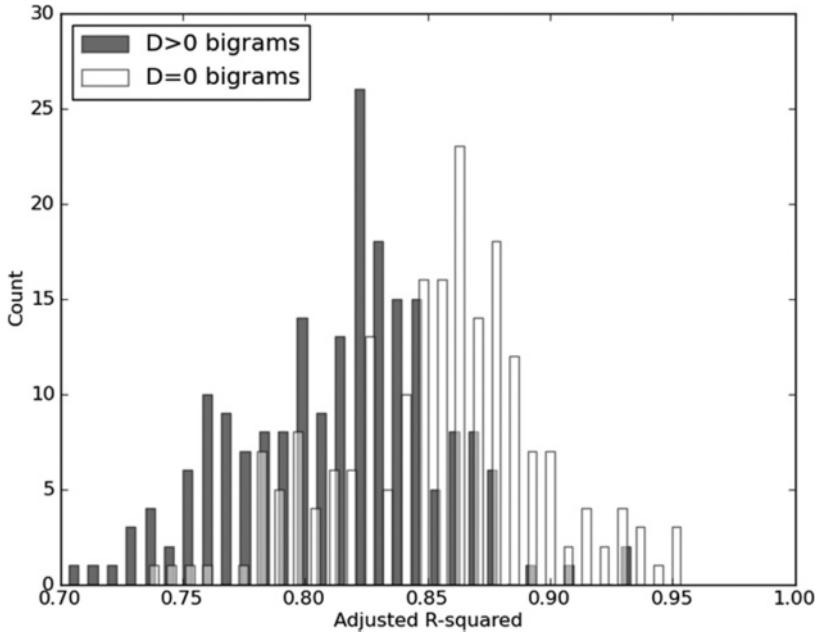

Figure 6: R^2 distributions of linear regressions for bigram vectors over letter constituent vectors (black = discriminant bigrams, white = nondiscriminant bigrams).

information about relative letter position is factored in, which is required for distinguishing among anagrams. We address this question in the next section.

5 Building Up the Code: A Letter-Based and Linear Account

The previous bigram analysis suggested the possibility that any string pattern can be approximated by a linear combination of letter-constituent activity patterns. We put this idea to the test by performing linear regressions of word patterns.

5.1 Word Patterns Regressed on Constituent Letter Patterns. Hidden patterns were produced for 283 anagrams and 283 nonanagrams presented at randomly selected locations and for their constituent letters presented alone at the same locations. Multiple linear regressions were carried out with the word pattern as the dependent variable and letter patterns as explanatory variables. For each word, we performed a control regression on nonconstituent letters at the same location.

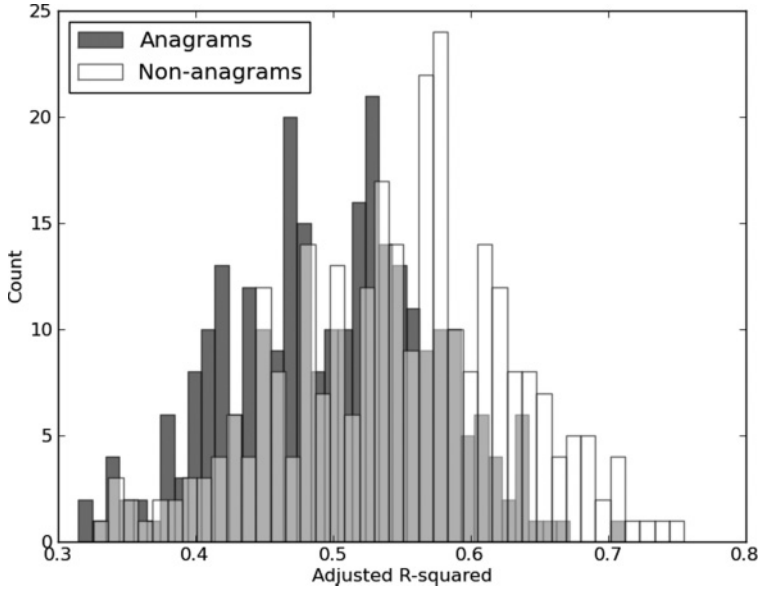

Figure 7: Linear regressions of word vectors over letter constituent vectors (black = anagrams, white = nonanagrams). R^2 distributions.

Figure 7 shows that regressions of word vectors on their constituent letter vectors again prove efficient (mean $R^2 = 0.53$), nonanagrams being better explained than anagrams (0.55 against 0.50, $p < 0.01$). These scores cannot be attributed to a general overlap between hidden vectors, since control regressions on nonconstituent letters at the same location yielded negligible scores (mean $R^2 = 0.05$).

Our results show that although it is less accurate, the linear type of relationship found for bigrams also holds for words, anagrams or not.[8] Taking profit of location uncertainty for any letter, such a code would indeed be able to produce different representations for any words, including anagrams, while keeping exemplars of the same word similar. The slightly (but significantly) inferior scores for anagrams, also found in bigram regressions, show that for anagrams, the mapping deviates a bit more from linearity. This is presumably used by the network to make hidden patterns for anagrams more distinguishable from one another than exemplars of the same word, allowing the second set of weights (input-to-output weights) to support accurate identification.

[8]More generally, unreported simulations suggested that R^2 scores decrease with word length.

