# Computational Models of Location-Invariant Orthographic Processing

Frédéric Dandurand [1+]
Thomas Hannagan [2]
Jonathan Grainger [3]

[+] Corresponding author

[1] Department of Psychology, Université de Montréal,
90 ave. Vincent-d'Indy, Montréal, Québec, H2V 2S9, Canada
Tel: ++1 514 343 4617
Email: frederic.dandurand@gmail.com

[2] Laboratoire de Psychologie Cognitive, CNRS, Aix-Marseille University
3, place Victor Hugo, 13331 Marseille, France
Email: thom.hannagan@gmail.com

[3] Laboratoire de Psychologie Cognitive, CNRS, Aix-Marseille University
3, place Victor Hugo, 13331 Marseille, France
Email: jonathan.grainger@univ-provence.fr

## Abstract

We trained three topologies of backpropagation neural networks to discriminate 2000 words (lexical representations) presented at different positions of a horizontal letter array. The first topology (zero-deck) contains no hidden layer, the second (one-deck) has a single hidden layer, and for the last topology (two-deck), the task is divided in two subtasks implemented as two stacked neural networks, with explicit word-centered letters as intermediate representations. All topologies successfully simulated two key benchmark phenomena observed in skilled human reading: transposed-letter priming and relative-position priming. However, the two-deck topology most accurately simulated the ability to discriminate words from nonwords, while containing the fewest connection weights. We analyzed the internal representations after training. Zero-deck networks implement a letter-based scheme with a position bias to differentiate anagrams. One-deck networks implement a holographic overlap coding in which representations are essentially letter-based and words are linear combinations of letters. Two-deck networks also implement holographic-coding.

**Keywords**: reading; orthographic processing; supervised learning; artificial neural networks; network coding analysis

## 1. Introduction

As they read text, skilled readers of languages that use an alphabetic script must map retinal images of letters onto abstract word representations. More specifically, learning to read involves the recognition of co-occurring letters as part of larger entities, that is, words. Reading also involves an ability to recognize words (strings of co-occurring letters) at various locations on the retina[1], thus achieving location-invariant word recognition. Among the proposed models of the complex processing involved in this cognitive task, many posit a hierarchical system of increasing invariance, in which simple visual features are gradually integrated into more abstract and complex features (e.g., Dehaene, Cohen, Sigman, & Vinckier, 2005). For instance, visual features can be combined into representations of letters which are dependent on their physical attributes (e.g., font, case and location). Then, gradual abstraction from these physical attributes is achieved. High in the hierarchy, abstract (i.e., shape-invariant) letter representations are combined in a word-centered position coding scheme (see below) to finally activate location-invariant lexical representations, that is, words.

Empirical evidence suggests that location-specific letters do not directly activate lexical representations. Instead, evidence coming from a variety of techniques, including masked priming, supports the idea that some intermediate level of representation exists between letters and words. This intermediate level of sublexical orthographic representation is thought to use some form of flexible, word-centered, location-invariant coding of letter position information (see Grainger, 2008 for a review). In general, masked priming experiments involve manipulating the degree of overlap or agreement between the letters that compose a target word and some string of letters used as a prime. Robust priming effects found in skilled readers include transposed-letter priming and relative-position priming. The transposed-letter priming effect describes the superior priming observed from primes formed by transposing two of the target's letters (e.g., gadren-garden) compared with primes formed by substituting

[1] In other words, independently of where an eye fixation is made on a word.

two of the target's letters (e.g., galsen-garden). The relative-position priming effect describes a processing advantage for targets preceded by primes formed of a subset of the target's letters (e.g., grdn-garden) compared with a prime formed of the same subset of letters in the wrong order (e.g., gdrn-garden). Quality of priming is generally measured using reaction times: better primes yield faster reaction times (typically the time it takes to respond that the target stimulus is a word in the so-called lexical decision task - Dufau, Grainger, & Ziegler, 2012). A common explanation for priming effects is based upon activation by the prime stimulus of representations that are subsequently involved in processing of the target word. In other words, prime processing leads to pre-activation of representations shared with the target, which generally results in facilitatory effects on target word recognition, such as seen with transposed-letter primes (Perea & Lupker, 2004; Schoonbaert & Grainger, 2004) and relative-position primes (e.g., Grainger, Granier, Farioli, Van Assche, & van Heuven, 2006; Peressotti & Grainger, 1999). Figure 1 illustrates how common and overlapping representations can activate multiple words which share letters. Inhibitory priming also exists where primes can hinder access instead of facilitating it (Davis & Lupker, 2006; Segui & Grainger, 1990).

Taken together, the evidence for transposed-letter priming and relative-position priming suggests that letter order is important, but only to a certain extent. In order to account for this attested flexibility in letter position coding, a number of models of visual word recognition have proposed an intermediate level of orthographic representation lying in between a location-specific (i.e., retinoptopic) coding of letters and location-invariant word representations. One such model, the Grainger and Van Heuven (2003) model of orthographic processing shown in Figure 1, was the inspiration for a connectionist model in which neural networks learned to map location-specific letter identities (letters coded as a function of their location in a horizontal array) onto location-invariant lexical representations (Dandurand, Grainger, & Dufau, 2010a). The model, trained using 1179 words of four letters, successfully captured transposed-letter and relative-position priming effects. Intermediate representations coded at the hidden layer of these neural networks were found to have two important characteristics (Hannagan, Dandurand, & Grainger, 2011). First, letters appeared to be represented in a semi-location-invariant fashion. Second, representations were well-characterized as a holographic overlap coding in which small changes of the inputs resulted in small differences in hidden layer representations. More specifically, differences in patterns of hidden layer activations were monotonically related to differences in identity and position of input letters. For example, patterns of hidden unit activations were more different for a two-letter substitution (POLL vs. BULL) than a single letter substitution (PULL vs. BULL) when position in the horizontal array was kept constant. Furthermore, larger differences were observed in patterns of activity when an input string was moved by two positions in the alphabetic array (#THAT##### vs. ###THAT###) than moved by a single position (#THAT##### vs. ##THAT####).

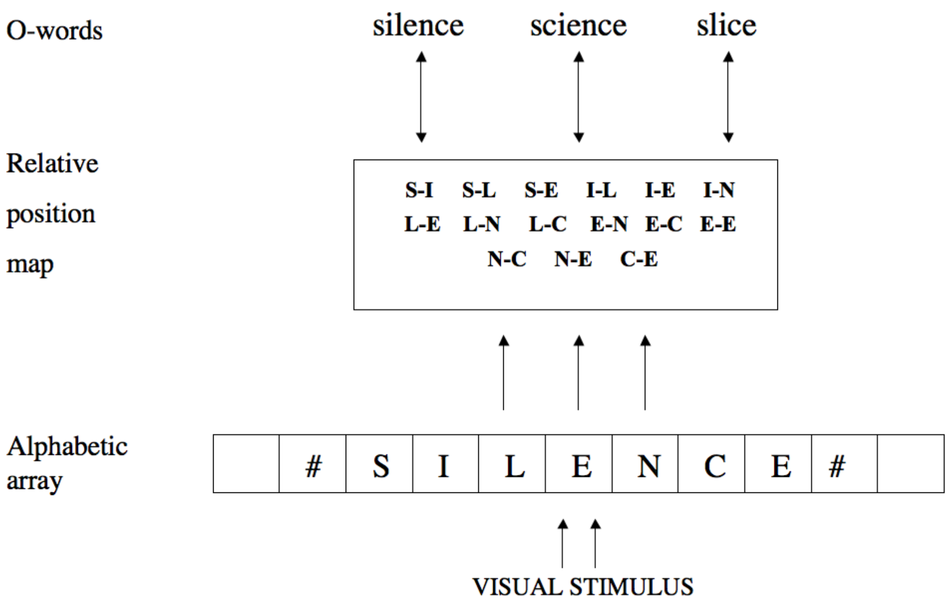


*Figure 1 - Grainger and van Heuven's model of orthographic processing. Visual features of some printed word activate location-specific character detectors along an alphabetic array. These letter identity informations are then combined into a relative position code. This code in turn controls the activations of whole-word orthographic representations (O-words) via bi-directional connections.*

Our previous work  (Dandurand et al., 2010a) showed that, in principle, connectionist models could capture important benchmark phenomena of skilled reading. It showed that learning location-invariant lexical representations leads naturally to the development of a flexible relative-position code. However, the model made an untested implicit assumption that readers learn to directly map location-specific letters to lexical representations. An alternative possibility is that readers learn two distinct levels of representation: first, a word-centered letter level that abstracts away the absolute position of letters on the retina, but maintains within-word positioning, and second, a location-independent lexical representation level. Also, the simulation setup was too limited and artificial to be deemed realistic, namely because words were too short (4 letters) and all letters were equally visible, which is unrealistic for human readers (Stevens & Grainger, 2003).

## *1.1. Research questions*

In the current work, we explicitly test the hypothesis of a word-centered level of representation. We also present key improvements aimed at making simulations much more realistic: (1) inclusion of realistic visibility constraints; (2) use of longer words (7 letters); and (3) testing models against real data from masked priming experiments. The use of seven letter words is particularly attractive because: (1) a rich collection of experimental data exist on priming using this word length, and (2) seven letters allow for fine-grained manipulation of priming phenomena, that is, there are more possibilities for changing letter order in a graded fashion with 7 than with 4 letters. We train networks using the 2000 most frequent words in a French lexicon[2] to match the language of the experimental data to be simulated, and this change in language also allows us to verify that the Dandurand et al. (2010a) results replicate in a different language.

In the current paper, we thus ask: (1) if a model with an explicit level of representation for word-centered letters performs better than a model that does not have such a level of representation; (2) if the simulations of priming effects replicate in a more realistic setting; and (3) what kinds of representations the hidden layer develops, namely whether holographic overlap coding is a suitable descriptive model of what networks have learned, and how such representations explain how networks can discriminate anagrams.

The present models, like the ones from previous work, are fully stimulus-driven by the presence of letters at specific positions, that is, a model of early visual processing. There are no top-down effects of other language-related processing such as phonology, semantics, or other high-level cognitive processes such as attention or working memory. For this reason, when evaluated on their ability to simulate priming effects, model performance is pitted against experimental results for masked priming. Masked priming occurs without awareness and is thought to be essentially stimulus-driven and mediated by early visual processes.

The present work rests on a long tradition of connectionist models for language processing which have roots in the seminal work in Parallel Distributed Processing (Rumelhart, McClelland, & PDP research group, 1986). Connectionist systems have successfully modelled a number of phenomena related to language, including past-tense formation (e.g., Marchman, 1993); pronunciation of text (e.g., Sejnowski & Rosenberg, 1987); and learning of grammar (e.g., Elman, 1991). Along with Dandurand et al. (2010a), the present work on visual word processing is one of the first that concerns specifically the mapping of position-specific letters onto abstract and position-independent lexical identities (i.e., words). As such, it addresses the important question of the nature and characteristics of the internal representations that develop during the process of learning such a skill.

## 2. Methods

### *2.1. Network topologies*

All simulations use standard feedforward neural networks. As mentioned, our previous work (Dandurand, et al., 2010a; Dandurand, Hannagan, & Grainger, 2010b) made an implicit assumption that learning lexical representations from location-specific letters is accomplished directly as a one-step process. The present work allows us to compare this hypothesis with an alternative hypothesis according to which an intermediate level of representation exists and consists of word-centered letter representations.

We thus compare three model topologies for learning the task (see Figure 2). The objective is to determine which model simulates human performance more closely. In the zero-deck topology, networks learn to map location-specific letters onto abstract lexical representations, or words, without a hidden layer. The one-deck topology is similar to the zero deck, but has a hidden layer. Finally, in the two-deck topology, two networks are stacked: a first network learns to map location-specific letters onto word-centered letters, and a second network maps

[2] This is the largest number of words that could reasonably be used in the simulations given the available computer memory and processing speed.

those word-centered letters onto lexical representations. While word-centered letters code for the position in the word (i.e., a slot-coded representation), they abstract away from the position on the retina.

