Psychological Review
2013, Vol. 120, No. 4, 903–916

0033-295X/13/$12.00 DOI: 10.1037/a0034195

How Do PDP Models Learn Quasiregularity?

Woojae Kim, Mark A. Pitt, and Jay I. Myung
Ohio State University

Parallel distributed processing (PDP) models have had a profound impact on the study of cognition. One domain in which they have been particularly influential is learning quasiregularity, in which mastery requires both learning regularities that capture the majority of the structure in the input plus learning exceptions that violate the regularities. How PDP models learn quasiregularity is still not well understood. Small- and large-scale analyses of a feedforward, 3-layer network were carried out to address 2 fundamental issues about network functioning: how the model can learn both regularities and exceptions without sacrificing generalizability and the nature of the hidden representation that makes this learning possible. Results show that capacity-limited learning pressures the network to form componential representations, which ensures good generalizability. Small and highly local perturbations of this representational system allow exceptions to be learned while minimally disrupting generalizability. Theoretical and methodological implications of the findings are discussed.

Keywords: PDP model, quasiregularity, network analysis, hidden representation

Supplemental materials: http://dx.doi.org/10.1037/a0034195.supp

The parallel distributed processing (PDP) approach to studying cognition has had a significant impact in cognitive science, ranging from the fields of perception to reasoning (Thomas & McClelland, 2008). The premise of the approach is that many neuron-like units interacting through inhibition and excitation can provide insight into not only the end-state of learning but also crucially the learning process itself. That is, PDP models can be used as tools to understand how mastery of an ability or skill is achieved.

A fundamental problem in learning to which PDP models have been applied is learning *quasiregularity*; that is, learning certain regularities but also learning violations of those regularities. Quasiregularity is ubiquitous in how humans structure the environment. It is required to form categories of objects, learn concepts, read words, and solve problems. To highlight one example, quasiregularity is especially prevalent in language, and one focus of research in language processing is to understand how humans learn *regular* items, which can be grouped together and thus described by rules (e.g., verbs that become past tense by adding *ed*), and at the same time learn *exceptions*, which violate such rules and require a context-sensitive response (e.g., the past tense of *find* being *found*, not *finded*). Rumelhart and McClelland (1987) showed that a PDP model can learn both the regularities and the exceptions.

Despite the ability of PDP models to learn quasiregularity, how they do so is poorly understood. In particular, little is known about the representations the model forms that enable it to distinguish regulars from exceptions. More specifically, what is the nature of the transformation between input and output? The complexity of PDP models has made it difficult to answer such questions, yet answers are vital. A model is more useful when we know how and why it can generate a certain data pattern than only that it can do so. Only in the former case can the behavior be linked to specific properties of the learning mechanism, and thereby provide a more complete and ultimately satisfying account of data.

The purpose of the current study was to explain, microscopically and macroscopically, how PDP models learn quasiregularity. To do so, we addressed two issues that have proven particularly challenging: clarifying the distinction between regulars and exceptions in the hidden representations of the model and explaining how the model can simultaneously generalize well (i.e., rule-like behavior) while also learn exceptions to the regularities. We chose the reading model of Plaut, McClelland, Seidenberg, and Patterson (1996) as a testbed in which to address these questions, in part because these authors performed model analyses themselves to address these issues, so it served as a logical starting point. We begin with a review of their model analyses, which set the stage for the present investigation.

The ability to pronounce English words correctly requires learning the regularity in the correspondence between spelling and sound. The regularity is evident in the fact that a reader can pronounce not only regular words (e.g., *hint, gave*) but also similarly spelled nonwords (e.g., *kint, mave*) that they have never encountered. However, mastery of standard spelling-sound correspondences is not sufficient. The reader also must learn how to pronounce many exception words (e.g., *pint*, *have*) that violate such regular associations between spelling and sound, which cause quasiregularity. Furthermore, the phenomenon of quasiregularity

This article was published Online First September 9, 2013.

Woojae Kim, Mark A. Pitt, and Jay I. Myung, Department of Psychology, Ohio State University.

This research is supported by National Institutes of Health Grant R01-MH093838 to Jay I. Myung and Mark A. Pitt. We thank Maximiliano Montenegro for helping with code optimization and suggesting reduced-scale network analysis.

Correspondence concerning this article should be addressed to Woojae Kim, Department of Psychology, Ohio State University, 1835 Neil Avenue, Columbus, OH 43210. E-mail: kim.1124@osu.edu

in English goes beyond a simple dichotomy of regulars and exceptions. There also exist subregularities in which a group of exception words (e.g., *brow, how, now*) share a pronunciation, mostly of the vowel, that differs from a larger set of words with the same spelling (e.g., *glow, know, low, show*).

PDP models of reading were introduced as an alternative to dual-route theories that maintained that regular and exception words are processed by independent systems (e.g., Coltheart, 1978). The Plaut et al. (1996) model is the second in a series of PDP models in which exceptions and regulars are processed in a single system. By changing the input representation (set of graphemes), it overcame limitations in nonword pronunciation that were found in its predecessor (Seidenberg & McClelland, 1989) and achieved a level of performance closer to that of humans in reading words and nonwords. Setting aside the empirical validity of the model, a major contribution of Plaut et al. was to demonstrate that a single computational system, rather than two independent systems, could learn quasiregularity.[1]

To gain insight into how their model learned quasiregularity, Plaut et al. (1996) took their investigation one step further and analyzed model performance. Analyses focused on how the model simultaneously learns regular and exception words (a description of the model can be found in Supplement A in the online supplemental materials). They demonstrated that a three-layer PDP network exhibits different degrees of context sensitivity when pronouncing exception versus regular words (Plaut et al., 1996, pp. 87–90). Their analyses measured the *componentiality* of the network's representation of an input word on both the output and the hidden representation levels. To understand the concept, suppose that the network is fed with the input string *print*. This monosyllabic word consists of three parts, or three orthographic clusters: the onset, *pr*, the vowel, *i*, and the coda, *nt*. *Input-to-output componentiality* is achieved if the network's pronunciation of the vowel, /i/, for example, is independent of its response to the onset, /pr/, when the same phonological vowel, /i/, is output regardless of the identity of the orthographic onset (e.g., whether *pr* is replaced by the onset of another regular word, e.g., *m* in *mint,* or even by the onset of a nonword, e.g., *k* in *kint*). Componentiality represents a basic skill required to pronounce nonwords and is a strong indication that the model has learned regularity. *Input-to-hidden componentiality* is defined in a conceptually similar manner. The hidden representation of an input word (e.g., *print*) is componential if the contribution of each orthographic cluster (e.g., *pr*) to the representation remains constant whether it is assessed in context (e.g., *int* → *print*; i.e., difference in hidden representation when the input changes from *int* to *print*) or in isolation (e.g., null → *pr*).

The results of Plaut et al.'s (1996) analyses showed that the network's input-to-output mapping is highly componential with regular words, indicating that the model learned regularity but is clearly noncomponential with exception words, meaning that the pronunciation of part of the word is ambiguous because it depends on context (pp. 87–88). However, when an analysis was performed on the hidden representations of the network, the representations of both regular and exception words were shown to be highly componential, with those of exception words being only slightly less componential than those of regular words (pp. 89–90). This result was in marked contrast to the preceding input-to-output analysis that showed distinctly different network responses depending on the word type. These seemingly incongruent findings across analyses led Plaut et al. to propose that the hidden representations preserve the orthographic structure of words at multiple levels, from individual graphemes to combined clusters, in order to respond differently to regulars and exceptions (pp. 90–91).

Since Plaut et al. (1996), little progress has been made toward understanding how PDP models can perform a task that seems to require distinct, conflicting functionalities. Instead, research efforts have concentrated more on adapting the model and exploring its behavioral ramifications for finer empirical details. This was done either by making changes to its existing components (e.g., recurrent structure incorporated into the output layer by Harm & Seidenberg, 1999), or by introducing an additional processing route (e.g., a purely componential route added by Zorzi, Houghton, & Butterworth, 1998; a semantic route added by Harm & Seidenberg, 2004). Successful attempts in this direction, however, did not provide a better understanding of multilayer PDP architectures. In fact, they have made it even more desirable because an adequate understanding of the functioning of a model should precede further refinements of it in order for such development to have maximal theoretical impact.

Present Study

The goal of the present study was to provide a clear understanding of how a PDP model learns quasiregular mappings. Our approach was first to study the model on a small scale to understand the details of the learning mechanism that are involved in the model's ability to generalize and learn exceptions. This was followed by simulations of the Plaut et al. (1996) model to assess the ramifications of the findings for a large-scale network.