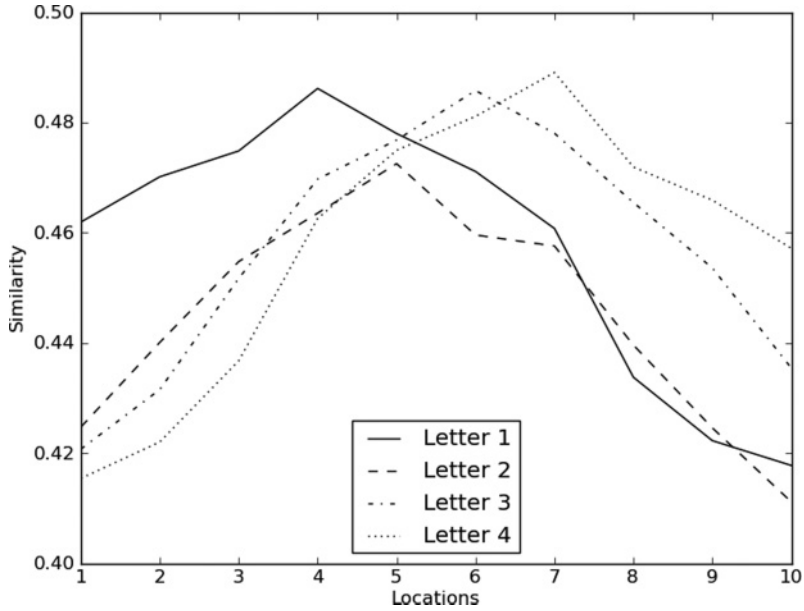

Figure 8: Average DGD similarities between a word and its letters across locations.

Critical in the coding scheme described here is the overlap between exemplars of the same letter, which implies an overlap between exemplars of any arbitrary string, making location-invariant word recognition not only possible but also flexible. In fact, the proximity effect in letter clusters apparent in Figure 3 (right)—the pattern of distances between codes for the same letter at different locations—appears closely related to the overlap model recently proposed by Gomez, Ratcliff, and Perea (2008).

5.2 Relationship Between DGD and the Overlap Model. In the overlap model, letter position is not a fixed value but rather a distribution of probabilities whose maximum coincides with the correct within-string position (e.g., in the word TIME, letter I has its position probability peak at position 2), the model allows for uncertainty about the true position of the letter, with decreasing, yet nonnull, probability of letters being displaced (e.g., TMIE). According to the model, this uncertainty explains the flexible coding observed in skilled readers.

Figure 8 shows how the similarity of hidden pattern activations in the DGD network between some word and its constituent letters varies as a function of location in the input array. This was computed as the average Euclidean distance on normalized vectors of all words in the base to their constituent letters at all possible locations.

Figure 8 gives an idea of how much one can infer from a word pattern about the location of its constituent letters. The curves present themselves neither as single spikes that would be situated at the correct location (no overlap, perfect information about location) nor constant lines across locations (perfect overlap, no information about location). Instead, for every letter curve, the similarity increases monotonically to culminate at the correct location (e.g., the curve for letter 1 peaks at location 4) and then to decline again monotonically. Hence, although each letter location can be deduced accurately from a word code, locations are more easily confused when they are close to one another.

This graph lends support to the idea that location uncertainty in the DGD network serves precisely the same role as position uncertainty in the overlap model: giving flexibility to the code. The uncertainty on letter locations in any word pattern is indeed the exact counterpart of the fact that exemplars of the same word receive similar hidden patterns and will cluster together.

But despite this close relation, the overlap model cannot explain density distributions and clustering patterns in DGD. Next we show that by using holographic reduced representations (Plate, 1995), we can build a distributed analog of the overlap model that meets these goals.

5.3 Holographic Overlap Coding. Holographic string encoding is a general method to implement various orthographic schemes, for instance, the overlap model or open bigrams, in a distributed way (Hannagan, Dupoux, & Christophe, in press). To create an overlap representation for any arbitrary string, each letter vector is bound to the adequate location vector (e.g., letter vector T in word %%%TIME%%% is bound to location vector 4), and the bindings are simply summed together. Letter and location vectors are created at random by drawing 91 components from a centered gaussian distribution of variance $\frac{1}{\sqrt{91}}$. Twenty-six independent letter vectors were generated in this way, but the 10 location vectors had correlated components to make them overlap: except for the first, each location vector l_i was obtained by copying $\rho = 90\%$ randomly chosen components from the previous vector l_{i-1}, and regenerating the remaining 10%. Binding is achieved using the circular convolution operator described below. These letter and location bindings are then summed together, along with a common independent vector ψ, resulting in the final holographic reduced representation for the word.[9] For example, for the word TIME presented in central location, we have:

$$\%\%\%TIME\%\%\% = T \otimes l_4 + I \otimes l_5 + M \otimes l_6 + E \otimes l_7 + 2\psi.$$

[9] Although in Plate (1995), chunked vectors were normalized in order to keep them in the same format and allow further compositions, a simple sum is sufficient for our purposes.

Where l_i are the correlated location vectors, ψ is a vector common to all words, and $\otimes$ is the circular convolution operator defined by

$$(X \otimes Y)_i = \sum_{k=1}^{91} X_k * Y_{(i-k) mod (91)}.$$

Circular convolution is a stable operator on vector format. If two vectors of same distribution are composed, the distribution is conserved. Importantly, each component of the binding integrates information from all components of both constituent vectors; information is completely distributed. Thus, by construction, holographic codes are densely distributed, and up to a linear transformation—each vector being centered on zero instead of 0.6 for DGD—their gaussian distribution agrees with DGD densities.