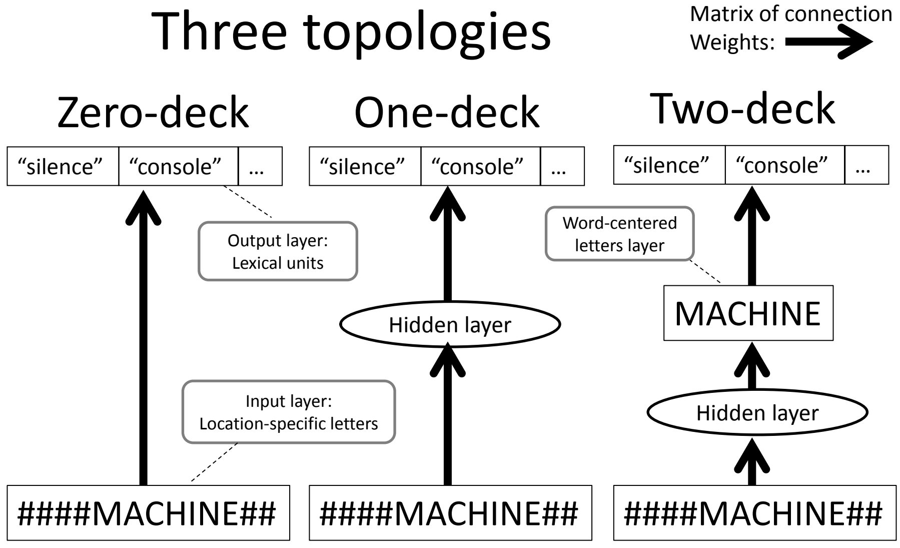

Three topologies
Matrix of connection Weights:
Zero-deck
One-deck
Two-deck
"silence"
"console"
…
Output layer: Lexical units
Word-centered letters layer
Hidden layer
MACHINE
Input layer: Location-specific letters
####MACHINE##

*Figure 2 – Comparison of three topologies for learning lexical representations from location-specific letters. The two-deck topology posits an explicit level called "word-centered letters layer", as an intermediate step.*

Networks' units (that is, neurons) are fully connected in a strictly layered fashion. All units use sigmoidal activation functions. In one- and the first deck of two-deck networks, the number of hidden units is equal to the square root of the number of training patterns rounded up to the closest integer, a standard technique also used in previous work (Dandurand et al., 2010a), yielding 119 hidden units[3]. To study the robustness of results, we also varied depth of training and number of hidden units (see Appendix 1). As shown in Figure 2, zero-deck networks and the upper section of two-deck networks contain no hidden layer; they simply compose lexical units' values as a linear combination of input units.

Other work also specifies an explicit level of representation for word-centred letters. For instance, Dandurand and Grainger (2008) used cascade-correlation neural networks to map position-specific words of four letters onto word-centred representations. These networks successfully learned regularities of the word structure ("wordness"). Furthermore, Shillcock and Monaghan (2001) also used a similar simulation approach (which they called shift invariant identity mapping) to investigate the processing constraints imposed by the visual hemifields. They compared a standard backpropagation model with one in which the input slot is split at its centre, sending these split inputs to two independent processing streams which simulate the visual hemifields. They found that network error was lower for exterior letters in the split model, but not lower in a standard non-split model, which suggests that

[3] sqrt(2000 words * 7 positions / word)

input splitting could account for the superiority effect of exterior (i.e., first and last) letters of words in reading.

### *2.2. Letter visibility*

Empirical evidence suggests that not all letters are equally visible when an eye fixation is made on text (Stevens & Grainger, 2003), including when strings are composed of a random series of consonants (e.g., Tydgat & Grainger, 2009). Two constraints may play a role in determining visibility. First, visual acuity depends on retinal position: it is best at the centre of the fovea, and degrades as eccentricity increases. Thus, the letter at fixation is the most visible. Second, outer letters of words are more visible, which may be explained by the fact that crowding is reduced for outer letters compared to inner letters (e.g., Grainger, Tydgat, & Isselé, 2010; Tydgat & Grainger, 2009).

In the present work, we use empirical data on within-word visibility from Stevens and Grainger (2003) for strings of 7 letters, and different fixation positions. In our models, fixations are always made on the central location of the horizontal letter array that simulates the retina. Fixation on different letters is simulated by shifting the word in the letter array. For example, a fixation on the seventh letter corresponds to an input string of SILENC**E**######; a fixation on the fourth letter to ###SIL**E**NCE###; and a fixation on the second letter to #####S**I**LENCE#. Figure 3 presents an example of within-word letter visibility for a fixation made on the fourth letter. Data for all fixation positions can be found in (Stevens & Grainger, 2003).

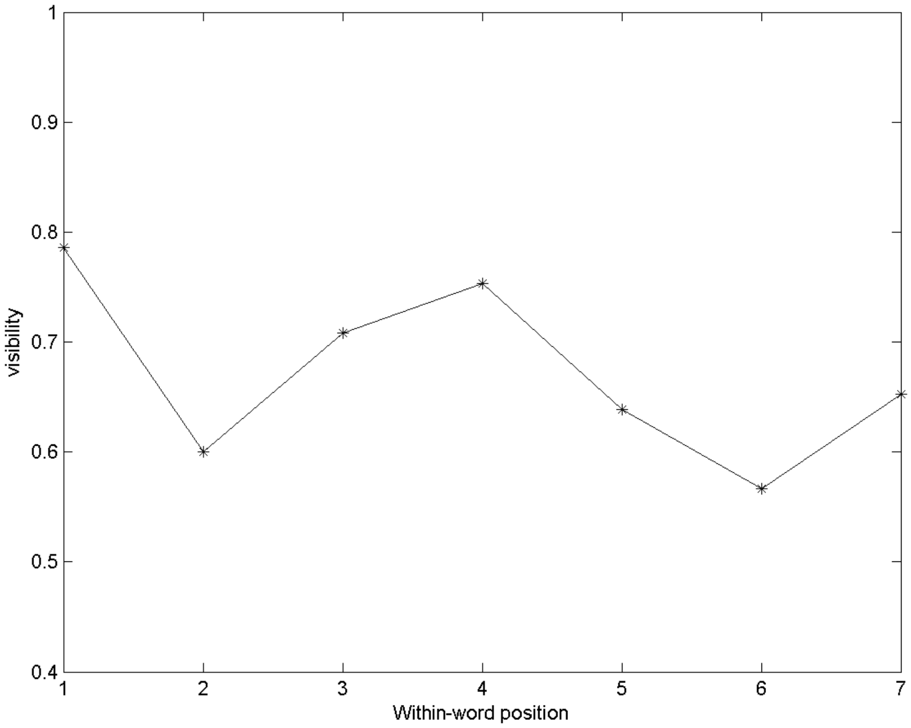


*Figure 3 – Letter visibility (probability of correct letter identification) as a function of within-string position, for a seven-letter string with fixation on the central position (i.e., the fourth letter), from the study of Stevens and Grainger (2003).*

Note that for the two-deck topology, inputs of the second deck -- the word-centred level -- are binary (0 and 1) because outputs of the first deck are binary. The visibility constraint is applied only to the location-specific letters, and not to the word-centred letters. The rationale is that letter visibility is hypothesized to be a perceptual effect related to, namely, visual acuity and crowding. In contrast, word-centred letters are abstract cognitive representations that are learned as independent of retinal position and visibility at the perceptual level (see below for more details on the computation of input values with and without the visibility constraint).

### *2.3. Other parameters*

#### *2.3.1. Input and output coding*

All input and output values are represented using local coding. For one-deck networks, inputs are presented using location-specific letter coding (see Table 1 for an example), and outputs are presented using the lexical unit coding (see Table 3 for an example). In contrast, two-deck networks have two stacked neural networks. The first one receives location-specific letters as inputs, and compute word-centred letters as outputs (see Table 2 for an example). The second network takes those word-centred letters as inputs, and generates lexical units as outputs.

#### *2.3.1.1. Location-specific letter coding*

Words are presented in seven positions along a thirteen-slot alphabetic array; and encoded using local coding. Each letter slot is encoded as a vector of 37 values (26 base letters (a to z) and 11 accentuated French letters[4]). A vector indicating the presence (target = 1) or absence (target = 0) of a given letter is generated. Slots in which no letter is present [0 0 0... 0] represent blanks. As illustration, Table 1 presents the encoded pattern for the word SILENCE in the central position (###SILENCE###). Other examples of inputs include SILENCE######, #SILENCE#####, and ######SILENCE. Elements in this vector of binary values are then multiplied by the visibility value for corresponding slots. Visibility values used are the letter identification probabilities for given positions in the alphabetic array (Stevens & Grainger, 2003) Fixation position is always on slot number 7 (i.e., the central position in the array). Networks are presented with a concatenation of the content of the table into a 482 bits vector (13 slots x 37 values/slot + 1 bias, always set to 1).

| | Presence of letter coding (1 bit per letter) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | À | Â | Ç | È | É | Ê | Î | Ï | Ô | Û | Ü |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[4] Which are: à, â, ç, è, é, ê, î, ï, ô, û, and ü

*Table 1 - Example of a location-specific letters input pattern for word SILENCE presented in central position (###SILENCE###). The first column indicates slot position.*

### *2.3.1.2. Word-centred letter coding*

Coding of word-centered letters is similar to the coding of location-specific letters, except for two differences. First, letters are always presented at the same within-word position, that is, there is only one possible position. Second, letter visibility is always ideal. As mentioned, this choice is justified by the fact that the visibility constraints are likely a low-level visual phenomenon. Word-centered coding is a cognitive construction that abstracts away from retinal position. As illustration, Table 2 presents the encoded pattern for word SILENCE. Networks were presented with a concatenation of the content of the table into a 260 bits vector (7 slots x 37 values/slot + 1 bias, always set to 1).

| | Presence of letter coding (1 bit per letter) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | À | Â | Ç | È | É | Ê | Î | Ï | Ô | Û | Ü |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | $\boxed{1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Table 2 - Example of the word-centred pattern for word SILENCE. The first column indicates within-word position.*

### *2.3.1.3. Lexical unit coding*

Lexical units are also encoded using local coding, with each word corresponding to a distinct unit. When training networks, a target value of 1 indicates the presence of the corresponding word, whereas a value of 0 codes for the absence of the word. Table 3 presents illustrations of lexical unit coding. The number of binary values in the vector equals the number of words in the training set, here 2000.

| Input word | Output vector (3 of 2000 shown) | | | | | |
|---|---|---|---|---|---|---|
| SILENCE | 1 | 0 | 0 | 0 | 0 | ... |
| LUMIÈRE | 0 | 1 | 0 | 0 | 0 | ... |
| MÉLODIE | 0 | 0 | 0 | 1 | 0 | ... |

*Table 3 – Examples of lexical unit coding.*

### *2.3.2. Composition of the training sets*

As mentioned, training sets comprised the 2000 most frequent words from the French lexical database Lexique (New, Pallier, Brysbaert, & Ferrand, 2004), in which only lemma forms were selected. As mentioned, words are presented at all seven locations (with uniform frequency) in the horizontal letter array, for a total of 14000 input patterns.

One of the most difficult aspects of the present task is arguably the segregation of anagrams. While regular words can be discriminated on the basis of differences of at least one letter, anagram identification must rely solely on the relative position of letters within word. In the training set, 3.5% of words have anagrams (N=138), consisting of four triplets (traiter, attire, retrait; étrange, argenté, renégat; respect, spectre, scepter; moindre, dominer, endormi) and 63 pairs.