Two issues were the focus of both analyses. First, the phenomenon of representational remapping (via the hidden units) is not well understood. As just discussed, Plaut et al. (1996) showed that on the output level, the network responds in distinctly different manners to regular and exception words, being highly componential for regulars but not for exceptions. However, when the hidden representations of the network are probed, a clear-cut distinction between the two types of words is not found in the network's internal states. These two outcomes would seem to be at odds with one another. How can representations that vary in componentiality be so similar? The problem of understanding the representational distinction between the two types of words has been particularly challenging (Bakker, 1995; Bullinaria, 1997).

The second issue we addressed is how two seemingly conflicting functionalities, the abilities to generalize well while also reading exceptions, can coexist in a single PDP system. One would think that learning exceptions should incur a cost in the network's ability to generalize. This question is closely connected with the preceding question and can be rephrased from the perspective of representational remapping as follows: How is it possible that noncomponential representations, which must exist to handle exceptions, can coexist with componential representations without adversely affecting reading? Should there not be interference be-

[1] In the interest of clarity, we emphasize that the focus of this study is on understanding how PDP models learn quasiregularity. Our use of the Plaut et al. (1996) model is not meant to imply our endorsement of it or any other model, or to speak to the validity of a single-route versus dual-route structure.

tween competing representations? Intuitively, a fully connected PDP network trained on a dense corpus of 3,000 words would preclude such a thing as absolutely no interference among the representations. We may therefore sharpen our question and ask: To what extent does learning exceptions interfere with the componential characteristic of internal representations, limiting generalizability? A difficulty in answering this question is that we lack an adequate characterization of how learning inconsistent input-to-output associations (i.e., exceptions) affects the formation of consistent input-to-output (i.e., regular, fully componential) representations. Much of the power of PDP models resides in this ability, yet understanding it has been difficult. By properly characterizing how the network remaps input patterns through its hidden layer and showing how such structural re-representation relates to its output performance (i.e., reading), this central aspect of the network behavior can be demystified.

Understanding Network Learning Through Statistical Modeling

Insight into model behavior can be gained by reframing the learning problem the network is trying to solve as one of statistical modeling, which provides an informative perspective on what is being achieved in quasiregularity learning. From the standpoint of statistical modeling, reading is a form of a multicategory classification task, and nonword reading requires generalizing from old, learned items (e.g., words) to classifying new, unseen items (e.g., nonwords). Good generalizability requires optimizing model design. Although many design choices (e.g., how to encode input representations) can affect network performance, here we focus on one of them, *capacity-limited learning*.

Capacity-Limited Learning Promotes Generalizability

The *capacity* of a model to learn, also called the *expressive* or *descriptive power* of a model, is an important concept in computational learning theory (Anthony & Biggs, 1997; Hastie, Tibshirani, & Friedman, 2009; Kearns & Vazirani, 1994). In the context of a classification task, the capacity of a model can be defined as the *maximum* number of items that it can learn to classify without errors no matter which categories the items would have been assigned to (e.g., Vapnik & Chervonenkis, 1971). The concept and measure of a computational model's capacity are central to the process of model building and evaluation. In computational learning theory, there is ample justification for constraining the expressive power of a model in order to maximize its generalizability (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1989; Valiant, 1984; Vapnik, 1992, 1998). In general, as a model's capacity grows (e.g., with more free parameters or wider ranges of parameter values), the ability to fit (e.g., pronounce) the current sample of data on which the model is trained also increases. However, its ability to fit new, unseen data (i.e., generalize) does not keep improving as capacity increases. A model's generalization performance is usually maximized when a certain, optimal level of capacity is reached, beyond which generalizability starts to deteriorate, because the model begins to overlearn the training words. This suboptimal situation is often referred to as overfitting.

Suppression of a model's full capacity is indispensable in PDP modeling. With no constraints whatsoever, it has been shown that the full descriptive power of multilayer PDP networks is enormous and can easily cause the model to overfit (i.e., overlearn), only to result in poor generalizability (Haykin, 1999; Hornik, Stinchcombe, & White, 1989). Following standard practice, Plaut et al. (1996) also used a well-known training scheme to prevent the network from overfitting due to magnified capacity, which would have resulted in poor nonword reading performance. They used a technique called *weight decay* (weights are given a tendency to decay toward zero), the idea behind which is to limit a network's capacity by keeping its connection weights from growing beyond a certain extent.[2] PDP modelers in psychology are aware of the effectiveness of constraining capacity for enhancing a model's generalization performance, yet the implications for learning quasiregularity have not been thoroughly investigated. As we show, a consideration of capacity is central to understanding why models can both generalize and learn exceptions.

Capacity-Limited Learning in Context

What are the implications of capacity-limited learning for a PDP model? We explore this question by taking a closer look at the consequences of restricting weight growth for generalization and exception learning. Consider how connection weights affect the input-to-output transformation in each layer of the network. The state of each recipient unit in a layer is determined first by computing a linear combination of all units' states in the preceding layer with corresponding weights, and then putting this value into an activation function. Symbolically, the activity, s_j, of an output unit j, given the activities of all inputs, s_i's, in the preceding layer is computed by

$$s_j = \sigma\left(\sum_i s_i w_{ij}\right).$$

A typical form of the activation function $\sigma(\cdot)$ is an *S*-shaped, nonlinear function, an example of which is shown in Figure 1. Now suppose that most of the connection weights are close to zero. Then, the linear sum in the above equation should also be close to zero for most inputs. In this state, the network's input–output mappings can no longer be viewed as nonlinear since the activation function mimics a linear function as its inputs approach zero, as shown by the thick solid part of the function. That is, for most inputs, the mapping to outputs becomes essentially a *linear* transformation.

A model that is reduced completely to a linear map can still read most regular words, and while making errors on many exception words, it retains the ability to read nonwords, thus generalizing well (Zorzi et al., 1998). To further understand why a linear map behaves in this way, let us consider its general property. An

[2] To similar effect, Plaut et al. (1996) also used *early stopping* (i.e., the training process is forced to stop early even though the network could further reduce errors). For example, instead of using weight decay, their Simulations 2 and 3 stopped training early (at the 1,300th and 1,900th epoch) along with the use of very slow learning (specifically, the ratio of learning rate parameters to the summed frequency of words was much lower than that used in Simulation 1). Yet another way to limit a multilayer PDP network's capacity is to reduce the number of hidden units. As will be discussed later, however, when improving the generalizability of a reading network, the magnitude of weights, rather than their absolute number per se, is the more relevant factor to consider.

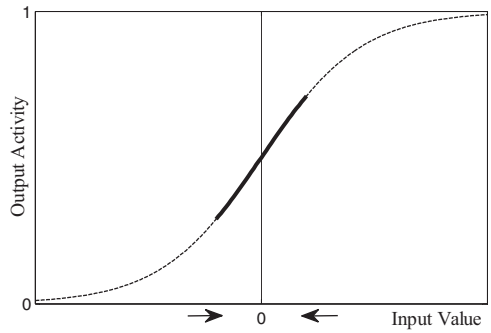

Figure 1. Sigmoidal activation function of processing units. The solid portion of the function and the arrows on the *x*-axis make the point that as the input approaches zero, the activation function becomes linear.

essential, defining characteristic of a linear map that transforms one vector into another is *additivity*. If a function f is a linear map, then it satisfies:

$$f(\vec{s}_1 + \vec{s}_2) = f(\vec{s}_1) + f(\vec{s}_2),$$

where $\vec{s}_1$ and $\vec{s}_2$ are vectors in the input space and $f(\cdot)$s are vectors in the output space. In a reading network, in which a vector may represent part of a word, this property essentially means that the contribution of part of an input word's representation to the word's *transformed* representation is independent of other parts of the input word. For example, suppose that a word that starts with the letter *p* (e.g., *pin*, *pint*, *pain*) is represented by the sum of two vectors, $\vec{s}_1$ and $\vec{s}_2$, where each vector represents an orthographic part of the word (e.g., $\vec{s}_1 = p$ and $\vec{s}_2 = in$). Then, if the network's orthographic-to-hidden connection is a linear map, it follows that the contribution of the orthographic cluster p ($= \vec{s}_1$) to the word's hidden representation ($= f(\vec{s}_1 + \vec{s}_2)$) can always be identified as the same vector ($= f(\vec{s}_1)$) no matter what the context is (e.g., $\vec{s}_2 = in$, *int*, or *ain*), because of the additivity ($f(\vec{s}_1 + \vec{s}_2) = f(\vec{s}_1) + f(\vec{s}_2)$). Additivity in a linear map is therefore synonymous with componentiality (see Supplement C for a technical treatment of their relationship).