To measure the correspondence between holographic codes and DGD in a general way, we gathered a variety of distance conditions into a single correlation plot. We computed the correlation between DGD and holographic distances in five conditions: a "within-letter" condition (e.g., %%%%A%%%%% versus %%%%%%A%%%) counting 45 distances between a letter across all locations, a "within-string" condition (e.g., %%%%IME%%% versus %%%%I%E%%%%) of 105 distances between all possible substrings obtained by deleting letters from a word, a "between-words" condition (e.g., %%%TIME%%% versus %%%YOGI%%%) of 105 distances between distinct words, a cross-anagram condition (e.g., %%%SWAP%%% versus %%%WASP%%%) of 110 distances between pairs of anagrams, and a shifted-anagram condition (e.g., %%%SWAP%%% versus %%%%SWAP%%) of 110 distances between the same anagram with a one-location shift. This amounted to 475 distances in five conditions. The upper left plot in Figure 9 shows cosine similarities. All similarities were high and regularly placed along the diagonal: first, the relatively weak similarities between words, followed by within-string similarities, and finally high within-letter similarities as well as anagram similarities. Although we will consider the anagram conditions in more detail, what is crucial for this analysis is that DGD and holographic codes coincide to a rather striking extent, with a correlation coefficient of 0.98.

We next asked whether the holographic code could reproduce the density and distance profiles presented in analysis 1. We thus sampled 100 words from the DGD training base, built their holographic codes, and computed the 4950 possible distances between them. The resulting distance distribution obtained for holographic codes is presented in Figure 9 (upper right). This gaussian distribution has a mean 2.5 and standard deviation 2.75. It agrees with the distance distribution in DGD (see Figure 2, right) in that both distributions are unimodal and exhibit large standard deviations. The lower mean in the holographic case (2.5 against 4.86) arises because

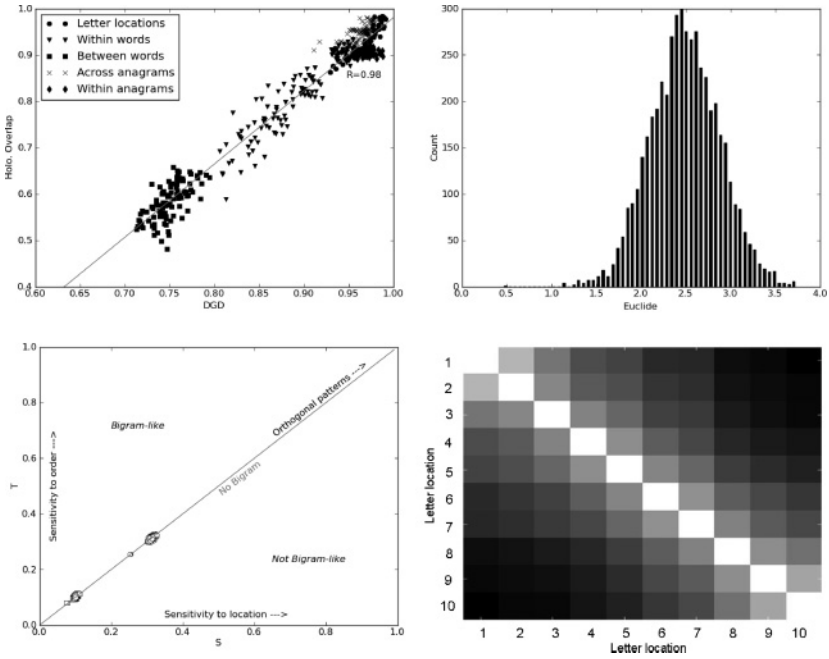

Figure 9: Tests of holographic overlap coding. (Upper left) Correlation of holographic and DGD distances on 475 distances in five conditions. (Upper right) Distance distributions (cf. section 1). (Lower left) S and T indices for bigram patterns (cf. section 3). (Lower right) Letter distances across locations (cf. section 2).

holographic vectors have a much narrower standard deviation than their DGD counterparts (0.11 against 0.53). A less superficial difference is in the distribution skewness, which is zero for holographic codes but negative for DGD. This difference is due to the fact that each code in DGD is also influenced by the whole training base, whereas each holographic code depends on only its constituent letters. Probably the negative skew in DGD reflects neighborhood and letter frequency effects, which may create the asymmetry by isolating word vectors or shrinking their lengths, thereby increasing the number of short distances.

We carried on with the bigram plot of section 3, measuring the extent to which representations were bigram-like. Holographic codes were produced for a sample of 400 bigrams; S and T indices were computed for the Euclidean and cosine distances, and averaged over 50 iterations. The bottom-left plot in Figure 9 shows S and T indices regularly placed on the diagonal, as expected from a letter-based code. S and T indices were indistinguishable in both the Euclidean (mean $S = 0.31$, mean $T = 0.31$)

and cosine (mean $S = 0.10$, mean $T = 0.10$) case. Bigram points were much more tightly clustered compared to DGD, reflecting the absence of vector length variation in the former but not the latter.

Finally, proceeding to the letter analysis, we reproduced the distances between letters seen at different locations that were presented in analysis 2. Distances were averaged over all letters and 50 iterations. Holographic codes are in excellent agreement with DGD. The plot in Figure 9 on the right shows a clear proximity effect, distances increasing with the deviation from the first diagonal. The centrality effect reported in DGD—shorter distances in the center than in the edge—was absent in holographic codes. Again, in DGD, this effect was attributed to differences in occurrence over locations during training and thus cannot be expected to arise in the holographic codes.