### *2.4. Network training*

Three networks are trained for each network topology. In each network, connection weights are initialized with random values within a range of -0.5 and 0.5. Training is performed using a standard gradient descent technique (McClelland & Rumelhart, 1988), more specifically the momentum gradient descent technique implemented in the LENS library[5], and using cross-entropy as a cost function (Hinton, 1989). For our simulations, we used a learning rate of 0.9 and a momentum term of 0.2.

### *2.5. Computation of output activations*

To compute activation values of lexical units (i.e., output units of the last layer), an input word is first converted to a location-specific letter coding (see section 2.3.1.1). Second, the visibility constraints are applied by multiplying values in the input vector by the visibility value corresponding to the position in the lexical array. Third, activation values of the next layer are computed in the standard manner[6]. Note that these activation values are continuous and bounded by 0 and 1. In zero-deck networks, there is a single set of weights, and lexical unit activation values are thus computed in one phase. In contrast, for one-deck networks, two phases are required, that is computation of activation values of the hidden layer units, and then activation values of output layer units.

Computation of lexical activation values in two-deck networks is done in two stages. First, activations of the outputs of deck 1 are computed in the same way as for one-deck networks. These output activations code for the continuous, graded representations of word-centred letters (see section 2.3.1.2) as recognized by deck 1. Second, those continuous activation values are used as inputs to the deck 2 network, and lexical unit activations are computed as the outputs of this second deck. In contrast with training which is performed with idealized all-or-nothing (0 and 1) inputs and targets for deck 2, network recall in deck 2 is performed with the actual, continuous outputs of deck 1 without any threshold or other transformation.

Finally, an answer is considered correct when the lexical output unit corresponding to the target word is activated above a threshold value of 0.9, while all other outputs are activated below that same threshold[7]. Networks were trained until they could correctly classify all training patterns, that is, reach perfect accuracy. We empirically found that the following SSE values yielded such accuracy: 100 for zero-deck networks and for decks 1 and 2 of two-deck networks, and 50 for one-deck networks.

### *2.6. Comparison with previous work*

The current simulations improve upon our previous work (Dandurand et al., 2010a, 2010b). Key differences between the current models and previous models are summarized in Table 4.

[5] The LensOSX can be downloaded at: http://hbrouwer.github.com/lensosx/
and the original Unix version at: http://tedlab.mit.edu/~dr/Lens/

[6] By computing the sigmoid-transformed weighted sum of the visibility-scaled input values multiplied by the learned connection weights between inputs and outputs.

[7] This measure is more stringent than the target supremum measure used in (Dandurand, Grainger, & Dufau, 2010). The target supremum measure quantifies the ability of some input pattern (e.g., a prime) to activate the output unit associated with the target word more than any other output unit. However, the measure does not enforce a single output to be above threshold.

| | Current models | Previous models (Dandurand et al., 2010a, 2010b) |
|---|---|---|
| Length of training words | 7 letters | 4 letters |
| Training corpus | (Lexique: New et al., 2004) | (McClelland & Rumelhart, 1988) |
| Number of training words (and thus of output units) | 2000 | 1179 |
| Language | French | English |
| Proportion of words with anagrams | 3.5% | 24.0% |
| Number of training patterns | 14000 | 8253 |
| Number of positions in which words are seen | 7 | 7 |
| Size of alphabetic array | 13 | 10 |
| Number of letters | 37 | 26 |
| Number of input units | 482 | 261 |
| Visibility regime | Realistic | Perfect |
| Zero-deck topology | Yes | Yes |
| One-deck topology | Yes | Yes |
| Two-deck topology | Yes | No |
| Replications per topology | 3 | 1 |

*Table 4 - Key differences between the current model and previous models (Dandurand et al., 2010a, 2010b)*

## 3. Results

### *3.1. Model evaluation test 1: Discriminating words and nonwords*

The first evaluation test consists in determining how trained networks are able to discriminate words from nonwords. Previous work demonstrated that the hidden layer appears critical for success on this task (Dandurand et al., 2010b). As reported in the previous section, networks are able to detect words, as shown by high accuracies on training word patterns.

To test their ability to also reject nonwords, we use four conditions roughly expected to vary in task difficulty[8] from easy to difficult: random string (e.g., HGTQNUK), single repeated letter (e.g., EEEEEEE), double-letter substitution (e.g., SIPENJE), and letter transposition (e.g., SILECNE). Random strings should be easily rejected because they generally share very few letters with any of the words in the corpus. Nonwords made from a single repeated letter should also be rejected easily as they overlap less than 25% with targets[9], and also because the perceptual regularity of the repeated letters is unlike real words. More difficult to reject are nonwords built using a double-letter substitution, that is, by changing two letters (position and replacement letter identities randomly selected) of a word. They have a 71% (5 out of 7

[8] No experimental data exist that directly pits these conditions against each other.

[9] Among the 2000 words in the training set, only a single word contains a letter that repeats 4 times (SENSASS), 80 words contain a letter that repeats 3 times (e.g., ARRIVER), 1275 have two repeats (e.g., PRENDRE), and 644 that have each letter of the word appearing only once (e.g., OUBLIER).

letters) overlap with their associated base words[10]. Finally, the letter transposition condition, in which letters in positions 5 and 6 are interchanged, should be most difficult to reject. This difficulty is reflected in the fact that skilled human readers sometimes mistake these nonwords for words, especially when exposure times are short (e.g., Frankish & Turner, 2007; Grainger, Lété, Bertrand, Dufau, & Ziegler, 2012).

We operationally define word-nonword discrimination as the ability of networks to classify or segregate the distributions of word and nonword patterns based on a fixed cutoff value or threshold. Given some input pattern, we consider a word as being recognized when its corresponding output unit is activated above some threshold. For the four nonword conditions here, no word output unit should be above threshold, meaning that no word is recognized. Any output unit activated above threshold is a false positive error.

Figure 4 presents details of nonword rejection performance for the three topologies.

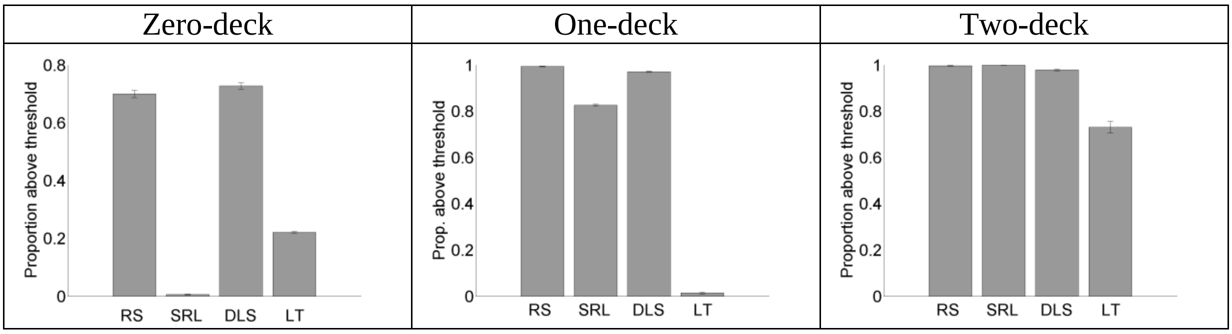


*Figure 4 – Correct nonword rejection rates at threshold level 0.9 for the four conditions (RS = Random String (e.g., HGTQNUK), SRL = Single Repeated Letter (e.g., EEEEEEE), DLS = Double-Letter Substitution (e.g., SIPENJE), and LT = Letter Transposition (e.g., SILECNE)) with standard error bars.*

We see that zero-deck networks fail to correctly reject nonwords made of a single repeated letter (e.g., EEEEEEE). One-deck networks perform better at rejecting these repeated-letter nonwords, but still fail to surpass performance on substitution nonwords. Only two-deck networks have the correct pattern of nonword rejection, and also perform the best on nonwords made with letter transpositions.

### *3.2. Model evaluation test 2: Simulating masked priming effects*

Next, we investigate the ability of networks to simulate the pattern of facilitatory priming effects observed in humans with the masked priming paradigm. Priming in networks is based on the principle that the greater the orthographic overlap between prime and target stimuli, the greater the target output is activated. More specifically, priming is operationally defined as follows: a word is considered primed by some input if its corresponding lexical output unit is activated above some threshold. Thus, rather than measuring the influence of a prime stimulus on subsequent target word identification, as in human studies, here we directly

[10] In the previous study with four letter words, the correct rejection rate of nonwords was 94.1% (Dandurand, Grainger, & Dufau, 2010). These nonwords were generated using a single-letter substitution. As a result, they had a 75% overlap (3 out of 4 letters) with a word (e.g., nonword AHLE based on word ABLE). We chose double-letter substitutions in the present study to approximately match this degree of overlap.

examine how well a given prime stimulus can activate a given target output. Note that the thresholds used here differ from those used for word-nonword discrimination tasks. In the discrimination task, a higher threshold value (0.9) is used to show that there exists a clear activation boundary between words and nonwords that networks can use to classify input strings. The threshold value is high to reflect a high confidence that some input string is indeed a word, thus reducing false positives errors. On the other hand, in the priming simulations presented here, the goal is to identify effects of weaker activations of a prime stimulus prior to stimulus classification, so a lower threshold value is used (0.5). In this way we measure how well a given prime stimulus can activate a given target output relative to a condition where there is no prime and therefore zero target output activation.

### *3.2.1. Relative-position priming*

Relative-position priming effects are measured under two conditions. In the first condition, the first four letters of the target word (1, 2, 3, and 4) are presented as a masked prime. For instance, for word SILENCE, the corresponding prime is SILE. In the second condition, odd letters are used (1, 3, 5 and 7). For word SILENCE, this prime corresponds to SLNE.

Results are presented in the left column of Figure 5. Data for relative-position priming in humans are from Table A2 (p. 883) in Grainger et al. (2006). Priming effects in humans is measured as the improvement (shortening) in response time when a target is preceded by a masked prime compared to a control condition. Note that the priming effect in humans is significant (*p*<0.05) for the condition 1234, but not for 1357, showing larger priming with letters 1234 than with letters 1357. Therefore, in models we expect priming with letters 1234 to be large, whereas priming with letters 1357 should be smaller.

As we can see in Figure 5, model evaluation test results suggest that the zero-deck and the two-deck networks successfully simulate the larger priming effect with primes formed of letters 1234 than 1357, whereas one-deck networks do not.

### *3.2.2. Transposed-letter priming*

Transposed-letter priming effects are measured under two conditions. In the first condition, letters at positions 4 and 5 are transposed, yielding a prime composed of letters 1235467. For instance, for word SILENCE, the corresponding prime is SILNECE. In the second condition, letters 4 and 5 are replaced with unrelated letters, yielding a prime of form 123DD67. For word SILENCE, an example of prime would be SILOPCE.