Note that the concept of componentiality that we use here is defined with respect to a transformation across two representation spaces (i.e., input-to-hidden or hidden-to-output). In a reading network, a linear map is a special case that may be regarded as a *componentiality-preserving* transformation of representations. That is, if a word's representation in one layer is componential in relation to the preceding layer (an orthographic representation is by design componential within itself or under the identity transformation), then a linearly transformed representation of it in the next layer is guaranteed to be componential. A linear map is therefore a sufficient condition for componentiality in mapping orthographic inputs onto their pronunciations, a necessity for reading nonwords (i.e., generalization), but a constraint that must be circumvented to read exceptions.[3]

Analysis of a Small-Scale Network

Although capacity-limited learning is an essential ingredient of good network generalizability, it alone is insufficient to explain network behavior. What is lacking is an understanding of how exceptions are also learned within a componentiality-preserving system. Returning to the questions raised earlier, how are they represented in the hidden structure of the network, and why does their encoding minimally impact generalization in the network? We addressed these questions, as well as confirmed the importance of capacity-limited learning, by analyzing a small-scale reading network, the simplicity of which was intended to make changes in network representation and output (production) tractable as we varied the inclusion of an exception word in the training set.

Method. The network had a three-layer, feedforward architecture. In the input layer, three units represented three graphemes, B, I and N. An input (e.g., IN) was mapped onto three hidden units in the next layer and then converted to three phoneme units, /b/, /i/ and /n/, in the output layer (additional details of the model can be found in Supplement B). To see how the network develops its hidden representations, we trained it on four artificial words with corresponding targets. In the regular training set, {B, I, N, BIN} should be pronounced as {/b/, /i/, /n/, /bin/}, respectively (Following Plaut et al., 1996, we adhered to the phonological spelling conventions in their Table 2). These mappings reflected regular spelling-sound associations that conformed to componentiality perfectly (i.e., B→/b/, I→/i/, and N→/n/). In the exception training set, the target pronunciation of BIN was set to /bi/ rather than /bin/. That is, the word BIN was made an exception (or ambiguous) word since the grapheme N should be silent when B and I are present as a context, but active when presented with no context. Note that in these two training conditions, the other possible combinations of the three graphemes, namely, BI, IN and BN, on which the network was not trained, are nonwords. The network training involved precisely the same algorithm as used for the full-scale, feedforward network in Plaut et al. (1996): The standard back propagation learning algorithm with the cross-entropy error function and connection-specific, adaptive learning rates (for details, see Supplement A). The training imposed relatively strong decay of weights such that the network does not learn to pronounce all four training words until it reaches the 100th pass of trainings set presentation.

Results. After training, the network correctly pronounced all four words in both training sets. Let us first focus on the case in which the network was trained on all regular words. The network's hidden representations of all words and nonwords are shown in the top, leftmost panel of Figure 2. Representations of training words are marked with gray-filled circles and those of nonwords with empty circles. Dashed lines connecting representations show hidden-unit activities that are interpolated between two input representations (e.g., for the null input (0, 0, 0) and the word N (0, 0, 1), the hidden representations of all linear interpolations between them, e.g., (0, 0, 0.4), are on the line). First, note that the componential, or additive, structure of input representations are mapped onto the hidden-unit space with virtually no structural changes. For

[3] What is required for complete componentiality in input-to-output mappings is in fact a linear map only between input and hidden representations. The hidden-to-output mapping simply functions as a linear discriminant regardless of whether the activation function at the output is linear or nonlinear (Duda, Hart, & Stork, 2001; Haykin, 1999). The reason for this is that, technically, the decision boundaries induced by the decoding rules for correct pronunciations are always linear and thus not affected by the shape of the activation function.

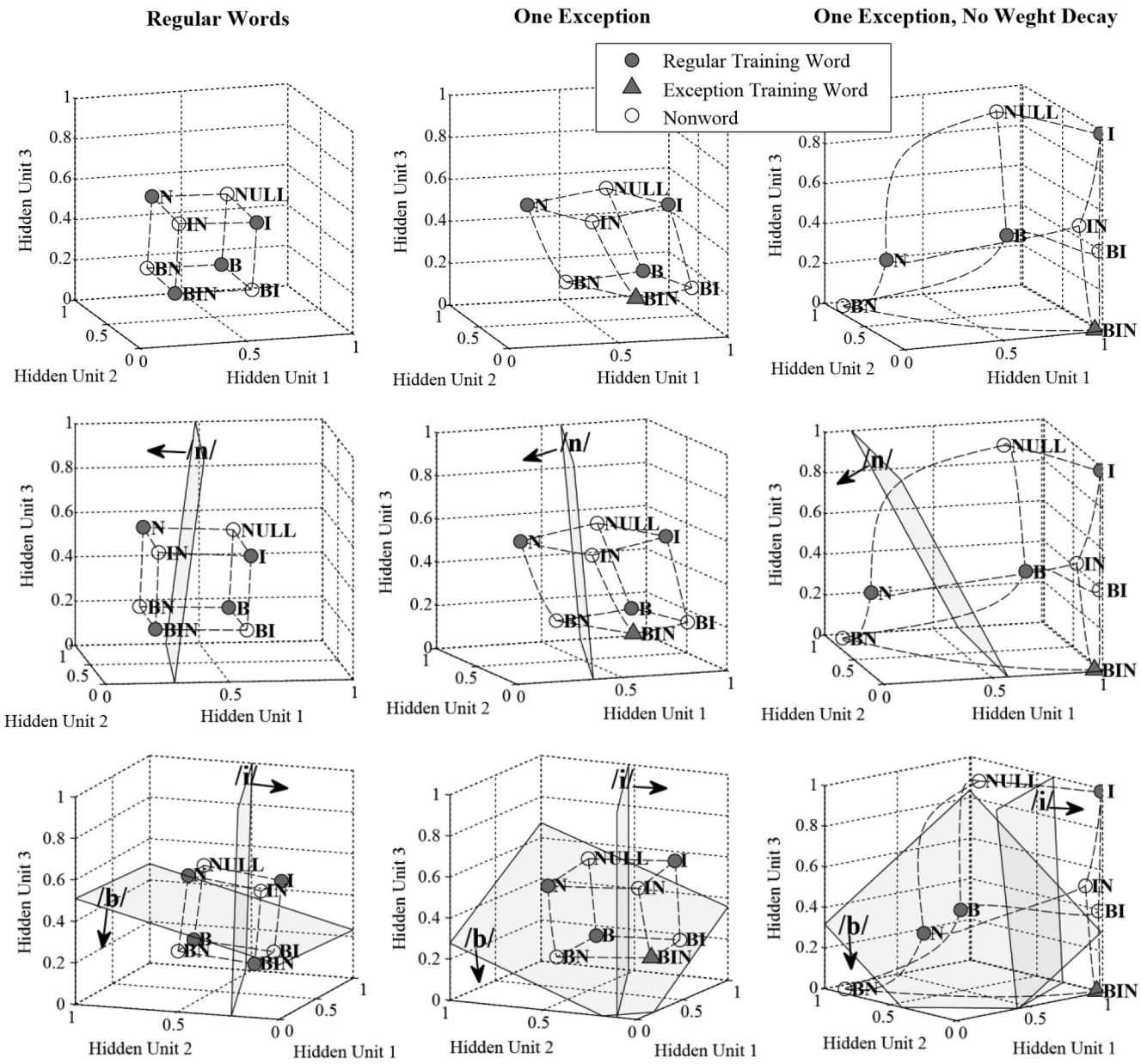

Figure 2. Three-dimensional depictions of the internal representations in a small-scale network after learning to read four words. The graphs in the left column show representations after learning a training set with only regular words. In the graphs in the center column, the training set contained one exception word. In the right column, weight decay was turned off while learning a training set with one exception word. The shaded regions added in the second and third rows are planes that identify the locations of the decision boundaries for pronouncing the three phonemes, with the arrows indicating the side of the boundary that yielded the phoneme's production (the phoneme is silent on the other side of the boundary). The third row shows the graphs from a different angle to provide another view of how the boundaries bisect the representations.

example, the hidden representation of the word BIN is the composition of three elemental representations corresponding to B, I, and N (i.e., sum of the three vectors formed by taking the null representation as the origin). The same property also holds for the representations of nonwords (i.e., BI, IN, and BN). A geometric interpretation of this componentiality is that the eight hidden representations form a parallelepiped, all sides of which are parallelograms. This well-formed structure also indicates that the input-to-hidden transformation of all inputs occurred within the linear range of the hidden-unit activation function (i.e., thick line in Figure 1).

The implications of this representational structure become clear when we examine how the network learned to associate the representations with outputs (i.e., phonemes). The decoding rule for

each phoneme-unit activity translates into a linear decision boundary, or a hyper-plane, in the hidden-unit space. The decision boundary for each of the three phonemes is shown in the leftmost panel in the middle and bottom rows (shaded areas) and labeled with the corresponding phoneme. Arrows next to the labels denote the side of each boundary on which representations are mapped onto the active phoneme. Note that the boundaries are positioned in such a way that all training words (filled circles) are pronounced correctly. Moreover, the orientations of the boundaries also ensures the network pronounces the three nonwords (empty circles) properly (i.e., BI→/bi/, IN→/in/, and BN→/bn/). Another thing to notice in these graphs is that the boundaries and representations align in a fairly linear (perpendicular) fashion, which leads to generalization. Remember that training merely dictates that the boundaries should classify the training words (filled circles) correctly, but due to the very nature of linearity imposed on both representation and decision, they are highly likely to classify nonwords (empty circles) in an expected manner. This simple visualization demonstrates graphically how regularity is learned and represented by the network in a way that facilitates generalization.