In summary, Figure 9 shows a remarkable correspondence between the holographic overlap model and DGD on a number of important criteria:

- Distance patterns within and between words, including anagrams
- Density and overlap distributions
- Letter clusters and the proximity effect

Because it does not take into account the influence of location on vector length, holographic overlap coding cannot reproduce the centrality effect for letters that was illustrated in Figure 3. However, we note that these codes naturally allow for the introduction of weights. These could modulate the importance of any given letter in the string according to location, letter frequency, or letter-positional neighborhood. One might also ask how, despite the absence of such information and a fortiori of any knowledge about a lexicon, holographic codes can still be in good agreement with DGD on the two anagram conditions (Figure 9, upper left plot, diamonds and cross). But closer inspection reveals that distances between pairs of anagrams (resp. between shifted anagrams) are slightly but systematically overestimated (resp. underestimated) by holographic codes. This is reflected in Figure 9 (upper right) in that all crosses (resp. diamonds) lie above (resp. below) the diagonal.

In fact, these deviations of DGD similarities for anagrams can inform us on how the network manages to know whether it has been presented with a shifted exemplar of a given anagram or with exemplars from other anagrams in the same set. Although quite close, mean similarities across anagrams are indeed significantly lower than for shifted anagrams (mean across $= 0.96$, mean shifted $= 0.97$, $p < 0.0001$). Significant differences still arise with a shift of two locations (mean across $= 0.96$, mean shifted $= 0.97$, $p < 0.001$), but the direction is inverted with a three-location shift (mean across $= 0.94$, mean shifted $= 0.96$, $p < 0.0001$). It would thus appear that using its hidden-to-output set of weights, the network can take profit of minute differences in activation patterns in order to sort out anagrams.

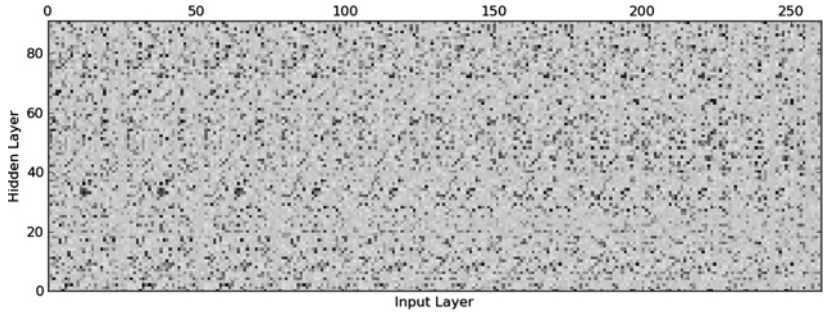

Figure 10: Connection weights between input and hidden units in the DGD network (white: small; black: large)

The coherent picture emerging is that the network has learned a set of connection weights that essentially stores knowledge about semi-location-invariant letters, including letter location frequencies. On presentation of an input, the knowledge is combined to produce a unique code for any string of letters. This combination is not linear, and it integrates information about the whole training base. It is thus quite surprising that the code can be acutely approximated with a simple sum of normally distributed and correlated vectors. More questions arise from the interpretation one should give to such vectors, considering that in the network, hidden activity patterns for any word are built not from the activity patterns of letter constituents but from a localist input vector and a matrix of connection weights.

5.4 Broken Symmetries and Holographic Coding. In order to understand why this correspondence exists, it is necessary to go beyond hidden patterns of activity. The network knowledge that produces hidden activation patterns is contained in the connection matrix between input and hidden units, presented in Figure 10.

Figure 10 suggests that network weights follow the exact same symmetry that the network was trained to achieve on input-output pairs: indeed, weight values appear to be translated every 26 input units. It thus seems that during training, the backpropagation algorithm drives the network onto a trajectory in weight space that converges close to the sufficient conditions described in the generalized group invariance theorem (Shawe-Taylor, 1993). Simply stated and in our case, the theorem holds that translation symmetry on network weights is sufficient to ensure location invariance in patterns. Notice, however, that perfect weight symmetry would require circularly connected input and hidden layers, and would necessarily produce identical activity patterns for anagrams.[10] The former is not true in

[10]More precisely the requirement is of a closed symmetry group, such as, for example, translation when input and hidden layers are circularly connected

DGD, and the network is precisely trained to avoid the latter. For these two reasons, weight patterns in Figure 10 are not exactly translation invariant but appear sometimes lighter or darker from left to right, showing that the symmetry on network weights is broken.

Figure 10 also provides some clues to understand the correspondence we have found: it is really a correspondence between holographic codes and net inputs in the network hidden layer. Recall that since the DGD network uses binary inputs, upon presentation of a word, the net input to any hidden unit is only the sum of four active connection weights (in Figure 10, four points lying on the same horizontal line). Similarly the net input to all hidden units (hereafter, the hidden net input) on presentation of a single letter at a given location is simply a column in Figure 10. It is this column in Figure 10 that corresponds to a letter-location binding vector in the holographic code. Now the meaning of the—lightly—broken symmetry on weights is that weight columns for any letters are obtained by the same transformation—for instance, the transformation to get from column A1 to column A2 is roughly the same as from column B1 to column B2. This is well captured by the binding of letter vectors (e.g., A and B) to location vectors (e.g., 1 and 2). The extent to which the symmetry is broken in network weights is reflected in the overlap between location vectors—the ρ parameter. Finally, the hidden net input for any letter string is simply obtained by summing the adequate columns in Figure 10, which coincides exactly with the sum of vectors in holographic coding.