Results are presented in the right column of Figure 5. Data for transposed-letter priming in humans are from Tables 4 and 5 in Schoonbaert & Grainger (2004). Note that the priming effect for condition 1235467 was shown to be significant (*p*<0.001) whereas priming with 123DD67 was not (*p*>0.05), suggesting that priming effects are larger with 1235467 than 123DD67. Test results suggest that all three network topologies successfully simulate the larger priming effect for condition 1235467 than 123DD67.

| | Relative-position priming | Transposed-letter priming |
|---|---|---|

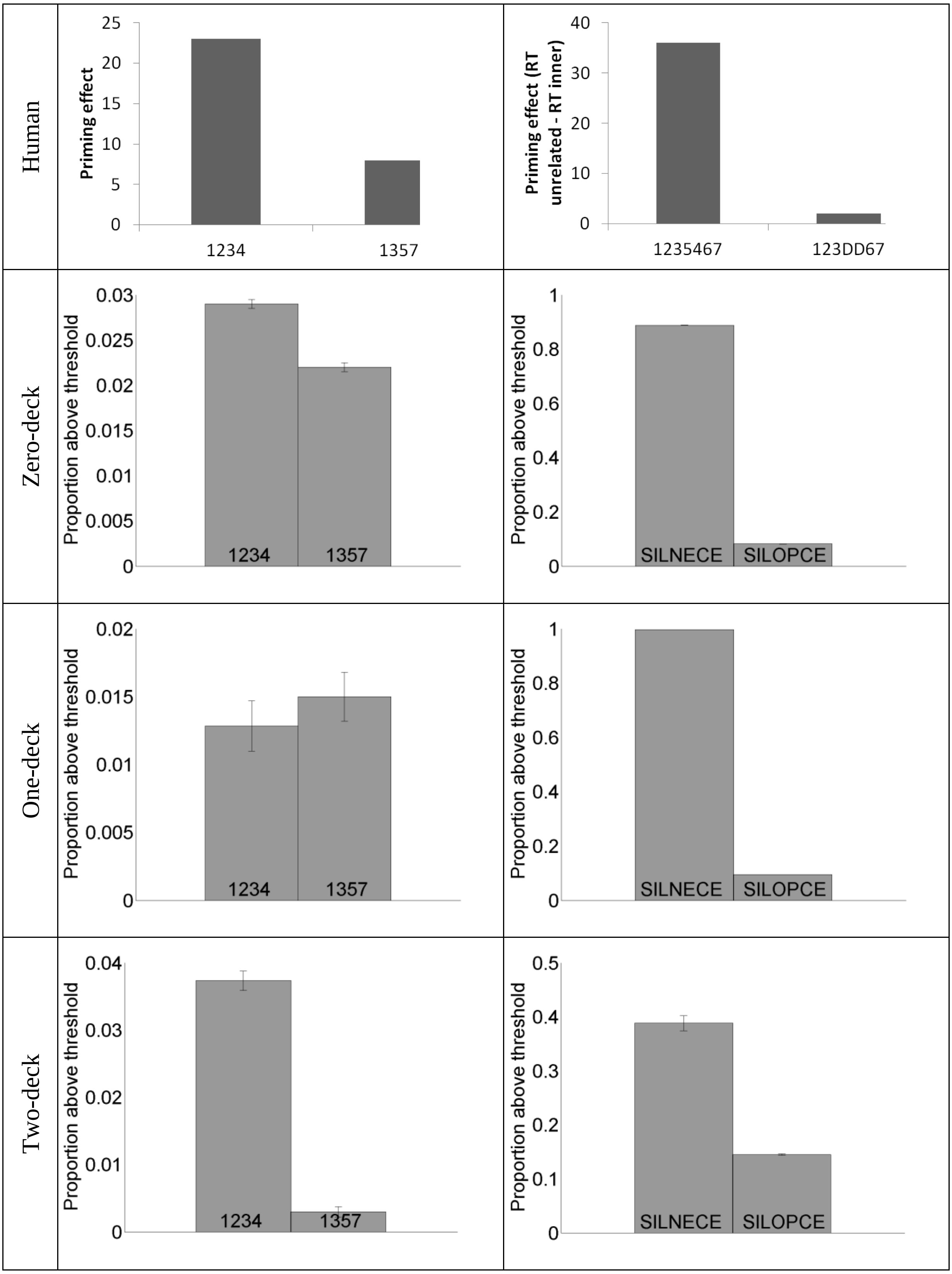


*Figure 5 – Relative-position and transposed-letter priming results comparing human with model performance for the three different topologies, with standard error bars.*

### *3.3. Summary of evaluation tests*

A summary of the performance evaluation tests is presented in Table 5. As we can see, only the two-deck model simulates all three phenomena covered. Before we can conclude the model comparison analysis, we need to take into account another important factor: model size.

| Model | Word-nonword discrimination | Relative position priming | Transposed letter priming |
|---|---|---|---|
| Zero-deck | No | Yes* | Yes |
| One-deck | No | No | Yes |
| Two-deck | Yes | Yes | Yes* |

*Table 5 – Summary of performance evaluation tests indicating if models successfully simulate human performance (Yes) or not (No). *: the magnitude of the effect is smaller, but the qualitative pattern is correct.*

### *3.4. Model size*

Conventional statistical wisdom states that the capacity of some model to cover some pattern of result has to be pitted against the number of degrees of freedom that the model has. Preferred models provide a good fit to the experimental data while having as few degrees of freedom as possible.

For our models, we operationally define degrees of freedom as the number of trainable connection weights in a network. Zero-deck networks have 964 000 connection weights (482 inputs x 2000 outputs); one-deck networks have 297 358 (482 inputs x 119 hiddens + (119 hiddens + 1 bias) x 2000 outputs). Finally, two-deck networks have only 72 198 units (482 inputs x 119 hiddens + (119 hiddens + 1 bias) x 7 letters + 7 letters x 2000 words).

This analysis provides a converging picture about the superiority of the two-deck topology. Not only does it best fit the pattern of experimental data, it also does it with the fewest degrees of freedom[11].

## 4. Analysis of internal representations

As we have seen, despite some differences in their ability to reject nonwords and to simulate priming effects, all three network topologies were able to learn the task to a near-perfect accuracy. We now investigate what knowledge networks acquire as they learn to solve the task, or in other words, what internal representations do networks develop? What are they encoding for? And also, how does this encoding allow networks to discriminate or segregate anagrams?

Analyzing neural network representations can be challenging due to the distributed nature of the knowledge in large matrices of connection weights. Can we find patterns or regularities in these numerous connections of our trained networks? In other words, can we describe the processing that connection weights collectively accomplish using simple, rule-like or mathematical terms?

[11] Further statistical tests are unnecessary here because there is no compromise or tradeoff between size and quality of fit (i.e., performance). A better fit with a smaller model is clearly the best option.

Our approach consists in two kinds of analyzes. On the one hand, we analyze the pattern of activation at the hidden layer for network topologies that contain such a hidden layer, that is, the one-deck network and the lower part of the two-deck network. The rationale is that hidden layer activations summarize the processing performed by the input-to-hidden connection weights. On the other hand, in the absence of a hidden layer, we need to directly look at the matrix of connection weights for the zero-deck network and the upper layer of the two-deck network.

### *4.1. Zero-deck topology*

Intuitively, zero-deck networks should use letters present in the target word as positive evidence for the target. In contrast, letters absent from the target provide negative evidence for the target. In our models, this translates into large connection weights from target words to letters present in them to enforce that these letters activate the target, while small or negative weights to letters absent from targets enforce non-activation or inhibition of the target. For instance, for target word SILENCE, letter S provides positive evidence and should be associated with a large positive connection weight, whereas letter K provides negative evidence and should be associated with a small or negative connection weight.

It is important to note that letters present in the target provide positive evidence at all positions in the horizontal array where they were trained. Networks have no hidden layer to consolidate signals from different input positions; therefore, all letters provide independent votes for the target. For example, both S############# and ######S###### are evidence for word SILENCE because the network learned to recognize SILENCE###### and ######SILENCE as instances of word SILENCE. Pattern ######S###### is also evidence for word CONSOLE because the network has learned pattern ###CONSOLE###. Letters near or at the extremities of the array may not be evidence for some target word, for instance the word SILENCE was never trained with a S at the last slot (############S).

We hypothesize that weights for letters present in the target are larger than those absent from the target, irrespectively of letter position for the positions trained. Figure 10 shows histograms of connection weights' magnitudes, normalized by the number of weights in each category[12]. As we can see, the hypothesis is confirmed: weights are on average larger for letters present in the target than letters absent.

[12] There are much fewer connection weights for letters present in the target word than weights for letters absent. Normalization aims at equating the areas under the two histogram curves.

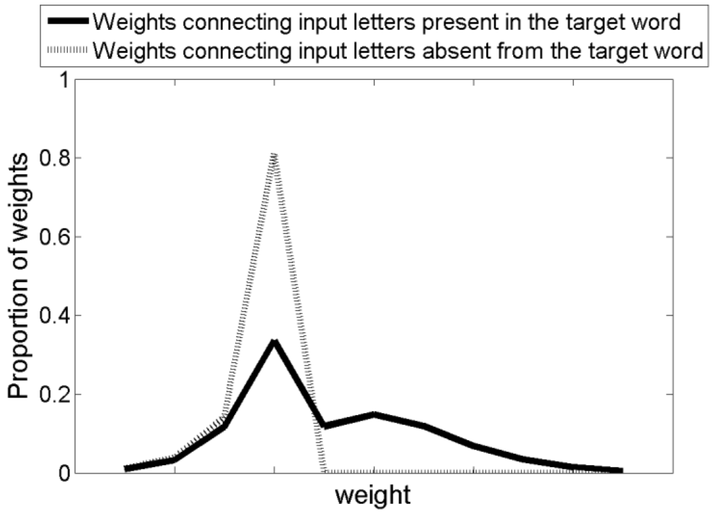


*Figure 6 – Histogram of connection weights in the zero-deck networks. Only non-empty bins are presented, in increasing order of weight magnitude.*

What we have seen so far is that presence of the target letters provides evidence for some target largely irrespective of where they occur. For example, letters (in alphabetical order) *a, e, i, r* (x2) and *t* (x2) provide evidence for words *traiter, attire,* and *retrait*. We next need to address how the zero-deck networks manage to discriminate anagrams since they cannot directly capitalize on letter position. To gain some insight into the segregation of anagrams, we compute the average connection weights for letters in the target only as a function of within-word position (also called word-centered position) and slot in array. Results are presented in Figure 7, for all words in the training set, and for the small subset of words that have anagrams.

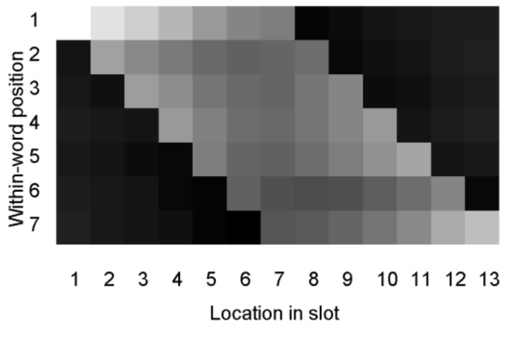
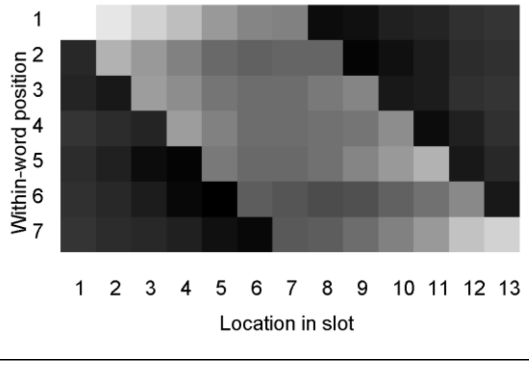
| All words | Anagrams only |
|---|---|
|  |  |

*Figure 7 – Color-coded average magnitude of connection weights as a function of within-word position and location in the input slot. Results are presented for all words (words with and without anagrams) and specifically for words that have anagrams.*

As we can see, there is a negative correlation between within-word position and location in slot for the first letter in the word (position 1), that is, connection weights decrease with the location in the slot. In contrast, the correlation is positive for the last letter (position 7), that is, weights tend to increase with location in slot. To test for this interaction, we performed a two-way ANOVA with within-word position and location in slot as two repeated factors. This within-word position by location in slot interaction is significant, $F(72,1928) = 979$, $p<0.001$. Main effect of position, $F(12,1988)=1984$, $p<0.001$, and of location, $F(6,1994)=354$, $p<0.001$, are also significant.

We can now understand how zero-deck networks segregate anagrams. Although they cannot use precise location of letters, this scheme allows them to weigh more the letters that appear toward the beginning of some word when they also appear towards the beginning of the slot, and vice-versa for letters that appear toward the end of the word, which is sufficient to discriminate between anagrams. For example, letter A is more likely to occur toward the beginning of the slot (input array) for word *attire* than in word *retrait.* In contrast, letter E is more likely to occur towards the end of the array for word *attire* than for word *retrait*. By capitalizing on this difference, networks can appropriately activate *attire* more than *retrait* when A precedes E in the array, and the opposite when A follows E. Note that this scheme works because letters are more likely to occur in some locations than others due to non-uniform training. In the training set, letters that occur early in words are more likely to also occur early in the array, and vice-versa. In a hypothetical circular array without a beginning and an end, this scheme would not work.