How are representations and decision boundaries altered by the introduction of an exception word? The results from the condition in which the network was trained with inconsistent spelling-sound correspondences in the training set (BIN→/bi/ while N→/n/) are shown in the middle column of Figure 2. The data in the top graph show that the representational structure is not quite as componential as that found with regular words. It has deformed slightly from a perfect parallelepiped. In particular, compared to other (non) words that have N in them (i.e., N, IN, BN), the representation of BIN has shrunk toward the plane on which words do not contain N (i.e., B, I, BI, NULL). The decision boundaries, shown in the lower graphs, are again positioned to ensure that all training words (filled circles) are pronounced correctly. Unlike when the training set consisted of only regular words (left column), however, in order to pronounce the word BIN correctly, the network tilted the /n/ phoneme boundary so that it bisects the parallelepiped differently. With the exception word, the boundary now separates N from the other three words. Without the exception, N and BIN are separated from the other two words.

There are multiple consequences of these adjustments made by the network. First, by reorienting the decision boundary, the network's input-to-output behavior is no longer fully componential. The pronunciation of BIN, particularly the coda N, is correct only if the activity of the input unit I is above a particular level, regardless of the activity of N. This is shown by the boundary for /n/, in the middle graph (Row 2 and Column 2), intersecting the line between BN and BIN that connects all levels of the vowel I contribution to the representation of BIN (e.g., the line contains the representation of BIN with I partially active to 0.3). That the line is crossed by the /n/ boundary translates that a nonzero contribution of the vowel I is needed to activate the phoneme /n/. That is, the correct phonological response of a word's cluster (i.e., silent N in BIN) depends on the activity of another orthographic cluster in context (i.e., vowel I). This example showcases how the network can exhibit clearly noncomponential output behavior with an input containing an exception word, yet its hidden representations are close to being componential.

Second, the slight reorientation of the decision bound and the slight deformation of the representations work together in a way that minimally impacts the network's responses to other words and nonwords.[4] The consequence of this cooperation is that the likelihood of the /n/ boundary being aligned linearly with the componential structure of all other representations is still kept high, ensuring that the network's generalizability (i.e., nonword reading performance) is well preserved. In fact, the results of this particular simulation include the case in which the network pronounced the nonword IN differently, as /i/ instead of /in/, under the influence of BIN→/bi/. Although relative to the regular network this is considered an error in pronunciation, it is actually an example of generalization occurring with exception words because on that side of the decision boundary, N should be silent. Even so, the representation of IN contains a correct componential response (i.e., I→/i/) and can be explained by the interplay of the representational structure and the linear decision bound. Because the network's internal representation system remains nearly componential, the likelihood of componential output responses is kept high when the linear boundary, induced from training words, is applied.

Recall that we attributed the network's tendency to approach a linear map to the capacity-limited training scheme. To verify the importance of capacity-limited learning for network generalizability (and quasiregularity more generally), we ran an additional simulation with the small-scale network in which the goal of training was solely to minimize the target-output errors (weight decay was turned off). The training set with the exception word was used, and the data are in the last column of graphs in Figure 2. After training, the network pronounced the four training words correctly. However, the top graph shows that the hidden representations are now distorted significantly from a componential structure. In particular, the word BIN has deviated so much that the contribution of N in context (i.e., BI→BIN) is no longer parallel to that of N in isolation (i.e., null→N). Such a large amount of structural distortion should hinder network generalization by lowering the likelihood of decision boundaries aligning properly. An instance of this is found in the network's mispronunciation of the nonword BI, as /i/ instead of /bi/, which is shown in the bottom graph. BI is on the wrong side of the /b/ boundary. This analysis shows that capacity-limited learning and the componential representations induced by enforcing it play a key function in good network generalization.

Summary. Analysis of a small-scale network begins to answer our two questions concerning how regulars and exceptions are represented in a single network, and how generalizability is related to exception learning. The linearity constraint imposed by capacity-limited learning ensures the network forms a componential representation system, and learns exceptions in a way that least distorts componentiality. Boundaries for phoneme decision are by design linear, making it possible for them to align with the componential representations to promote generalization to nonwords. When exceptions are learned, generalizability is retained because the network accommodates exceptions by slightly altering the

[4] In fact, the representational deviation and the decision bound repositioning are not both required for this small-scale network to learn inconsistent mappings. Reorienting the decision boundary alone, for example, is sufficient. The situation with full-scale corpus learning is likely much more highly constrained.

linear system. It seems that such adjustments can occur locally, concentrating on the representations and decision boundaries that involve the exception being learned. Because the rest of the network is left largely undisturbed, adverse effects on generalization to nonwords are unlikely to be severe.

Analyses of the Plaut et al. Model

The preceding analysis provides an understanding, on a mechanistic level, of how the linearity constraint that governs both representation and decision is conducive to the network being able to generalize while learning exceptions. A full understanding of quasiregularity learning, however, requires studying representation and generalization in a full-scale network. The regular and exception words that were created for the small-scale analysis do not represent a realistic composition of regulars and exceptions in a large corpus. For a word to be regarded as an exception, regularity must be established strongly by a much larger group of regular words that share common spelling-sound correspondences. Also, the small corpus is too simple to contain the different levels of inconsistent mappings that exist in a corpus of 2,998 words. There are isolated exceptions, which do not share similar spelling-sound associations with other words, and there are neighborhoods of exception (or ambiguous) words, in which similarly spelled words are divided into two or more groups that are pronounced similarly within a group but differently between groups. For example, the body (vowel plus coda) *ead* in a word can be pronounced as /Ed/ and /ed/, with words in the former category (e.g., *bead, plead*) being less common than those in the latter category (e.g., *dead, head*).

Analysis of a full-scale network is also necessary to understand the ramifications of the learning mechanisms identified in the small-scale network. What are the consequences of the slight adjustments to representations and decision bounds when the size and complexity of the network are more realistic? Are they far-reaching, affecting the pronunciation of many nonwords, or more tightly constrained? Answers to these questions should provide a clear understanding of why exception learning is minimally detrimental to generalizability (Question 2). Analysis of a large network should also allow us to assess more fully whether, as the small-scale analysis suggests, exceptions are representationally nothing more than a slight deviation from regulars (Question 1).

The hidden representations and decision criteria that a network develops from learning a large corpus may be too complex to understand if the network is analyzed only at the state of correctly pronouncing all training words. Insight into network behavior could be gained by studying the incremental change that occurs when the network must learn a neighborhood of words additionally. We took this approach to studying the full reading network, comparing network behavior without and then with a small neighborhood of words in the training set. Our interest was in how the network adjusts its internal structure and output pronunciation depending on the words in the neighborhood (e.g., regulars or exceptions).

Method

The architecture, training corpus, input/output representations, and training algorithm of a PDP reading network that we analyzed were identical to those in Simulation 1 of Plaut et al. (1996). The model was a three-layer, feedforward network containing 105 grapheme (input) units, 100 hidden units, 61 phoneme (output) units. To ensure capacity-limited learning, the strength of weight decay was tuned to a level at which the network pronounces all training words correctly (except homographs) with the smallest possible weights. We verified that our implementation performed equivalently to that of Plaut et al. in reading words and nonwords. We also replicated their analyses exploring network componentiality (Supplement A).

Simulation of incremental learning. Given the full corpus of 2,998 English monosyllabic words, the network was first trained on all words but a chosen neighborhood of words (i.e., 2,998 − *n* words). The training procedure was then repeated with all training parameters (including initial states) held constant, except the neighborhood was now included in the corpus (2,998). To control for idiosyncratic results from using different initial states, training in each pair of conditions was repeated 100 times using initial weights randomly drawn from −0.1 to 0.1. Dependent measures were averaged over these replications.

Neighborhood definition. Neighborhoods were chosen based on the following rationale. Inconsistency in spelling-to-sound mappings is not only created by the presence of isolated exceptions (e.g., *pint* vs. *hint, mint, tint*) but also by groups of words with subregularities (e.g., *bead, knead, plead* vs. *bread, dead, head*). That is, two or more groups of words exist whose bodies are spelled identically but pronounced differently. For example, the body *ove* in a word is most often pronounced /Ov/ (e.g., *cove*, *drove*, *stove*, etc.), but there are also words in which it is pronounced /ˆv/ (e.g., *glove*, *love*, *shove*). In the current training corpus, there are four of the latter type among a total of 15 that contain the body *ove*. This orthographic neighborhood of words is more complicated because *ove* can also be pronounced /Uv/ (e.g., *move, prove*). When ambiguity in pronunciation is present, it may be said that words exhibiting the same pronunciation (i.e., subregularity) are *friends* but are *enemies* of words in the other subgroups (Jared, McRae, & Seidenberg, 1990). From this perspective, a word is regarded as an exception when there are considerably fewer friends than enemies (e.g., *move* in the above example).