However, net inputs are only half of what makes a hidden activation pattern—the other half being the sigmoid activation function, which is beyond the scope of these holographic codes. It thus appears that looking at distances between hidden net inputs is a good proxy to characterizing distances between hidden activations, which certainly did not have to be true considering the nonlinearity of the sigmoid function. One possible explanation to this, suggested by the density distribution in Figure 1 (left), is that net inputs are generally close to zero on both sides, which keeps activation patterns close to the middle of the activation range, thus limiting the distortion of distances by the sigmoid (when net inputs are close to zero, the sigmoid function is close to linear). Inspection of the weight matrix indeed shows that weights follow a normal distribution that is centered on zero (mean $= 0.02$), although with a large standard deviation (SD $= 1.95$).

What might perhaps be surprising in this account is that in the end, holographic vectors correspond to weight vectors, or sum thereof, but not to activation vectors, as would have been expected. We find this particularly satisfying considering how holographic reduced representations blur the traditional distinction between connection weights and unit activations. Indeed, these codes were designed precisely so as to reduce associations between representations (traditionally embodied in connection weight matrices) to representations (traditionally an activation vector), so that arbitrary nested associations could be built (Plate, 1995).

6 General Discussion

The main goal of this letter was to understand the string encoding strategy developed by the DGD network for solving a location-invariant word recognition task.

6.1 Understanding the Network. In analysis 1, we found that DGD used densely distributed word representations in order to achieve location invariance. Analysis 2 showed that letter vectors cluster together by location, which suggested that the network had learned semi-location-invariant letter representations. Analysis 3 falsified the bigram alternative and suggested the possibility that any string vector could be well approximated by a simple linear combination of its letter constituent vectors. This was confirmed in analysis 4, where we also found parallels between semi-location-invariant letter representations and the overlap model. Finally, analysis 5 uncovered a remarkable correspondence between the kind of linear mapping achieved by holographic overlap coding and DGD, with a correlation score of 0.99 showing that both generate virtually the same topology on word vectors. We also argued that this correspondence is fundamentally about holographic representations and hidden net inputs and that it arises from a broken translation symmetry in network weights.

Critical in this account is the broken translation symmetry, as it reveals the deep relationship that exists between proximity effects and anagrams. Indeed a network with perfect translation symmetry on its weights would assign exactly the same hidden pattern to exemplars of the same word, but also to anagrams of this word. This network would fail to recognize visual words. On the other hand, a network without translation symmetry would assign orthogonal patterns to anagrams but also to exemplars of the same word. Such a network would not be location invariant. The network that takes the best of both worlds has broken translation symmetry. This network can assign different patterns to anagrams but also assigns very similar patterns to exemplars of the same word. The symmetry appears to be further broken in such a way as to make anagram patterns deviate from a linear mapping (as Figure 7 shows) and as to allow for exemplars of a pair to be distinguished based on distance differences (as Figure 9, upper left, suggests).

Nevertheless one might ask how relevant these insights really are for visual word recognition. How does this correspondence help us understand the network? How general are these results? How plausible is the DGD network from the biological and behavioral viewpoints? These are legitimate questions, and in the following sections, we attempt to answer them in this order.

6.2 Working with a Code Rather Than a Network. From a purely practical point of view, the holographic correspondence that is supported

by this broken translation symmetry has a significant impact. It tells us that as far as word distances are concerned, it is actually not necessary to produce a neural network of 1300 or more sigmoid units and 100,000 or more connections through painstakingly long training. For all intents and purposes, holographic overlap coding makes the same predictions almost instantly using 261 vectors of dimension 91. This would be particularly useful for research in letter position coding, which has recently been using such distances almost exclusively (Grainger et al., 2006; Gomez et al., 2008; Davis & Bowers, 2006; Whitney, in press; Guerrera & Forster, 2008; see Davis, 2010, for an exception).

Apart from the practical advantage, the correspondence can also explain some aspects of the network behavior in a new way. For instance, in holographic overlap coding, the similarity between letter exemplars comes from the overlap between location vectors and, critically the fact that circular convolution respects similarities:

$$\forall i,\ j,\ X,$$
$$\cos(l_i, l_j) = \cos(X \otimes l_i,\ X \otimes l_j).$$

This property and the built-in similarities between location vectors account for clustering and proximity effects. Perhaps unintuitively, the inferior similarity between exemplars of different letters, even at the same location, involves exactly the same property:

$$\forall i,\ j,\ X,\ Y$$
$$\cos(X \otimes l_i, Y \otimes l_j) \leq \cos(X \otimes l_i, Y \otimes l_i)$$
$$= \cos(X, Y)$$

Using the properties of holographic vectors, we could derive other properties of the network, such as the absence of a difference between S and T indices. Indeed, this amounts to establishing the following identity:

$$\forall i,\ X,\ Y$$
$$\cos(X \otimes l_i + Y \otimes l_{i+1}, Y \otimes l_i + X \otimes l_{i+1})$$
$$= \cos(X \otimes l_i + Y \otimes l_{i+1},\ X \otimes l_{i+1} + Y \otimes l_{i+2}).$$

We leave the proof of this identity to future research, along with the interpretation of the ψ vector.