In sum, the processing strategy or coding scheme that zero-deck networks develop appears to be primarily based on the number of letters shared between inputs and targets independently of position. This implements a sort of voting scheme in which input letters provide independent votes for the target words, positive votes for words that contain the letters and abstentions or votes against the words that do not contain these letters. Letter presence is then modulated by the interaction between location and position, which allow networks to use the correlation between location and position to factor in some information about the relative position of letters, allowing anagrams to be discriminated. This coding scheme also accounts for the priming effects: larger priming as the number of letters shared between primes and targets increase, and larger priming as the agreement increases between the order of letters in the prime and in the target.

### *4.2. One-deck topology*

For the one-deck networks, we analyze the activations at the hidden layer. Those activations summarize the processing performed by the input-to-hidden connection weights. Previous analyses of a one-deck network learning four letter words in a ten-slot array uncovered two important characteristics of the coding at the hidden layer (Hannagan et al., 2011). First, coding was found to be primarily letter-based in a semi-location-invariant fashion, with no evidence for the coding of bigrams, i.e., letter pairs. Second, representations at the hidden layer were well-characterized as a holographic overlap coding in which small changes of the inputs resulted in small differences in hidden layer representations. More specifically, differences in patterns of hidden layer activations were monotonically related to differences in identity and position of input letters that is, a proximity effect. In overlap coding, this is explained by using a probability function for letter position rather than a fixed value. The probability is highest for the position where the letter is actually presented and then drops off in a monotonic fashion as a function of distance from actual presentation (Gomez, Ratcliff, & Perea, 2008).

For the present networks, we expect to observe a similar coding strategy. First, we test the hypothesis that coding is primarily letter based, by calculating distances between activation patterns at the hidden layer for all letters (including accentuated) and all positions (i.e., A############, #A###########, ..., ############A, B############, ..., #######B#####, ... ############B, ..., ###K#########, ..., ##########S##, ... #######Z#####, ... ############Ü). If letter coding is used at the hidden layer, patterns of activations should better cluster (i.e., differences should be smaller) for the same letter at different position (e.g., #######B##### vs. ############B), than any pattern of any other letter (#######B##### vs. ##D##########). As we can see in the first row of Figure 8, the hypothesis is confirmed, especially for letters which are more frequent (the effect is smaller in the lower-right quadrants of the graphs, corresponding to accentuated letters which are relatively infrequent).

Second, we test for the proximity effect by plotting differences in patterns of activity for some given letter as a function of position in the array (averaged across all letters). Similarity between activation patterns should be inversely related to distance. As we can see in the second row of Figure 8, the hypothesis is also confirmed.

Third, we investigate how word representations are built. We hypothesize that they are simply built as a linear combination of letter patterns. For instance, the activation pattern for ###SILENCE### is equal to ###S######### + ####I######## + #####L####### + ######E###### + #######N##### + ########C#### + #########E###. Using a linear regression of word patterns against their component letter patterns, we find that a fair amount of variance can be explained by this scheme ($R^2$ = 0.47, $p$ <0.001). Part of the departure can probably be explained by the fact that some words are easier to recognize than others based on their constituent letters, and that some letters may be ignored or downplayed in forming a representation for the word. For instance, for words that contain infrequent letters that are sufficient to uniquely identify it, networks may capitalize on these infrequent letters, largely ignoring frequent letters.

Finally, we test the holographic overlap coding hypothesis by computing the correlation (R) between two sets of distances: first, the distances between activation patterns elicited by a number of input strings presented to the network, and second, the distances between holographic overlap coding patterns for the same strings. We find near-perfect agreement with the pattern of activation of hidden units (R = 0.96).

| Letter cluster | Proximity effect |
|---|---|

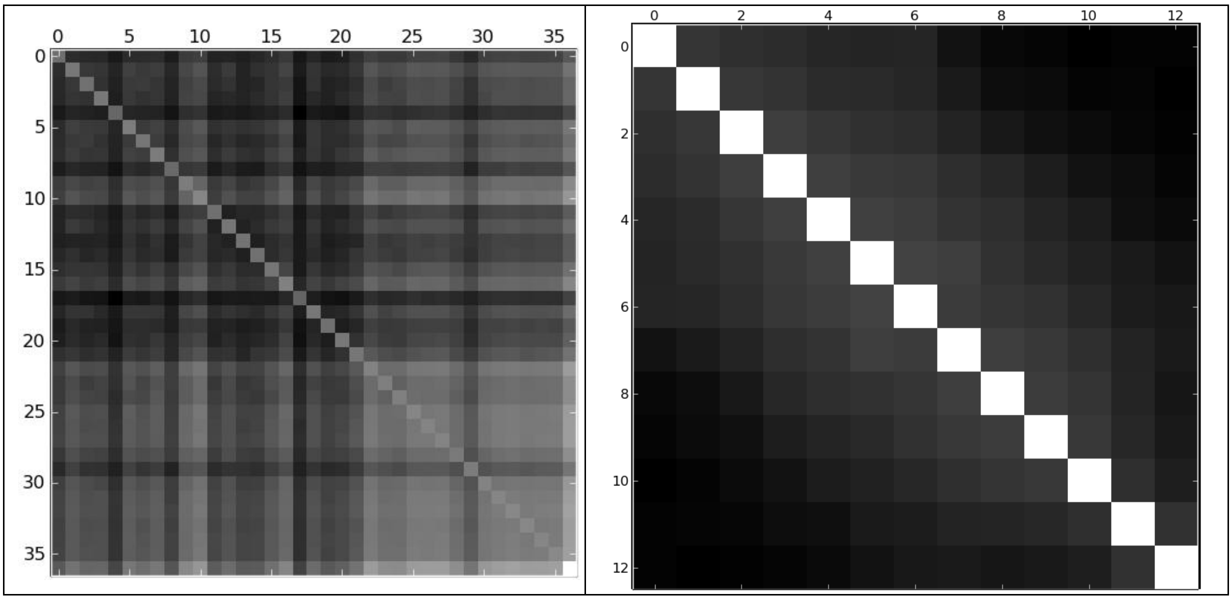


*Figure 8 - Results of analyses of internal representations at the hidden layer of one-deck networks: letter cluster, and proximity effect.*

In sum, the same conclusions about one-deck network coding can be drawn as previously reported (Hannagan et al., 2011): letters are represented in a semi-location independent fashion (letter cluster and proximity effect), and a fair amount of variance of the activation patterns for words can be explained by a simple linear combination of individual letter patterns. Hidden pattern activations are in near-perfect agreement with holographic overlap coding.

### *4.3. Two-deck topology*

We now turn to an analysis of the two-deck networks; performing separate analyses for the two decks.

### *4.3.1. From location-specific letters to word-centered letters*

The lower deck performs the mapping of location-specific letters to word-centered letters. Although the task differs in terms of outputs units, we hypothesize that the representations at the hidden layer will be similar to those of the one-deck models. More precisely, we expect to find semi-location invariant letter-based representations subject to the proximity effect. We also expect that variance of activation patterns at the hidden layer ($R^2$) can be explained by letter combination (e.g., SILENCE = S############# +  #I############ + ##L##########, …) and holographic overlap coding. Results are presented in Figure 9.

| Letter cluster | Proximity effect |
|---|---|

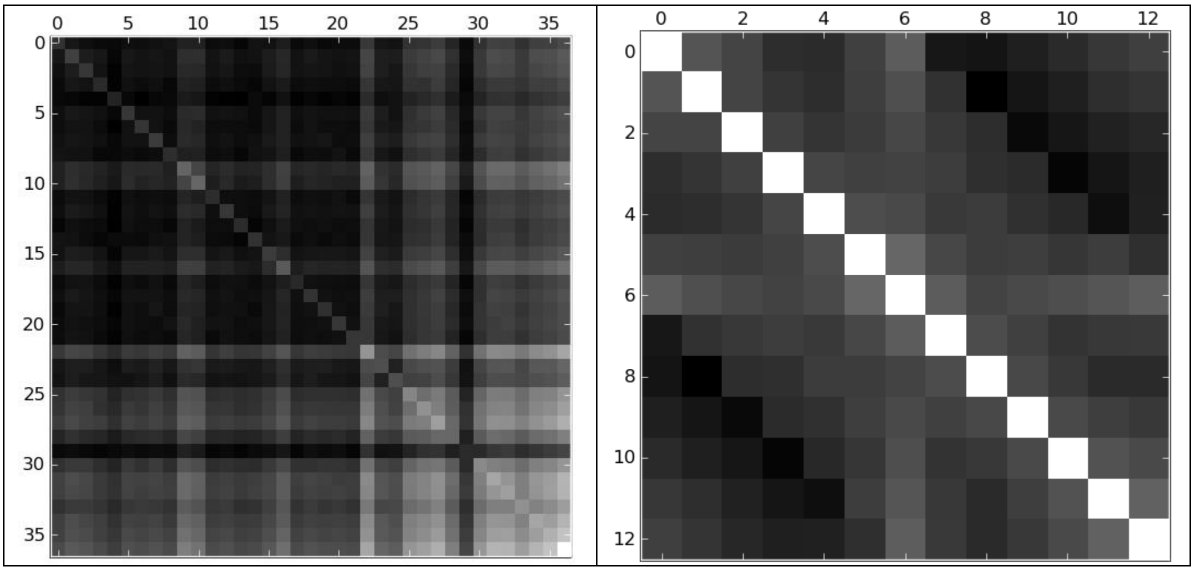


*Figure 9 - Results of analyses of internal representations at the hidden layer of lower decks of two-deck networks: letter cluster, and proximity effect.*

We find a good correlation between patterns of activation at the hidden units and holographic overlap coding (R=0.90). However, the hypothesis that word patterns are built as linear combinations of letter patterns does not explain much of the variance ($R^2$=0.25).

In sum, only the holographic overlap coding hypothesis was confirmed. For the rest, the coding is more difficult to interpret, with no clear letter-based coding, nor proximity effect. The authors have performed a number of additional analyses looking for patterns and abstract explanations, all of which have remained unfruitful. It may be that processing in the lower decks of the two-deck networks cannot be described in simple terms (i.e., the weight matrix itself is the simplest explanation), or maybe future work will reveal some pattern.

### *4.3.2. From word-centered letters to lexical units*

Finally, we turn to the analysis of the upper deck of the two-deck networks. The task of the network is to map word-centered letters onto lexical output units. This task resembles the one performed by zero-deck networks, but with an important difference: position of letters in the seven-slot long array of word-centered letters is fixed and therefore directly informative. For example, letter S for word SILENCE is always seen at position 1. As a consequence, letter S at the fourth slot position is positive evidence for word CONSOLE, but negative evidence for word SILENCE. We therefore hypothesize that connection weights are larger for letters at the correct position than any other weights (letters absent from the target and letters present but at an incorrect location). The histograms presented in Figure 10 confirm this hypothesis.