We took the set of 192 words in Appendix A of Plaut et al. (1996), which contains 48 regular consistent, 48 regular inconsistent, 48 ambiguous, and 48 exception words and identified in the training corpus all orthographic neighborhoods to which they belonged. By convention, we focused on body-level consistency of words in our analysis (e.g., Treiman, Mullennix, Bijeljac-Babic, & Richmond-Welty, 1995). For example, *plead* belongs to a body neighborhood of 16 words that end in *ead*, but only five other words share the same phonological rime (*ead*-/Ed/; e.g., *bead, knead*). We use the term *neighborhood* to refer to something even more specific, the subset of orthographic neighbors that also shares the same rime (i.e., a group of friends). Using this convention, 154 distinct neighborhoods involving 1,109 words were identified.

We quantified the degree to which a neighborhood of words violates spelling-sound consistency by summing the frequency of words in a neighborhood and dividing this value by the summed frequency of all words that share the same orthographic body. A value of one indicates complete consistency (i.e., no enemies) for the group of words. There were 47 such fully consistent neighbor-

hoods out of the 154. The remaining 107 neighborhoods were distributed between .01 and .99 in the consistency measure. For the sake of analysis convenience, the 154 neighborhoods were divided into five classes according to their consistency level (CL): *CL8–10* (66 neighborhoods with the relative frequency $.80 < f_i \leq 1.0$), *CL6–8* (25 with $.60 < f_i \leq .80$), *CL4–6* (14 with $.40 < f_i \leq .60$), *CL2–4* (30 with $.20 < f_i \leq .40$), and *CL0–2* (19 with $0 < f_i \leq .20$). In terms of the dichotomy of regular versus exception words, the vast majority of words in the CL8–10 neighborhood are regulars by standard grapheme-phoneme correspondence rules, whereas words in the CL0–2 neighborhood are exceptions (all of which were isolated words). Words in the other neighborhoods fall in between these extremes.

Dependent measures. The effect that learning one of these neighborhoods has on the network's representational system (Question 1) was examined by comparing the hidden representation of all 2,998 words (and nonwords) in the network before and after inclusion of the neighborhood in the training corpus. Operationally, the change in representational structure was defined with respect to the network's proximity to a componential system (i.e., a linear map) before and after incremental learning (see Supplement C for technical details). By examining representational change in every training word, network performance could be assessed at two levels: Locally, we could learn which, if any, words were most affected as a result of having to learn the additional neighborhood. Globally, we could learn how much the network as a whole had to adjust to learn the additional neighborhood. The effect of such incremental learning on network generalizability (Question 2) was assessed by comparing the rate of nonword mispronunciation before and after learning the neighborhood (Supplement D describes how nonwords were created and how mispronunciations were defined).

Analysis 1: Characteristics of Re-representation in the Network

Analyses of the training results focused first on examining changes in the structure of hidden representations when neighborhoods varying in consistency level were added to the corpus. The network's closeness to a linear map for a particular probe word was measured by a correlation coefficient between the word's actual hidden representation and its theoretical, componential representation (roughly speaking, when a linear activation function is assumed; see Supplement C for detail). If the network were completely undisturbed by the introduction of a neighborhood into the training corpus, the difference between the two correlations across the *without-neighborhood* and *with-neighborhood* networks would be 0. Deviation from this value, in either direction, indicate how much the network moves farther from (or closer to) a linear map as a result of learning an additional neighborhood that varied in consistency. This measurement of change in linearity was made for all probe words and nonwords and was repeated for each of the 154 neighborhoods that the network had to learn additionally. Change values were then divided by the number of words in the neighborhood to normalize for differences in neighborhood size, thereby providing a measure that is independent of the overall number of neighbors. Unnormalized data yielded a qualitatively similar outcome.

Figure 3a shows the change in the linearity of hidden representations of all training words caused by the network learning an additional neighborhood. The change measure ($r_{\text{with}} - r_{\text{without}}$) is on the *y*-axis. On the *x*-axis is the orthographic similarity of all probe words to the words in the learned neighborhood, also measured by a correlation coefficient between the input representation of the probe word and those of the neighborhood words over the 105 grapheme units. Since a neighborhood could contain more than one word, the *maximum* correlation coefficient between the probe word and any of the words in the neighborhood was used as the measure of orthographic similarity. For example, when *came* is in the neighborhood and also the probe word, the similarity will be highest, placing this word at the left endpoint of the scale (i.e., $r = 1$). The probe word *mouth* shares little in common with the neighborhood to which *came* belongs (i.e., *fame, game, tame*, etc.), making its similarity measure fall near the right endpoint of the

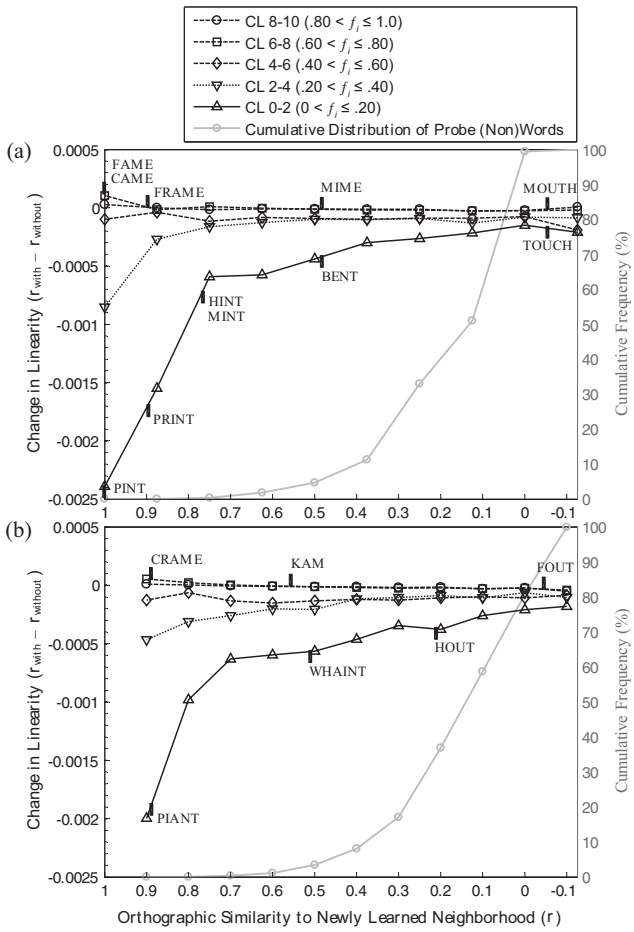

Figure 3. Degree to which the hidden representations of all words (a) and 68,104 nonwords (b) in the network deviated from a linear map after learning an additional neighborhood of words that varied in the consistency with which the orthographic body was pronounced (e.g., *here* vs. *were*). The data are arranged along the *x*-axis as a function of the orthographic similarity of each probe word (or nonword) to the neighborhood. The shaded function represents the cumulative distribution of words (or nonwords) across the bins into which the deviation measures were aggregated. Standard errors are not shown since they are small enough for differences between all compared conditions to be statistically significant with $p < .001$. CL = consistency level.

scale ($r = -.05$). For each neighborhood, all 2,985 probe words (13 of the 26 homographs were excluded to avoid redundancy) were grouped into 10 equally spaced bins on the scale, and their changes in proximity to a linear representation were averaged within each bin. These values were then averaged over all neighborhoods within each of the five consistency levels. Values greater than zero indicate increased linearity, and values less than zero, decreased linearity.

Results suggest that learning a new neighborhood of words causes deviations from linearity only when neighborhood consistency is low. When a neighborhood contained mostly consistent words (CL8–10) there was no change in the network's representational structure; this is indicated by the change measure being indistinguishable from zero across probe words at all levels of orthographic similarity. As a concrete example, consider the case in which the network newly learned the regular neighborhood to which the word *came* belongs (i.e., *came, fame, game*). Samples of mean representational changes measured in response to probe words of the same similarity are shown in the figure by thick, vertical lines with example words next to them. A similarly spelled word (e.g., *frame*) undergoes virtually no more change than words that are spelled very differently (e.g., *mouth*). Even the neighborhood words themselves (e.g., *came, fame*) barely adjust their representations after training. The CL8–10 results show that the network can learn a whole neighborhood of regular words (an average of 11.3 words) without disturbing its representation system. These regular words fit into the existing level of componential structure in the network, not requiring it to be distorted any further than what was required to learn the training set without the neighborhood.