This correspondence is relevant to explain both transposition and relative priming effects reported by Dandurand et al. (2010). In holographic overlap coding, interchanging the positions of two letters does not result in a loss of information on these letters like a double substitution would, because location vectors are correlated. However because close location vectors are

more correlated, the disruption is minimal with central contiguous transpositions. Dandurand et al. (2010) indeed reported very strong transposition priming, with central and contiguous transposition of a single pair of letters being almost as effective as identity primes.

Similarly, deleting a letter in a string not only deprives the holographic code of the corresponding information, but also shifts to the left the location vectors to which letter vectors are bound. Again, since location vectors are correlated, we see that the resulting hidden pattern will still bear some similarity with the original, possibly less so than in the case of single transpositions because some letter information has been purely lost. Dandurand et al. (2010) also reported relative position priming effects in the network, although smaller than in the case of transpositions.

These examples show that thinking of the network in terms of holographic codes can shed some light on its behavior. It also holds the promise to do so with formal proofs, since any of these effects can be translated as an equation involving well-defined vectors and operators.

6.3 Generality of the Results. An important question concerns the extent to which the correspondence holds, and there is no reason to assume that it must be restricted to the visual word domain. Presumably any visual recognition task with similar statistics in the training environment that uses localist, feature-coded inputs and localist outputs would yield the same results. It is easy to devise holographic representations for any such feature-coded visual input, exactly as in the case of words. In fact, at first sight, using localist inputs would appear to be the critical component of this correspondence. This is because localist inputs do ensure that each hidden unit will have a linear combination of exactly these features as a net input, which is well mirrored in the holographic code in that each vector for a string obtains exclusively from the vectors of its letters (as well as a common ψ code). However, it is unclear at this point whether a transition to fully distributed inputs would actually make holographic codes irrelevant, or only the letter-based and overlapping scheme. Indeed, holographic coding is not committed to any scheme, and using distributed inputs might simply mean performing a larger weighted sum of holographic vectors to generate any given string code. Rather, what appears to be mandatory for both localist and distributed inputs is that the net input distribution remains centered on zero so that the (linear) chunking operator can emulate the (nonlinear) activation function. This would appear to be a requirement on the statistics of the training base as much as on input-output format.

Given this localist input-output format, how much does this correspondence owe to the location-invariant nature of the task? In DGD, the requirement for location-invariant recognition forces the network to use very similar weights for any given letter at any location. These agree very well with holographic letter and location bindings, from which every string code is built. As noted previously, there are at least two reasons why DGD uses

semi-location-invariant rather than fully location-invariant representations. First, we have seen that symmetry is broken due to the noncircular layer topology. Second, even with circularly connected layers, the existence of anagrams still forces the network to maintain some location specificity in letter representations. Without anagrams, there would be no need to distinguish between locations since every word would be uniquely defined by its constituent letters, whatever their locations. On the contrary, removing the location-invariant constraint from the recognition task would result in purely location-specific letter representations. It would seem that holographic coding can accommodate the whole spectrum of invariance by adjusting the correlation parameter ρ between location vectors from zero (to cover the fully location-specific case) to a value of one (for the fully location-invariant case).

Finally given localist inputs-outputs and a location-invariance task, one might wonder whether our results are due to this particular network structure. At first sight, it might appear that the correspondence is limited to networks with a single hidden layer. Indeed, although holographic letter vectors correspond to weight columns, the holographic code for a given word is designed to approximate activity patterns in the hidden layer. But this is not taking into account the intrinsic compositional abilities of the code: holographic representations can be combined over and over again in arbitrarily deep structures.[11] It is conceivable that combinations at each level of the holographic structure could correspond to representations in each layer of the network. In this case, because by definition holographic codes have a fixed format, the correspondence must be limited to networks with layers of equal size and in which the density of representations is constant. The idea that backpropagation networks could be used for the same purpose as holographic representations is not completely new. Pollack's (1990) RAAM network is a feedforward autoencoder network that achieves the same goal as holographic representations—representing and accessing compositional structures. There are nevertheless major differences between RAAM and DGD, especially the fact that in the former, hidden representations come to serve as inputs during training.

In summary, our results might hold across various (but not all) input-output formats, training environments, and network structures. The main limitations appear to be on the distribution of net inputs, (which has to be centered on zero), the connectivity (which has to be feedforward), and the size of hidden layers (which has to remain constant). In these conditions, while the encoding scheme might be modified, the holographic correspondence would be expected to apply even for arbitrarily deep hierarchical networks, trained on another visual domain with possibly distributed input-outputs.

[11] The normalized sum must then be used as the chunking operator.

6.4 Plausibility of the DGD Network. From a behavioral point of view, there are some limitations of DGD that prevent one from carrying out more thorough comparisons with human experimental results. First, its base is restricted to four-letter words, which seriously constrains the distance comparisons that can be made and also questions the neighborhood effects induced by this nonrepresentative sample of the human lexicon. Second, letter visibility was assumed to be perfect across all locations, which disagrees with experimental results (Stevens & Grainger, 2003). In order to overcome these limitations, we have carried out new simulations using a larger training base of seven-letter words and introducing realistic letter visibility patterns in network inputs. This study will be presented in a forthcoming paper featuring more extensive comparisons between network distances and human priming. In particular, this will allow the assessment of network performances with respect to some masked orthographic priming results where the overlapping code we have described would be expected to make incorrect predictions. One example is the observed absence of priming when primes are formed of a subset of letters from the target, but the order of some of the letters is changed (Peressotti & Grainger, 1999; Grainger et al., 2006).[12]

Another behavioral concern is that for backpropagation to operate adequately, supervision must be provided, and every word requires several passes into the training base. But although single-shot learning is a desirable property in spoken language learning, where it is known that infants can learn words after one unique exposure (Mayor & Plunkett, 2010), it is unclear to what extent the same holds during reading acquisition. Similarly the argument for unsupervised learning is much more relevant in other areas of language than in string encoding. During reading acquisition, children are explicitly taught how to read words and are given constant visual and auditory feedback in order to do so. At the same time, some researchers have argued that this supervised form of learning using a training base would be compatible with a vision of the hippocampus and the neocortex as complementary learning systems (McClelland, McNaughton, & O'Reilly, 1995), the former repeatedly replaying recent patterns to the latter where they get consolidated. Hence, the supervision and lack of incrementality, although presumably inappropriate in their current forms, may not be critical flaws of the backpropagation algorithm and of the model we considered.