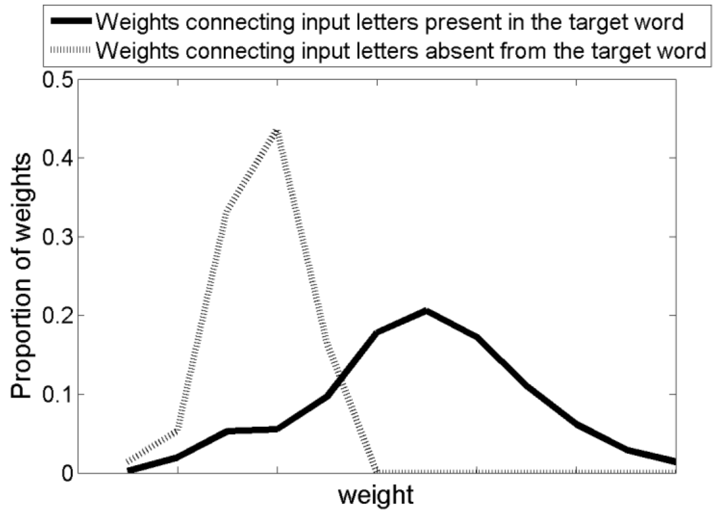


*Figure 10 – Histograms of connection weights for the upper section of the two-deck network. Only non-empty bins are presented, in increasing order of weight magnitude.*

Processing in the upper decks of the two-deck networks is therefore straightforward to interpret. With large connection weights, letters at the correct positions increase target word activation, whereas everything else is either ignored or inhibits the target.

## 5. General discussion

To summarize the results, the three topologies studied (zero-, one- and two-deck) were able to learn the mapping task from location-specific letters onto abstract lexical units to a near-perfect accuracy, including the discrimination of anagrams. Networks generally performed well at rejection of nonwords, but only the two-deck topology correctly rejected strings of repeated letters more readily than single letter substitutions, which is what we expect from skilled human readers. Thus, among the three topologies studied, the two-deck topology best captures human performance, and is thus the most adequate cognitive model. It is also the model with the fewest degrees of freedom, as measured by the number of trainable connection weights.

Except for the relative-position priming effect in the one-deck networks, networks showed the correct pattern of relative-position and transposed-letter priming effects. These effects can be explained by the flexible orthographic representations schemes that networks developed. In the case of the zero-deck networks and the upper deck of the two-deck networks, the scheme is based on presence of the letters in the target word: letters present in and absent from the target word counting, respectively, as positive evidence (with positive connection weights) and as negative evidence or abstention (small or negative connection weights). Anagram segregation in the upper deck of the two-deck networks is straightforward because word-centered letters are in fixed position. In contrast, zero-deck networks include a bias to weigh more letters' votes when their positions in words match location in the array, allowing them to segregate anagrams. For one-deck and the lower deck of two-deck networks (see Table 6 for a summary), we found a large agreement between patterns of activation at the hidden layer and holographic overlap coding. While a finer description of network processing remains elusive for the lower deck of the two-deck network, processing in one-deck networks can be interpreted as implementing letter-based coding where words are represented as linear combinations of letter representations. Results of the one-deck networks replicate those from earlier studies (Hannagan et al., 2011), suggesting that representations learned at the hidden layer are robust. More specifically, the very same representations were found with a different set of simulation parameters, namely word length, word count, and language.

| Characteristic | One-deck topology | Two-deck topology |
|---|---|---|
| Letter-based coding | Yes | No |
| Proximity effect | Yes | No |
| Word representations approximately equal to linear combinations of letter representations | Yes | No |
| Holographic overlap coding | Yes | Yes |

*Table 6- Summary and comparison of hidden layer representations in the one- and the two-deck networks*

The triangle model and its successors (e.g., Harm & Seidenberg, 2004; Plaut, McClelland, Seidenberg, & Patterson, 1996; Seidenberg & McClelland, 1989) propose an overall cognitive architecture for reading, including not only orthography, but also phonology and semantics. In contrast, our model is exclusively concerned with orthographic processing. We are, of course, well aware of the evidence for an early involvement of phonological representations during silent word reading (e.g., Grainger & Ferrand, 1996; Grainger, Kiyonaga, & Holcomb, 2006). The present modeling work should therefore be seen as an attempt to capture one key component of the overall reading network as described in the triangle model and alternative theoretical frameworks such as the bi-modal interactive-activation model (BIAM: Diependaele, Ziegler, & Grainger, 2010; Grainger & Holcomb, 2009) and other dual-route models (DRC: Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; CDP+: Perry, Ziegler, & Zorzi, 2007). This allows us to focus on what could be considered to be the "hard problem" in visual word recognition: that is how location-specific visuo-orthographic information is transformed into a location-invariant orthographic representation. In this respect it is important to note that computational models that attempt to capture the overall reading network (triangle model, DRC, CDP+, BIAM) all use some form of word-centered letter representation as input.

### *5.1. Lexical decision*

In the present paper, we have been careful to use the term word-nonword discrimination because we are not claiming that networks implement a complete model of lexical decision (see Dufau et al., 2012 for a recent example). Still, rejecting strings of letters as nonwords is not trivial because networks never see counter-examples; they are always trained with positive evidence for words, and thus may be expected to over-generalize.

### *5.2. Robustness of results*

The same pattern of results was robustly found for different depths of learning (Appendix 1), number of hidden units (Appendix 2) and cost functions (Appendix 3), with only one exception relative to the influence of the number of hidden units, to be discussed below. Also, previous results obtained with zero- and one-deck networks (Dandurand et al., 2010a, 2010b) trained with four letter words and perfect letter visibility were replicated here with seven-letter words, realistic letter visibility, and a larger vocabulary. Furthermore, we found that zero- and one-deck networks developed the same internal representations when learning four and seven-letter words. The principles uncovered in the section on the analysis of internal representations are general, and should readily generalize to more realistic training sets containing more words and words of different lengths. The failure of the one-deck model with more hidden units to reject repeated letter nonwords (see Appendix 2) suggests that the greater number of hidden units encourages a more location-specific coding of letter identity which more closely resembles the input coding. In this way, with more hidden units the one-deck network begins to resemble the zero-deck network (compare Figure 12 with Figure 4).

### *5.3. Evolutionary argument for reading*

Dehaene and Cohen (2007) have argued that reading is too recent in evolutionary time to have prepare us with specific adaptations for visual word recognition. Instead, they propose the neural recycling hypothesis: cultural learning in humans, including reading, relies on reconverting pre-existing cerebral predispositions for novel use, within what is possible given the strong genetic constraints on cerebral structures. More specifically, an area called the "visual word form area" (VWFA) is consistently activated during visual word recognition (Dehaene, Le Clec'H, Poline, Le Bihan, & Cohen, 2002). Today any theory on how humans can recognize words should therefore be informed by, and consistent with, what has been firmly established about the organization of the VWFA. Our comparison of several models ultimately favors the deepest hierarchical network, which displays a total of 4 layers and which involves a transition from a location specific level to a location invariant but position specific level. This is consistent both with experimental constraints on location specific and position specific levels of representations in this region (Dehaene et al., 2004), and more generally with empirical results demonstrating that the VWFA is organized in a hierarchy of levels (Vinckier et al., 2007). One finding of the later study that is not consistent with our modeling work is that the hierarchical organization observed experimentally involves a succession of detectors of increasingly large letter combinations. We speculate that the full connectivity between any two layers in our models may be the reason why the backpropagation algorithm consistently finds a way to solve the task using letter-based schemes, and that this option would disappear if one was to introduce sparse, local connectivity and spatial considerations in these models. It would be interesting to carry out further modeling work with such local models to investigate whether they exhibit the letter combination detectors that have been theoretically anticipated and experimentally supported by several groups (e.g., Binder, Medler, Westbury, Liebenthal, & Buchanan, 2006; Tydgat & Grainger, 2009; Vinckier, Qiao, Pallier, Dehaene, & Cohen, 2011).

### *5.4. Word-centered orthographic representations*

In our simulations, two-deck models with their explicit level of word-centered letters better matched human ability at word-nonword discrimination. This suggests that such word-centered letter representations, or some variant thereof, might well be an indispensable ingredient of skilled orthographic processing in humans. These results further suggest, as mentioned above, that indeed the “hard problem” in orthographic processing might well be understanding how such word-centered sublexical orthographic representations are learned from location-specific visuo-orthographic input representations.

One of the objectives of modeling skilled reading is to understand how abstract semantic representations are built upon simpler representations all the way down to contrasts on the retina which are, by necessity, location-specific (retinotopic). Representations should emerge without any explicit external intervention, presumably using a combination of learning and constraints (structural, connectivity, etc. designed to mimic genetic and other biological constraints). While this goal is ambitious and complex, progress can be made, and has actually been made, using simplifications that involve explicitly imposing certain representations at some levels. As mentioned, models of the triangle tradition use such localist, word-centered letters as inputs. More generally, models of orthographic processing typically assume that letter detectors exist before the inputs of the model, impose localist representations for lexical units at the outputs, and impose representations that use some form of word-centered sublexical orthographic representation (spatial coding: Davis, 2010; overlap coding: Gomez et al., 2008; open bigrams: Grainger, 2008; Seriol: Whitney, 2001), while abstracting away from retinal location.

Having to explicitly impose the level of word-centered sublexical orthographic representations is presently a limitation of our model,  as well as of other computational models (e.g., Shillcock & Monaghan, 2001, and models of the triangle tradition and different types of dual-route model). Future work will explore how models can build or learn all representations by themselves. Models that combine unsupervised and supervised learning appear especially attractive (e.g., contrastive backprogation: Hinton, Osindero, Welling, & Teh, 2010). Despite this limitation, we found that when imposing the constraint of word-centered letters, the pattern of results is more consistent with expert human readers. This is consistent with the successes of the triangle-based models at cognitive modeling that are based on word-centered letters, and can be interpreted as a prediction of our model for a level of representation to be found in the brain. More generally, we have shown that neural networks can learn location-invariance (something necessary to explain how skilled readers cope with variability in precise location of eye fixations on words), either when directly learning lexical representations or when learning word-centered letters, something that models of the triangle tradition have sidestepped by taking word-centered letters as inputs.

Becoming a skilled reader clearly involves a long period of learning in children. Our model shows how connectionist models can take location-specific letters and learn to map them on to lexical units, i.e., perform word recognition. While the mapping is learned in our model for simplicity, the same functional processing (including the appropriate representations) could also be achieved with additionally imposing system-wide constraints (e.g., genetic). From a neurological perspective, skilled readers get location-specific contrasts on their retinas as inputs. Future work is necessary to complete lower-level processing, namely how retinal contrasts can be mapped onto word-centered sublexical orthographic representations.

Finally, one reason for why the zero-deck and one-deck networks performed less well than the two-deck network, might be that they are trying to do too much with too little hierarchical structure. This therefore points to hierarchical models going beyond standard 3-layered (input, hidden, output) structure as one promising avenue for the future modeling of visual word recognition.

### *5.6. Future directions*

As future directions for this line of research, models could implement receptive fields. This would make them more biologically plausible. The current fully-connected networks (e.g., every hidden unit is connected to every input unit) implement a letter-based overlap coding scheme, with no evidence for coding of open bigrams (Hannagan et al., 2011). Structural constraints that have been hypothesized independently by several researchers (e.g., Whitney, 2001; Grainger, Granier, et al., 2006) may be essential for enforcing the coding of combinations of letters such as bigrams.

Another known plausible biological constraint absent from the present models is stochasticity. In future models, noise could be added at various places in the networks (e.g., inputs, connection weights, transfer functions).

Furthermore, networks could be trained with a more realistic regime based on word frequencies, rather than a uniform word frequency (see Dufau et al., 2010, for an application of this kind of training regime with self-organizing maps, and Glotin et al., 2010, for an application with ART (Adaptive Resonance Theory) networks). The training regime could also reflect a realistic distribution of fixation positions, rather than using a uniform distribution of fixations, and words of different lengths. Ultimately, the goal would be to train models with a lexicon reflecting a realistic number of words known by skilled human readers.

Finally, the best of the three models proposed, the two-deck topology, supposes an explicit level of representation for word-centered letters, rather than direct learning of lexical representations from location-specific letters. The model thus predicts that humans develop an explicit level of representation for word-centered letters as they learn to read. Future experimental work could attempt to directly test this prediction.