Inspection of the function in the CL6–8 condition shows that linearity was also fully maintained after introduction of this neighborhood, the function being flat and indistinguishable from that of the CL8–10 condition. Deviation in linearity started to occur in the CL4–6 condition, yet the change was very small across all probe words, as shown by the function being constantly close to zero. Despite these neighborhoods being more inconsistent in spelling-sound correspondence, the network was able to preserve componentiality in their representation.

Only in the two lowest consistency conditions (CL0–2, CL2–4) are considerable deviations from linearity found. The effects are most pronounced in the CL0–2 condition, with the solid line being consistently below the other four lines across most levels of orthographic similarity (differences were statistically significant in all bins, with $p < .001$). To learn the words in this consistency category (all of which were isolated exception words), slight decreases in the linearity of most representations (i.e., distortions from componentiality) were required. A more notable difference is that the decrease is not uniform across all probe words, but highly localized, with the function dipping down abruptly for probe words most similar to words in the neighborhood.

To appreciate how local the effect of learning words from the CL0–2 neighborhood is in hidden-unit space, it is helpful to illustrate with an example word. When the network must learn the isolated exception word *pint* additionally, it changes the representation of the word itself, not surprisingly. The representations of similarly spelled words such as *print*, *mint*, and *hint*, all of which differ by a single grapheme, deviate slightly from linearity, and words that differ by two or more graphemes (e.g., *bent*) undergo even less distortion. There are many more of the latter words than the former words. This is shown by the solid gray line, which depicts the cumulative distribution of the 2,985 probe words. Each open circle represents the cumulative proportion of probe words in each bin, averaged across all 19 CL0–2 conditions. Note that the distribution starts at maximal similarity ($r = 1$), to which only the isolated exception word belongs, and rises very slowly, reaching only 5% by the middle (.5) of the similarity scale. On average, only 2% of probe words are in the orthographic vicinity of the newly learned exception, having a similarity rating of greater than 0.6. These data highlight just how locally the network warps the hidden unit space to accommodate a word that is without any friends amidst 10 or more enemies (12 on average). Probe words in close orthographic proximity to the exception, which form a tiny proportion of all words, are affected to a substantially greater degree than distant words.

One might think that a group of inconsistent words (CL2–4 and CL4–6), as opposed to an isolated exception (CL0–2), would cause a higher level of ambiguity, and thus learning the group would entail greater disruption of network structure. The current result suggests this is not the case. The CL2–4 and CL4–6 conditions required learning an average of, respectively, 2.4 and 4.8 friends at a time amidst 7.1 and 4.8 enemies, and as seen in Figure 3a, the network deviated from linearity much less than when it had to learn an isolated exception word. These data show that *inconsistency* is really a term applicable only to isolated exceptions, as any evidence of it in network structure quickly vanishes when there are just a couple of friends in the neighborhood. Network structure can remain highly componential even when multiple pronunciations of a word body must be learned.

The network behavior in Figure 3a fits with the insight gained from the statistical learning perspective presented earlier. Due to capacity-limited learning, the network's overall representation system is kept as componential as possible even with some exception words in the training set. Each time the network has to learn a neighborhood of new words, representations of them already exist in the network as nonwords, being extrapolations from the training set with relatively high componentiality maintained. Therefore, when learning new regular words, the network does not need to change the representation system at all. The current data suggest this tendency to absorb a neighborhood of friends into componential structure remains strong even when the neighborhood members are subregular in the presence of enemies (e.g., *bead, knead, plead* amid *bread, dead, head*). This decision is probably driven by capacity considerations: If capacity is limited, it must be more cost effective to expend capacity to accommodate an isolated exception word than two or more words from an inconsistent neighborhood, thereby minimizing exertion of total capacity.

Figure 3b shows the results of the analysis when the network was probed with nonwords (68,104 were used; details of their creation are in Supplement D). That is, the graph shows the change in linearity for nonwords when the network had to learn a neighborhood of words differing in consistency. The results are essentially the same as what was found with the training words as probes. Distortions in linearity due to learning a neighborhood emerge noticeably in the two lowest consistency conditions (CL2–4, CL0–2), and these effects are largest for probe nonwords that are orthographically very similar to the words in the neighborhood. For example, when the network learned the word *pint*, an

isolated exception, additionally, a nonword like *piant* underwent greater representational distortion than a distant nonword like *hout*. In contrast, in the case of learning a regular neighborhood (e.g., *came, fame*), no substantial local deviation from linearity was observed. Again, results from the conditions of ambiguous neighborhood learning (CL4–6, CL2–4) showed that the network exploits subregularities in neighborhoods of words, resulting in a much lower level of distortion in linearity than the case of learning a single exception (CL0–2).

Another outcome in Figures 3a and 3b that should not go unnoticed is that only very small deviations from linearity were required to learn words from neighborhoods with low consistency. Note that even though the greatest deviation from linearity is observed with the exception word itself in the CL0–2 condition (.0024), which is significantly different from zero ($p < .001$), it is infinitesimal on the measurement scale. This suggests that, as found in the small-scale analysis, even exception representations are very close to being linear (i.e., componential). This point can be seen even more clearly by examining the raw (unsubtracted) measurements of linearity *after* learning the full corpus of 2,998 words. The values are .9341 and .9246, respectively, when averaged over word and nonword probes. These data indicate that despite being perturbed locally, the network's re-representation system, even when probed exhaustively, remains close to a linear map.

Summary. The results of Analysis 1 answer our first question regarding why hidden representations of regular and exception words are similarly componential: The network learns exceptions in a way that minimally disrupts the linearity of representation. Very small and highly local perturbations are made to an otherwise global linear map in order to preserve componentiality. More specifically, when the ratio of friends to enemies becomes highly unbalanced, a small amount of componentiality must be sacrificed to learn the word. The current data also resolve the puzzle identified by Plaut et al. (1996), in which hidden representations of regulars and exceptions appear similarly componential yet can be treated in qualitatively different ways by the network.

Analysis 2: Effect of Neighborhood Learning on Network Generalizability

The small-scale analysis showed that learning exception words can involve the network adjusting both its representation system (input-to-hidden mapping) and its decision criteria (hidden-to-output mapping). Whereas Analysis 1 focused on the former aspect of learning, the second analysis focused on the latter by examining how exception learning affects nonword pronunciation (i.e., generalizability).

The incremental learning methodology was again used, except that model output was measured. Because the focus was on generalization, pronunciation of the 68,104 nonwords was measured without and with each of the 154 neighborhoods of varying consistency levels in the training corpus. The correctness of the pronunciation was judged according to pronunciation rules (Supplement D). Then, for each nonword, the *change* in network pronunciation from the without-neighborhood to the with-neighborhood condition was encoded into three different values: −1 for the change from incorrect to correct pronunciations, 0 for no change, and 1 for the change from correct to incorrect pronunciations. An accumulation of positive values indicates an increase in mispronunciations as a result of learning an additional neighborhood (e.g., an exception word), and negative values denote a decrease in mispronunciations. This measurement of change was calculated for all 154 neighborhoods. The measurements were then divided by the number of newly learned words in each neighborhood to normalize the effect for differences in neighborhood size.

The results of the simulation are shown in Figure 4. The data are displayed like those in Figure 3, except that the *y*-axis is now the change in mispronunciation rate after learning the neighborhood. Nonwords were grouped into 12 similarity bins, and individual changes in their pronunciation (i.e., −1, 0, or 1) were accumulated within each bin, normalized, and then averaged over all learning conditions within each of the five consistency classes.

Results show that generalization suffered only when the network had to learn a neighborhood in the lowest consistency class.[5] This is shown by the change in the mispronunciation rate being above zero only in the CL0–2 condition (solid line). The magnitude of the effect is extremely small in size and highly local in extent. On average, only the 35 nonwords (0.00051% of all nonwords) that are most similarly spelled to the exception word (those in the first two bins on the similarity scale) were subject to a small likelihood (0.7 to 1.6% chance) of mispronunciation after learning. Nonwords in the more distant bins were rarely mispronounced.

When learning a neighborhood of words in all other consistency conditions (CL2–4, CL4–6, CL6–8, CL8–10), the network made no additional pronunciation errors, regardless of the similarity of the probe nonwords to the neighborhood of words. Instead, small improvements in generalization (decreases in mispronunciation), were observed, primarily for nonwords most similar to the newly learned neighbors. That this pattern was found for the CL2–4 and CL-4–6 conditions provides further evidence that only a few instances of a given grapheme-phoneme mapping (2 or more friends) are needed for the network to learn the subregularity and thus improve generalizability, as the mispronunciation rate is reduced slightly after learning.