From a biological point of view, backpropagation has also been deemed implausible because it requires calculating input-output errors and propagating them backward in a very precise way (Crick, 1989). At the same time, there is a well-localized region achieving string encoding in the brain, the visual word form area (VWFA; Cohen et al., 2000), which, contrary to the DGD network, has both feedforward and feedback connectivity as well as

[12]We thank Carol Whitney for pointing this out.

limited overlapping receptive fields. However, it is known that backpropagation in a feedforward network is asymptotically equivalent to contrastive Hebbian learning when weak feedback connections are added (Xie & Seung, 2003). Hence, ultimately the one implausible feature of the DGD network might well be the absence of any constraint on receptive fields. Interestingly it has been argued that because it supports the internal organization of the VWFA in a hierarchy of units of increasing complexity (Vinckier et al., 2007), this constraint could be sufficient to produce units that activate for specific open bigrams (Dehaene, Cohen, Sigman, & Vinckier, 2005), as observed, for instance, in Binder, Medler, Westbury, Liebenthal, and Buchanan (2006). We are currently investigating this hypothesis using a hierarchical framework with limited receptive fields inspired from the Visnet model (Wallis & Rolls, 1997).

In summary, behavioral plausibility in the DGD network is limited mostly by its simplifying assumptions of perfect and homogeneous input visibility and by its four-letter-word lexicon. We have argued that biological plausibility might not be hindered by the absence of feedback or the use of backpropagation, since they might in fact cancel out, but by the assumption of full connectivity between layers. These simplifying assumptions were justified by the modeling approach presented in Dandurand et al. (2010), and improving on them is not expected to jeopardize the holographic correspondence (which can accommodate weighted letter inputs and absorb varying proportions of anagrams in the ρ parameter). However, these modifications might trigger a change of granularity in the scheme used by the network, possibly toward using letter combinations as building blocks for visual word codes.

7 Conclusion

The DGD model is perhaps the simplest string encoding network able to achieve location-invariant word recognition. Understanding how it works was the purpose of this letter and a prerequisite to the study of more sophisticated variants. We have been able to establish that the network does not solve location invariance by extracting knowledge about letter combinations from the language environment. Rather, it combines semi-location-invariant letter representations to assign exemplars of the same input word to the same region of hidden layer space and exemplars of different input words to different regions.

Our results were obtained partly by standard data analysis techniques, such as clustering and linear regressions, and partly by introducing new tools, such as transformation indices to characterize bigram knowledge and holographic reduced representations to emulate hidden patterns. In the process, we have uncovered a surprisingly acute empirical correspondence between word distances in the network and in holographic overlap coding. The correspondence stems from a slightly broken translation symmetry

in connection weights. We have argued that the correspondence might hold beyond this particular model for a certain family of networks (fully feedforward and hierarchical, with equal layer format) and for a variety of input-output representations.

Future research could investigate how backpropagation achieves convergence to this solution and disentangle its precise relation to the group-invariance theorem. These insights also pave the way to the study of other network variants that can be more thoroughly compared to behavioral results, bringing us closer to understanding human visual word recognition.

Appendix: Silhouette Fitness Score

Here we recall the silhouette cluster fitness definition (Kaufman & Rousseeuw, 1990). Let us first define the average distance between a point i and cluster X as the distance between i and all points in cluster X, that is:

$$d_{i,X} = avg_{j \in X}(d(i, j)).$$

Let us also note the minimum distance between i and any cluster X as

$$m_i = \min_X \{d_{i,X}\}.$$

The fitness of cluster C is given by

$$f(C) = avg_{i \in C} \frac{m_i - d_{i,C}}{\max\{m_i, d_{i,C}\}}. \tag{A.1}$$

When all elements in C could equally well have been classified in another cluster, the numerator is null and the fitness score is zero. On the contrary, when all points in C are superimposed (and not all points are in C), we have an ideally defined cluster, and this fitness score returns 1.

Acknowledgments

This research was conducted under the ERC research grant 230313 awarded to J.G.

References

Binder, J. R., Medler, D. A., Westbury, C. F., Liebenthal, E., & Buchanan, L. (2006). Tuning of the human left fusiform gyrus to sublexical orthographic structure. *Neuroimage*, *33*, 739–748.

Cohen, L., Dehaene, S., Naccache, L., Lehéricy, S., Dehaene-Lambertz, G., Hénaff, M., et al. (2000). The visual word-form area: Spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. *Brain*, *123*, 291–307.

Crick, F. (1989). The recent excitement about neural networks. *Nature*, *337*, 129–132.

Dandurand, F., Grainger, J., & Dufau, S. (2010). Learning location invariant orthographic representations for printed words. *Connection Science*, *22*, 25–42.