## 6. Acknowledgements

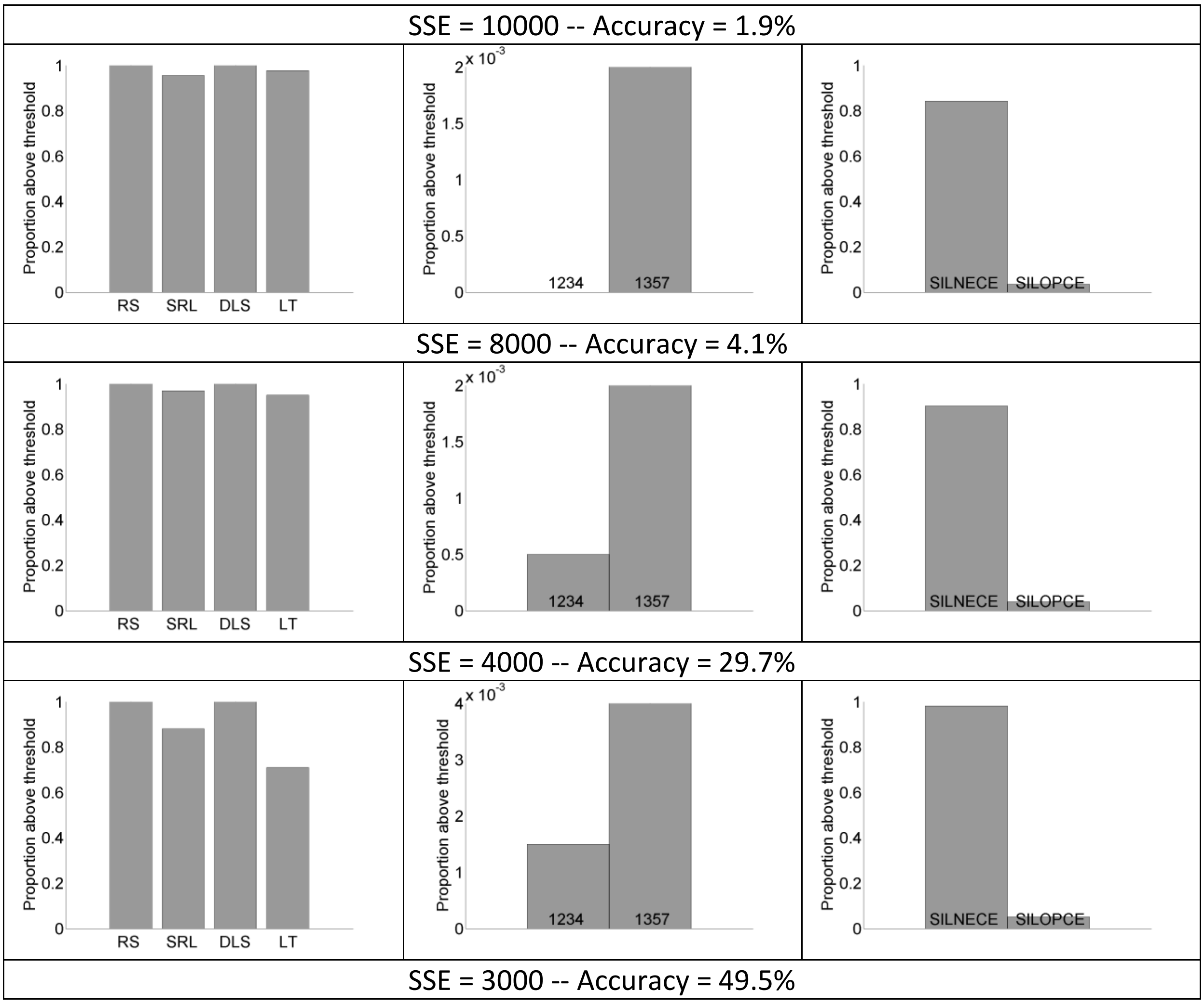
We are grateful to Myriam Chanceaux for her help running some of the simulations. This work was funded by a post-doctoral fellowship to FD from the Natural Sciences and Engineering Research Council of Canada, and a European Research Council (ERC) advanced grant 230313 to JG.

## 7. Appendix - Empirical study of robustness of results

To verify the robustness of results, we varied depth of training and number of hidden units.

### *7.1. Appendix 1: Depth of training*

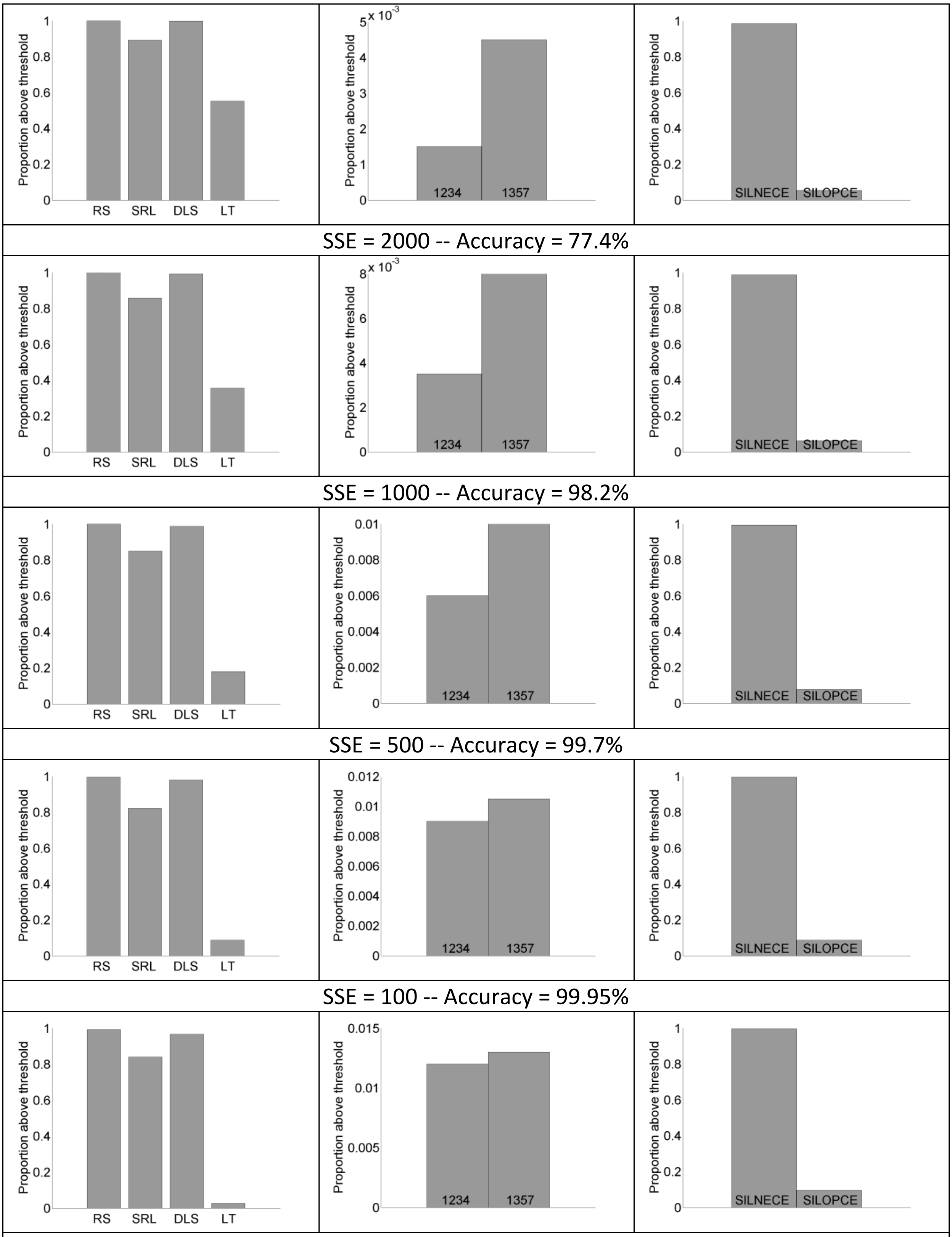
In the simulations reported in the main text, we used a target SSE value of 50 because it lead to sufficient training to effectively reach 100% accuracy on the criterion given in section 2.4. Here, we train a one-deck network of 119 hidden units, simulate early stopping by reporting performance at SSE levels ranging from 10000 down to 50 (yielding accuracies ranging from 1% to 100%). Results are presented in Figure 11. While the effects naturally get larger with increased training, the patterns of results are consistent across depths of training, namely the pattern of discrimination (RS > DLS > SRL > LT), of relative-position priming (contiguous letters 1234 < non-contiguous letters 1357), and of transposed-letter priming (central letters from the same word SILNECE > central letters from a different word SILOPCE).


SSE = 10000 -- Accuracy = 1.9%

SSE = 8000 -- Accuracy = 4.1%

SSE = 4000 -- Accuracy = 29.7%

SSE = 3000 -- Accuracy = 49.5%



SSE = 2000 -- Accuracy = 77.4%

SSE = 1000 -- Accuracy = 98.2%

SSE = 500 -- Accuracy = 99.7%

SSE = 100 -- Accuracy = 99.95%

SSE = 50 -- Accuracy = 100%

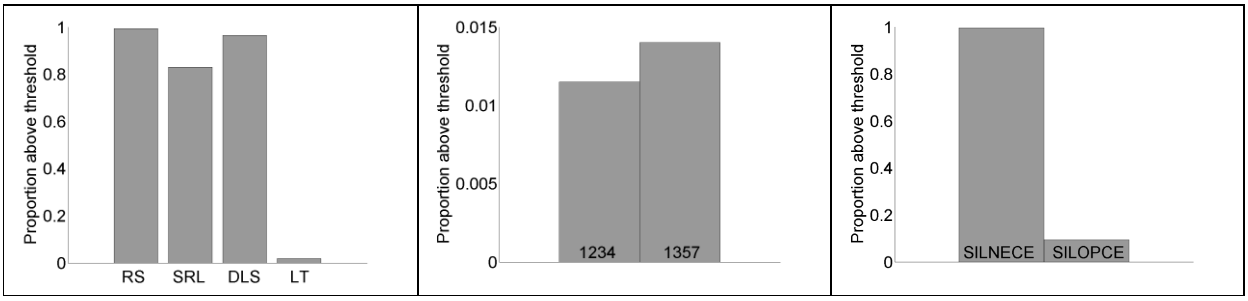


*Figure 11 - Patterns of results at different depths of training. Results are reported in three columns: Left: word-nonword discrimination; Centre: relative-position priming and Right: transposed-letter priming.*

### *7.2. Appendix 2: Larger number of hidden units*

Second, we trained a one deck network with ten times the square root of the number of training patterns. Here, that means 1190 hidden units. As can be seen in Figure 12, patterns of results are similar, except for the rejection rate of strings of single letter which is worse with the increased number of hidden units.