An unexpected outcome in the data is a greater decrease in mispronunciation observed in the CL2–4 condition (dotted line) relative to neighboring consistency conditions (dashed lines) across similarity bins 1 – 0.6. It seems that there is a stronger generalization benefit with these neighborhoods. Especially prevalent in this condition are orthographic neighborhoods with more than one rime (e.g., *there, were*) before learning a new neighborhood (e.g., *here*). In this situation, it may be that the gradient of generalizability from learning an additional neighborhood increases relative to the case of just one neighborhood being present.

[5] The rate of mispronouncing the 68,104 nonwords after learning the full corpus of 2,998 words was 18.12% when an optimal capacity constraint was used, and 29.54% when the constraint was removed (i.e., weight decay was turned off). However, our interest here is not in the absolute level of performance, but the *change* in performance as a result of learning an additional neighborhood. Use of such a large set of nonwords ensures the network is probed extensively for evidence of incremental change in order to measure mispronunciation rates accurately. Error rates will vary depending on the composition of nonwords. In creating our large set, we did not care how difficult to pronounce they would be, although most of them are highly unlikely onset-vowel or vowel-coda combinations that never appeared in the training words. See the General Discussion for a further discussion of this topic.

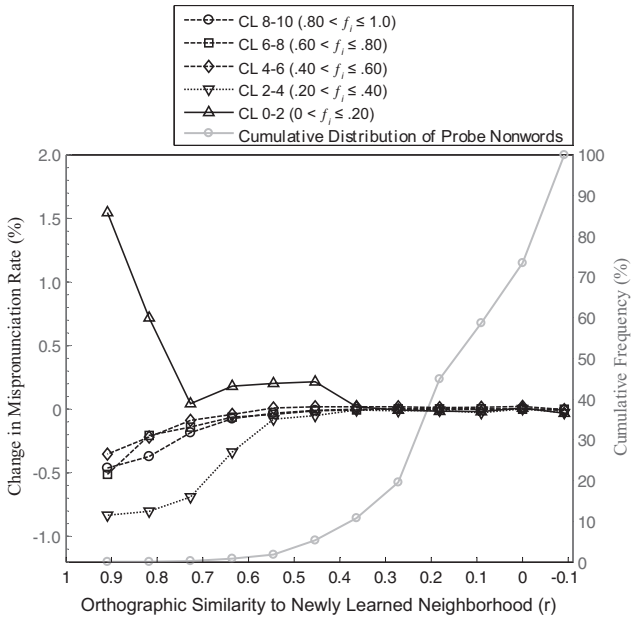

Figure 4. Change in the rate at which the network mispronounced 68,104 nonwords as a result of learning a neighborhood of words whose pronunciation varied in consistency. The data are arranged along the *x*-axis as a function of the orthographic similarity of each probe nonword to the neighborhood. CL = consistency level.

Finally, good network generalization requires not only forming componential representations but also, as was shown in the small-scale analysis, properly aligning decision bounds so that nonword representations are located on the same side of the boundary as their corresponding words. Critical to boundary placement is having enough instances of consistent pronunciations. Generalization from a neighborhood of regular words (CL8–10) is straightforward precisely because they are frequent enough to remove any uncertainty about pronunciation, and thus boundary placement. When exceptions (one or a group) are present in the training set and thus require distortions in representations, decision bounds have to be repositioned as well to ensure correct pronunciation of each exception word. As the results in Figure 4 show, a single instance is insufficient for good generalization to nearby nonwords. This is because a single exception must leave too much uncertainty about where to place the boundary, leading to mispronunciations. As the data from the CL2–4 and CL4–6 conditions show, only one or two additional examples are needed to eliminate that uncertainty and regain generalizability. More instances not only strengthen componential representations but also promote proper boundary placement, both of which work together to enhance generalization performance.

Summary. These simulation results answer the second question of why learning exceptions is minimally detrimental to the network's ability to generalize. Generalization is achieved when the network aligns linear decision bounds with componential representations by learning regular instances. This alignment of decision bounds relative to representations is tilted to learn exceptions (one or many), but any adverse effect on generalizability is kept to a minimum because the adjustment is small and local.[6] When learning subregular instances (i.e., friends > 1 surrounded by enemies), generalizability improves rather than degrades, demonstrating generalization from exceptions. These results show how impressively effective the learning mechanisms, which were identified in the small-scale network, are in a dense network of nonwords.

General Discussion

The goal of the present investigation was to explain how PDP models learn quasiregularity. We sought an explanation by studying how a PDP network represents both regular and exception items (Question 1) yet generalizes well (Question 2). We adopted a statistical modeling approach to the problem and analyzed the behavior of small-scale and large-scale networks of reading aloud. Results showed how multiple mechanisms work together to ensure regular and exception words are represented and pronounced in a way that leads to generalization. Capacity-limited learning forces the model to form componential representations of the input. Componentiality ensures good generalization because it increases the likelihood of the network's decision bounds aligning with the representations of novel items (i.e., nonwords in the Plaut et al., 1996, model) in a way that extrapolates regularity from training items. Exceptions are learned by slightly tweaking the representations and decision boundaries in the vicinity of the exceptions, thereby minimizing generalization errors. Analyses also showed that such accommodations are required only for highly inconsistent, isolated exceptions. Even groups of inputs (words in the Plaut et al., 1996, model) no larger than three items are enough for the network to represent without distortion and generalize from without error.

Generality of PDP Network Learning in Quasiregular Domains

The present study of PDP network learning focused on a specific type of quasiregularity in English word reading, but the basic functionality uncovered will hold for PDP models in other quasiregular learning domains (e.g., language, categorization, concept formation). What will differ is the degree of regularity across domains. For example, generation of English past tense or recognition of gender in French nouns may involve a greater or smaller number and variety of exceptions. Languages differ in the degree to which their morphology and writing systems are composed of regular and exception items. Whether reading aloud or producing the past tense, these tasks would most likely be modeled successfully by local warping of a linear representation system to account for the quasiregularity in the learning domain. Although the net-

[6] One lingering question is how the phenomenon of local perturbation is fundamentally possible in a distributed representation system. It would seem to violate the notion of a distributed system. Technically, this is a question about the sparsity of the representational space of a network. According to Cover (1965), a classification problem recast nonlinearly in a high-dimensional space is more likely to be solved with linear boundaries than in a low-dimensional space. This implies that, in high dimensions (i.e., networks with many hidden units), the linearity of representations is less likely to be globally deformed through network adjustment, and partial perturbations to linearity could help the network learn noncomponential mappings without disturbing nearby componential representations.

work's architecture and connectivity may differ depending on the nature of the task or the level of implementation precision (e.g., recursive feedback of hidden representation), we believe that the same principle of representation learning will hold as long as a similar type of capacity control drives the course of learning.

The notion of optimal capacity in network learning, studied extensively here, is in fact closely related to the degree of consistency in a particular learning domain. Given a certain training sample size (e.g., 3,000 words), a model's performance on both training and test items depends not only on the expressive capacity of the model but also on the complexity of the given task (i.e., what proportion and degree of inconsistent mappings are present). This relationship entails the concept of a minimum training set size, namely, sample complexity, that is required to achieve a certain level of generalization (Anthony & Biggs, 1997; Blumer et al., 1989). For example, a reading network's nonword reading improves rather than degrades when a small subset of exception words (e.g., 48 exception words used in Plaut et al., 1996) is removed from the training corpus, and, as the data in Figure 4 suggest, this is due to fewer mispronunciations of nonwords in the orthographic vicinity of the exceptions. Another dimension of complexity is severity of exceptions; severe exceptions (e.g., *pint*→/pom/ vs. *pint*→/pInt/) have more detrimental effects on network performance than less severe ones (see Supplement E for a simulation demonstrating the effects of training data complexity). As long as a reasonable degree of consistency is present in the training sample, the network exhibits quasiregular behavior (e.g., learn regularity from exceptions) because, under the pressure of capacity constraint, it tends to exploit componential correspondences even amid exception items.

The conclusions regarding local warping can seem at odds with the network's overall mispronunciation rate, which for the set of 68,104 nonwords was 18.12% (footnote 5). An understanding of this apparent inconsistency can be gained by considering the relationship among sample complexity, network capacity, and generalization. Although effects of exception learning are highly local, these effects should accumulate over many exceptions, thus lowering overall generalization performance. The question then turns to how local these effects can possibly be. Although a detailed answer requires further research, computational learning theory provides some insight. Given a certain quasiregular task (e.g., English or German word reading), the amount of regularity that a PDP network can learn depends on sample complexity. A consequence of this is that if a training set of limited size (e.g., only monosyllabic words) is used, there is a bound beyond which it cannot learn no matter what training algorithm is used. In this situation, the only way to increase generalizability is to use a larger training set. This statement holds true for PDP models because they are universal approximators (Hornik et al., 1989) that learn regularity by relying solely on training input under capacity constraint. Thus, the high mispronunciation rate we observed could be reduced substantially by using a larger training set, which should improve generalizability by letting the model learn from additional examples.