Davis, C. J. (2010). The spatial coding model of visual word identification. *Psychological Review*, *117*, 713–758.

Davis, C. J., & Bowers, J. S. (2006). Contrasting five theories of letter position coding. *Journal of Experimental Psychology: Human Perception and Performance*, *32*, 535–557.

Dehaene, S., Cohen, L., Sigman, M., & Vinckier, F. (2005). The neural code for written words: A proposal. *Trends Cogn. Sci.*, *9*, 335–341.

Ellis, A. W., & Lambon Ralph, M. A. (2000). Age of acquisition effects in adult lexical processing reflect loss of plasticity in maturing systems: Insights from connectionist networks. *Journal of Experimental Psychology: Learning Memory and Cognition*, *26*, 1103–1123.

Gomez, P., Ratcliff, R., & Perea, M. (2008). The overlap model: A model of letter position coding. *Psychol. Rev.*, *115*, 577–600.

Grainger, J. (2008). Cracking the orthographic code: An introduction. *Language and Cognitive Processes*, *23*, 1–35.

Grainger, J., Granier, J., Farioli, F., Van Assche, E., & Van Heuven, W. (2006). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance*, *32*, 865–884.

Grainger, J., & Van Heuven, W. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *Mental lexicon: "Some words to talk about words"* (pp. 1–23). Happauge, NY: Nova Science.

Guerrera, C., & Forster, K. I. (2008). Masked form priming with extreme transposition. *Language and Cognitive Processes*, *23*, 117–142.

Hannagan, T., Dupoux, E., & Christophe, A. (in press). Holographic string encoding. *Cognitive Science*.

Hinton, G. E. (1987). Learning translation invariant recognition in a massively parallel network. In G. Goos & J. Hartmanis (eds.), *PAKLE* (pp. 1–13). Berlin: Springer-Verlag.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: Wiley-Interscience.

Mayor, J., & Plunkett, K. (2010). A neuro-computational account of taxonomic responding and fast mapping in early word learning. *Psychological Review*, *117*, 1–31.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights on the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419–457.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*, 375–407.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Perea, M., & Lupker, S. J. (2003a). Does jugde activate court? Transposed-letter similarity effects in masked associative priming. *Memory and Cognition*, *31*, 829–841.

Perea, M., & Lupker, S. J. (2003b). Transposed-letter confusability effects in masked form priming. In S. Kinoshita and S. J. Lupker (Eds.), *Masked priming: The state of the art* (pp. 97–120). New York: Psychology Press.

Peressotti, F., & Grainger, J. (1999). The role of letter identity and letter position in orthographic priming. *Perception and Psychophysics*, *61*, 691–706.

Plate, T. A. (1995). Holographic reduced representations. *Transactions on Neural Networks*, *6*, 623–641.

Plate, T., Bert, J., Grace, J., & Band, P. (2000). Visualization of feed-forward neural networks. *Neural Computation*, *12*, 1337–1353.

Plaut, D. C., & McClelland, J. L. (2010). Locating object knowledge in the brain: A critique of Bowers' (2009) attempt to revive the grandmother cell hypothesis. *Psychological Review*, *117*, 284–288.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*, 77–105.

Schoonbaert, S., & Grainger, J. (2004). Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*, *19*, 333–367.

Shawe-Taylor, J. (1993). Symmetries and discriminability in feedforward network architectures. *IEEE Transactions on Neural Networks*, *4*, 816–826.

Shillcock, R., & Monaghan, P. (2001). The computational exploration of visual word recognition in a split model. *Neural Computation*, *13*, 1171–1198.

Shultz, T. R., & Elman, J. L. (1994). Analyzing cross-connected networks. In J. D. Cowan, G. Tesauro, & J. Alspetor (Eds.), *Advances in neural information processing systems*, *6*. San Francisco: Morgan Kaufmann.

Simard, P., Victorri, B., Le Cun, Y., & Denker, J. (1992). Tangent prop: A formalism for specifying selected invariances in an adaptive network. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in neural information processing systems*, *4* (pp. 895–903). San Francisco: Morgan Kaufmann.

Smith, M. A., Cottrell, G. W., & Anderson, K. L. (2001). The early word catches the weights. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems*, *12* (pp. 52–58). Cambridge, MA: MIT Press.

Stevens, M., & Grainger, J. (2003). Letter visibility and the viewing position effect in visual word recognition. *Perception and Psychophysics*, *65*, 133–151.

Van Assche, E., & Grainger, J. (2006). A study of relative-position priming with superset primes. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *32*, 399–415.

Vinckier, F., Dehaene, S., Jobert, A., Dubus, J., Sigman, M., & Cohen, L. (2007). Hierarchical coding of letter strings in the ventral stream: Dissecting the inner organization of the visual word-form system. *Neuron*, *55*, 143–156.

Wallis, G., & Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in Neurobiology*, *51*, 167–194.

Whitney, C. (2001). How the brain encodes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin and Review*, *8*, 221–243.

Whitney, C. (2008). SERIOL reading. *Language and Cognitive Processes*, *23*, 143–164.

Whitney, C. (in press). A comparison of the seriol and solar theories of letter-position encoding. *Brain and Language*.

Whitney, C., & Berndt, R. S. (1999). A new model of letter string encoding: Simulating right neglect dyslexia. *Progress in Brain Research*, *121*, 143–163.

Xie, X., & Seung, H. S. (2003). Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Computation*, *15*, 143–163.

Received October 28, 2009; accepted June 14, 2010.