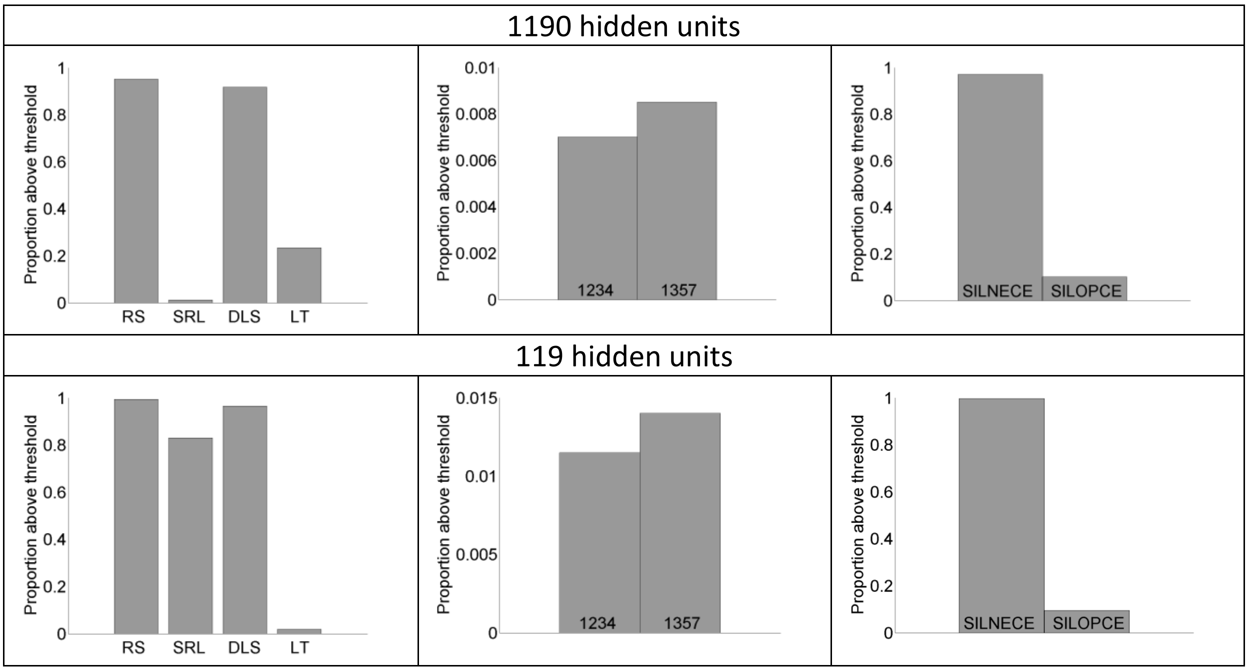


*Figure 12 - Patterns of results comparing one-deck networks containing 119 and 1190 hidden units. Results are reported in three columns:  Left: word-nonword discrimination; Centre: relative-position priming and Right: transposed-letter priming.*

### *7.3. Appendix 3: Using SSE instead of cross-entropy as a cost function*

Finally, we compare one-deck networks trained with SSE as a cost function to minimize rather than cross-entropy. As can be seen in Figure 13, patterns of results are identical.

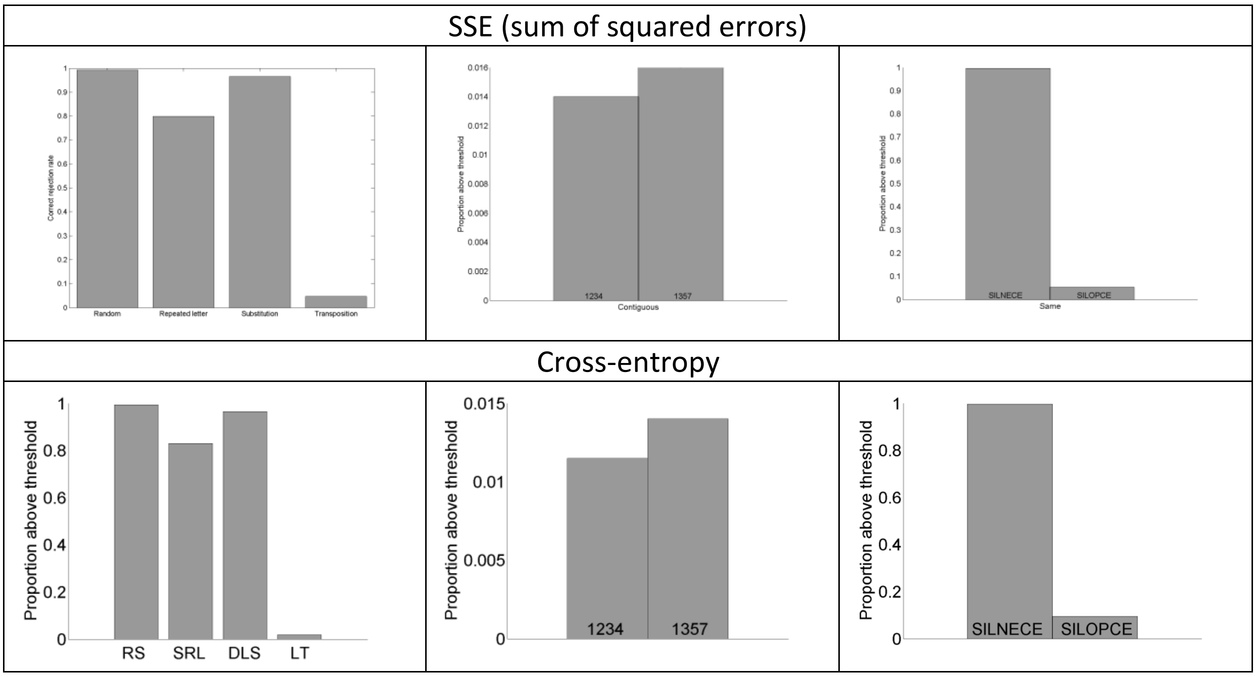


*Figure 13 - Patterns of results comparing one-deck networks trained using SSE and cross entropy as cost functions. Results are reported in three columns: Left: word-nonword discrimination; Centre: relative-position priming and Right: transposed-letter priming.*

## 8. 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.

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. (2001). DRC: A dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*, 204–256.

Dandurand, F., & Grainger, J. (2008). Compact representations of word location independence in connectionist models. In *Proceedings of the Second French Conference on Computational Neuroscience* (pp. 163–166). Marseille, France.

Dandurand, F., Grainger, J., & Dufau, S. (2010a). Learning location invariant orthographic representations for printed words. *Connection Science*, *22*(1), 25–42. doi:10.1080/09540090903085768

Dandurand, F., Hannagan, T., & Grainger, J. (2010b). Neural networks for word recognition: Is a hidden layer necessary? In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 688–693). Presented at the CogSci 2010, Austin, TX: Cognitive Science Society.

Davis, C. J. (2010). The spatial coding model of visual word identification. *Psychological Review*, *117*(3), 713.

Davis, C. J., & Lupker, S. J. (2006). Masked inhibitory priming in English: Evidence for lexical inhibition. *Journal of Experimental Psychology: Human Perception and Performance*, *32*, 668–687.

Dehaene, S., & Cohen, L. (2007). Cultural recycling of cortical maps. *Neuron*, *56*(2), 384–398.

Dehaene, S., Cohen, L., Sigman, M., & Vinckier, F. (2005). The neural code for written words: a proposal. *Trends in Cognitive Sciences*, *9*(7), 335–341.

Dehaene, S., Jobert, A., Naccache, L., Ciuciu, P., Poline, J., Le Bihan, D., & Cohen, L. (2004). Letter binding and invariant recognition of masked words: behavioral and neuroimaging evidence. *Psychological Science*, *15*, 307–313.

Dehaene, S., Le Clec'H, G., Poline, J. B., Le Bihan, D., & Cohen, L. (2002). The visual word form area: a prelexical representation of visual words in the fusiform gyrus. *Neuroreport*, *13*(3), 321–325.

Diependaele, K., Ziegler, J., & Grainger, J. (2010). Fast phonology and the bi-modal interactive activation model. *European Journal of Cognitive Psychology*, *22*(5), 764–778.

Dufau, S., Grainger, J., & Ziegler, J. C. (2012). How to say 'no' to a nonword: A leaky competing accumulator model of lexical decision. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*(4), 1117.

Dufau, S., Lété, B., Touzet, C., Glotin, H., Ziegler, J. C., & Grainger, J. (2010). A developmental perspective on visual word recognition: New evidence and a self-organising model. *European Journal of Cognitive Psychology*, *22*(5), 669–694.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, *7*(2), 195–225.

Frankish, C., & Turner, E. (2007). SIHGT and SUNOD: The role of orthography and phonology in the perception of transposed letter anagrams. *Journal of Memory and Language*, *56*, 189–211.

Glotin, H., Warnier, P., Dandurand, F., Dufau, S., Lété, B., Touzet, C., … Grainger, J. (2010). An Adaptive Resonance Theory account of the implicit learning of orthographic word forms. *Journal of Physiology, Paris*, *104*(1-2), 19–26. doi:10.1016/j.jphysparis.2009.11.003

Gomez, P., Ratcliff, R., & Perea, M. (2008). The overlap model: A model of letter position coding. *Psychological Review*, *115*, 577–601.

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

Grainger, J., & Ferrand, L. (1996). Masked orthographic and phonological priming in visual word recognition and naming: Cross-task comparisons. *Journal of memory and language*, *35*(5), 623–647.

Grainger, J., Granier, J. P., Farioli, F., Van Assche, E., & Van Heuven, W. J. B. (2006). Letter position information and printed word perception: The relative-position priming constraint. *Journal of Experimental Psychology: Human Perception and Performance*, *32*(4), 865–884.

Grainger, J., & Holcomb, P. J. (2009). Watching the word go by: On the time-course of component processes in visual word recognition. *Language and Linguistics Compass*, *3*(1), 128–156.

Grainger, J., Kiyonaga, K., & Holcomb, P. J. (2006). The time course of orthographic and phonological code activation. *Psychological Science*, *17*(12), 1021–1026. doi:10.1111/j.1467-9280.2006.01821.x

Grainger, J., Lété, B., Bertrand, D., Dufau, S., & Ziegler, J. C. (2012). (in press) Evidence for multiple routes in learning to read. *Cognition*.

Grainger, J., Tydgat, I., & Isselé, J. (2010). Crowding affects letters and symbols differently. *Journal of Experimental Psychology: Human Perception and Performance*, *36*(3), 673–688.

Grainger, J., & Van Heuven, W. J. B. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *The Mental Lexicon* (pp. 1–23). New York: Nova Science Publishers.

Hannagan, T., Dandurand, F., & Grainger, J. (2011). Broken symmetries in a location invariant word recognition network. *Neural Computation*, *23*(1), 251–283.

Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: cooperative division of labor between visual and phonological processes. *Psychological review*, *111*(3), 662.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*(1-3), 185–234.

Hinton, G. E., Osindero, S., Welling, M., & Teh, Y. W. (2010). Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, *30*(4), 725–731.

Marchman, V. A. (1993). Constraints on plasticity in a connectionist model of the English past tense. *Journal of Cognitive Neuroscience*, *5*(2), 215–234.

McClelland, J. L., & Rumelhart, D. E. (1988). *Explorations in parallel distributed processing*. Boston, MA: MIT Press.

New, B., Pallier, C., Brysbaert, M., & Ferrand, L. (2004). Lexique 2: A new French lexical database. *Behavior Research Methods, Instruments, and Computers*, *36*(3), 516–524.

Perea, M., & Lupker, S. J. (2004). Can CANISO activate CASINO? Transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory & Language*, *51*(2), 231–246.

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

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested incremental modeling in the development of computational theories: The CDP+ model of reading aloud. *Psychological Review*, *114*, 273–315.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, *103*, 56–115.

Rumelhart, D. E., McClelland, J. L., & PDP research group (Eds.). (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

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.

Segui, J., & Grainger, J. (1990). Priming word recognition with orthographic neighbours: Effects of relative prime-target frequency. *Journal of Experimental Psychology: Human Perception and Performance*, *16*, 65–76.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96*(4), 523–568.

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

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

Stevens, M., & Grainger, J. (2003). Letter visibility and the viewing position effect in visual word recognition. *Perception & Psychophysics*, *65*(1), 133–151.

Tydgat, I., & Grainger, J. (2009). Serial position effects in the identification of letters, digits, and symbols. *Journal of Experimental Psychology: Human Perception and Performance*, *35*(2), 480–498.

Vinckier, F., Dehaene, S., Jobert, A., Dubus, J. P., 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*(1), 143–156.

Vinckier, F., Qiao, E., Pallier, C., Dehaene, S., & Cohen, L. (2011). The impact of letter spacing on reading: A test of the bigram coding hypothesis. *Journal of Vision*, *11*(6). Retrieved from http://www.journalofvision.orgwww.journalofvision.org/content/11/6/8.short

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

## Figure captions