Last, the current study focused on a particular kind of capacity constraint that is achieved by limiting the magnitude of connection weights rather than the number of hidden units. Although the number of hidden units, or more generally the number of model parameters, is commonly associated with model capacity, the *optimal* capacity for maximal generalization is not always determined on the scale of the number of parameters. When learning involves mapping high-dimensional inputs and outputs (e.g., graphemes and phonemes in reading), a minimum number of hidden units is required to learn the training sample and generalize, usually resulting in a network in which the number of connection weights is considerably larger than the number of training examples (e.g., the current reading network adapts 16,761 weights to learn 2,998 training words). In this particular situation, the notion of an optimal number of hidden units is not as appropriate as in general cases; rather, the magnitude of weights becomes more relevant. This was shown formally by Bartlett (1998), who proved that *if* a large PDP network is employed for a pattern classification problem and the training algorithm finds a network with small weights that learns the training set, then the number of hidden units becomes inconsequential, and instead, the size of the weights is the major factor that determines generalization performance.

Implications for PDP Modeling

Insufficient understanding of PDP models' inner workings has made it difficult for their explanatory value to be rightly appreciated (McCloskey, 1991). Even though a fair number of studies showed the emergence of varying degrees of context sensitivity in networks' hidden representations (e.g., Elman, 1990; Plaut & Gonnerman, 2000), a sufficiently clear understanding has not been gained concerning fundamentally why those representations hold discriminative power and simultaneously ensure good generalization. Conventionally, the focus has been on the structure of the content in a learning domain (e.g., hierarchy of categories). A network analysis was then performed, hoping to discover analogous structure in the network's hidden-unit space to a degree commensurate with conceptual distinctions exhibited in the learning domain (e.g., Bullinaria, 1997; Elman, 1990; Hanson & Burr, 1990; Plaut & Gonnerman, 2000; Rogers & McClelland, 2004; Sejnowski & Rosenberg, 1987). Some of these analyses took advantage of general-purpose multivariate techniques (e.g., hierarchical cluster analysis, principal component analysis), in which an implicit assumption is that the structure to be found must be observable using the metric built into the methods.

The methodology applied in the present study enabled us to probe representation from an alternative perspective, and in doing so provided a comprehensive and coherent account of a PDP model's quasiregular performance. Our analysis turned attention to the constraints embedded in learning, considered their implications for representation formation and output performance, and sought to observe possible aberrations of all degrees across the entire learning system. This approach will not only apply directly in probing PDP models in other quasiregular domains but also serve as a useful strategy for modelers who want to test different types of capacity-limiting schemes and study their consequences on representation learning (e.g., see Weigend, Rumelhart, & Huberman, 1991, for *weight elimination*; Simard, LeCun, & Denker, 1992, for *tangent propagation*; Nowlan & Hinton, 1992, for *soft weight sharing*; Gutjahr, 1999, for *orthonormal weights*).

The present study's emphasis on capacity-limited learning has a natural connection to the issue of a PDP models' testability versus its generality: Any good model should not generate an arbitrarily wide range of predictions but still be able to provide a good fit to

a sufficient variety of empirical data (Estes, 1988). It is now widely known that PDP models, when properly trained, do not overfit or give an arbitrarily good fit to any data. Nonetheless, details have not been well studied such as whether a particular type of constraint applies generally enough for modeling various task domains or it is better suited to making a particular kind of predictions, and to what extent adjusting the strength of a certain constraint can cause networks to yield different behavioral patterns. To that end, the current analysis of network learning suggests that a PDP modeler could exploit the ability to tune the predicted degree of localization (e.g., how far and how often exception learning affects nonword reading) to provide a better empirical fit to a data set. This could also be used to model differences in generalization behavior across domains or across individuals and could be achieved by varying the type and the strength of the capacity constraint being employed.

Conclusion

The power of PDP models lies in their ability to learn representations, allowing them to master quasiregularity in many domains exceedingly well. The results of the current study provide a window into the source of that power. Local perturbations of an otherwise linear system ensure maximal generality and good exception learning. Mechanistically, these traits are achieved by, when necessary, warping the representational space and tilting decisional boundaries in an otherwise componential system. The elucidation of these basic properties of network functioning should assist modelers in understanding the causes and consequences of network behavior and in assessing their suitability as models of mind and brain.

References

Anthony, M. H. G., & Biggs, N. (1997). *Computational learning theory*. Cambridge, England: Cambridge University Press.

Bakker, P. E. (1995). *On the implementation of quasiregular mappings by feedforward connectionist networks* (Unpublished doctoral dissertation). University of Queensland, Brisbane, Queensland, Australia.

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory, 44,* 525–536. doi:10.1109/18.661502

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM, 36,* 929–965. doi:10.1145/76359.76371

Bullinaria, J. (1997). Analyzing the internal representations of trained neural networks. In A. Browne (Ed.), *Neural network analysis, architectures and algorithms* (pp. 3–26). Bristol, England: IOP.

Coltheart, M. (1978). Lexical access in simple reading tasks. In G. Underwood (Ed.), *Strategies of information processing* (pp. 151–216). London, England: Academic Press.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers, 14,* 326–334. doi:10.1109/PGEC.1965.264137

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York, NY: Wiley.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14,* 179–211. doi:10.1207/s15516709cog1402_1

Estes, W. K. (1988). Toward a framework for combining connectionist and symbol-processing models. *Journal of Memory and Language, 27,* 196–212. doi:10.1016/0749-596X(88)90073-3

Gutjahr, S. (1999). Improving generalization performance of neural networks by constructing orthonormal weight vectors. In F. Masulli & R. Parenti (Eds.), *Proceedings of the third ICSC symposia on intelligent industrial automation (IIA'99) and soft computing (SOCO'99), Genova, Italy* (pp. 742–747). Rochester, NY: ICSC Academic Press.

Hanson, S. J., & Burr, D. J. (1990). What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences, 13,* 471–489. doi:10.1017/S0140525X00079760

Harm, M. W., & Seidenberg, M. S. (1999). Phonology, reading acquisition, and dyslexia: Insights from connectionist models. *Psychological Review, 106,* 491–528. doi:10.1037/0033-295X.106.3.491

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,* 662–720. doi:10.1037/0033-295X.111.3.662

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York, NY: Springer. doi:10.1007/978-0-387-84858-7

Haykin, S. O. (1999). *Neural networks: A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2,* 359–366. doi:10.1016/0893-6080(89)90020-8

Jared, D., McRae, K., & Seidenberg, M. S. (1990). The basis of consistency effects in word naming. *Journal of Memory and Language, 29,* 687–715. doi:10.1016/0749-596X(90)90044-Z

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. Cambridge, MA: MIT Press.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science, 2,* 387–395. doi:10.1111/j.1467-9280.1991.tb00173.x

Nowlan, S. J., & Hinton, G. E. (1992). Simplifying neural networks by soft weight-sharing. *Neural Computation, 4,* 473–493. doi:10.1162/neco.1992.4.4.473

Plaut, D. C., & Gonnerman, L. M. (2000). Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing? *Language and Cognitive Processes, 15,* 445–485. doi:10.1080/01690960050119661

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. doi:10.1037/0033-295X.103.1.56

Rogers, T. T., & McClelland, J. L. (2004). *Semantic cognition: A parallel distributed processing approach*. Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing? In B. MacWhinney (Ed.), *Mechanisms of language acquisition* (pp. 195–248). Hillsdale, NJ: Erlbaum.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96,* 523–568. doi:10.1037/0033-295X.96.4.523

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

Simard, P., LeCun, Y., & Denker, J. S. (1992). Efficient pattern recognition using a new transformation distance. *Advances in Neural Information Processing Systems, 5,* 50–58.

Thomas, M. S., & McClelland, J. L. (2008). Connectionist models of cognition. In R. Sun (Ed.), *The Cambridge handbook of computational psychology* (pp. 23–58). London, England: Cambridge University Press. doi:10.1017/CBO9780511816772.005

Treiman, R., Mullennix, J., Bijeljac-Babic, R., & Richmond-Welty, E. D. (1995). The special role of rimes in the description, use, and acquisition

of English orthography. *Journal of Experimental Psychology: General, 124,* 107–136.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27,* 1134–1142. doi:10.1145/1968.1972

Vapnik, V. N. (1992). Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems, 4,* 831–838.

Vapnik, V. N. (1998). *Statistical learning theory*. New York, NY: Wiley.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications, 16,* 264–280. doi:10.1137/1116025

Weigend, A. S., Rumelhart, D. E., & Huberman, B. A. (1991). Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing Systems, 3,* 875–882.

Zorzi, M., Houghton, G., & Butterworth, B. (1998). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance, 24,* 1131–1161. doi:10.1037/0096-1523.24.4.1131

Received January 2, 2013
Revision received July 22, 2013
Accepted July 22, 2013 ■