

Reading Exception Words and Pseudowords: Are Two Routes Really Necessary?*

David C. Plaut

Department of Psychology
Carnegie Mellon University

James L. McClelland

Department of Psychology
Carnegie Mellon University

Mark S. Seidenberg

Neuroscience Program
University of Southern California

To appear in J. P. Levy, D. Bairaktaris, J. Bullinaria, and P. Cairns (Eds.) (1995). *Proceedings of the Second Neural Computation and Psychology Workshop*. London: University College London Press.

This paper describes simulation experiments demonstrating that a unitary processing system in the form of a connectionist network is capable of learning to read exception words and pronounceable nonwords aloud. We trained such a network on the 3000 word corpus used by Seidenberg and McClelland (1989). After training, the network was able to read over 99% of the training corpus correctly. When tested on the lists of pronounceable nonwords used in several experiments, its accuracy was closely comparable to that displayed by human subjects.

The work addresses the ongoing debate about the nature of the mechanisms that are used in reading words aloud. One view, defended most recently by Coltheart, Curtis, Atkins, and Haller (1993), states that adequate performance on both pronounceable nonwords and exception words depends on the use of two separate mechanisms, one that applies rules of grapheme phoneme correspondence and another that retrieves pronunciations specific to particular familiar words. An alternative view, expressed by Seidenberg and McClelland, is that a single system may be capable of learning to read both kinds of letter strings.

The work relates more generally to the ongoing debate about the nature of the processing systems underlying human language use. The question is, should these systems be viewed as systems that learn and apply an explicit system of rules, augmented where necessary with an explicit enumeration of exceptions; or should these systems be viewed instead as connectionist systems that gradually develop sensitivity to the structure inherent in the mapping they are asked to learn, through a gradual learning procedure.

The connectionist approach has considerable appeal, because it accounts for the fact that regularity is a graded phenomenon and for the fact that human language users are sensitive to these gradations. In the domain of spelling-to-sound translation, Glushko (1979) was the first to emphasize that in fact the crucial variable for word pronunciation tasks is not regularity, but degree of consistency of the spelling-to-sound correspondences exhibited by one word to the correspondences exhibited by its neighbors. Since Glushko's work it has been clearly established that the more consistent a word's spelling to sound correspondences are

*This research was supported financially by the National Institute of Mental Health (Grants MH47566 and MH00385), the National Institute on Aging (Grant Ag10109), the National Science Foundation (Grant ASC-9109215), and the McDonnell-Pew Program in Cognitive Neuroscience (Grant T89-01245-016). We thank Marlene Behrmann, Derek Besner, Max Coltheart, Joe Devlin, Geoff Hinton, and Eamon Strain for helpful discussions and comments.

Correspondence concerning this paper should be sent to Dr. David C. Plaut, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213-3890, p1aut@cmu.edu.

with those of its neighbors, the more rapid and accurate responses to that word will be. The connectionist approach, as exemplified by the model of Seidenberg and McClelland (hereafter SM89), captures this graded consistency effect. It also captures the interaction of consistency and frequency that has been repeatedly found in experiments like those of Taraban and McClelland (1987) and Waters and Seidenberg (1985). As Seidenberg and McClelland showed, their connectionist model accounted in detail for the pattern of performance obtained in a large number of different experiments investigating frequency and consistency effects.

However, the model proposed by Seidenberg and McClelland did not perform adequately in reading pronounceable nonwords. Attention has focused particularly on the nonwords used by Glushko (1979) and by McCann and Besner (1987). Besner, Twilley, McCann, and Seergobin (1990) pointed out that the SM89 model fell far short of human performance on either of these two word lists.

Two interpretations of these results have been proposed. Besner, Coltheart and their colleagues have suggested that the deficiencies of the SM89 model reflect the fact that no single system can actually read both exception words and pronounceable nonwords adequately. SeidenbergMcClelland (1990) suggested instead that limitations in the SM89 simulation might have prevented the model from successfully capturing the nonword reading data. They suggested that the model's performance on nonwords might be improved with either a larger training corpus or a different choice of input-output representations.

In the work we report here, we have focussed our attention on the issue of representation. We will show that when a different representation is used, the networks' ability to read pronounceable nonwords dramatically improves.

1 Representations

To understand the motivation for the new representation we have chosen, it is necessary to consider other possible representations that might be used. The most obvious thing to try is what we call a "slot-based" representation. On this approach, the first input letter goes to the first slot, the second to the second slot, etc. Similarly in the output, the first phoneme goes in the first slot, and so on. With enough slots, words up to any desired length can be represented.

In a connectionist system, in which the content of each slot is represented by a pattern of activation, and in which the knowledge that determines how the content of the slot is processed is stored in a set of connection weights, there is a problem with this scheme. The problem is that the same knowledge must be replicated several times. To see this, consider the words LOG, GLAD, and SPLIT (see Table 1). The fact that the letter L corresponds to the phoneme /l/ must be stored three separate times in this system. So the knowledge underlying this regularity is dispersed across positions, and there is no generalization of what is learned about letters in one position to the same letter in other positions. One can try to alleviate the problem a little by aligning the slots in various ways, but it doesn't go away completely. Adequate generalization still requires that the correspondences be learned separately in each of several slots.

Seidenberg and McClelland tried to get around this problem by using a scheme that avoided the specific limitations of the slot-based approach, but in the end it turns out to have a version of the same problem. They represented letters and phonemes, not in terms of their absolute spatial position, but in terms of the adjacent letters to the left or the right. This approach, which originated with Wickelgren (1969), makes the representation of each element context sensitive without being rigidly tied to position. Unfortunately, however, the knowledge of spelling to sound correspondences is still dispersed across a large number of different contexts, and adequate generalization still requires that the training effectively covers them all.

The hypothesis that guided the work we report here was the idea that this dispersion of knowledge is what prevented the network from exploiting the structure of the English spelling to sound system as fully as human readers do. We set out, therefore, to design a representation that minimizes this dispersion.

Table 1: The Dispersion Problem

| Slot-based representations | | | | | | | | | | |
|--|-----|-----|-----|-----|----------------|-----|-----|---|---|---|
| Left-justified | | | | | Vowel-centered | | | | | |
| 1 | 2 | 3 | 4 | 5 | -3 | -2 | -1 | 0 | 1 | |
| L | O | G | | | | | | S | U | N |
| G | L | A | D | | | S | W | A | M | |
| S | P | L | I | T | S | P | L | I | T | |
| Context-sensitive triples (“Wickelgraphs”) | | | | | | | | | | |
| LOG: | | | | | *LO | LOG | OG* | | | |
| GLAD: | | | | *GL | GLA | LAD | AD* | | | |
| SPLIT: | *SP | SPL | PLI | LIT | IT* | | | | | |

The limiting case of this approach would be to have a single set of letter units, one for each letter in the alphabet, and a single set of phoneme units, one for each phoneme. Unfortunately, this approach suffers from the problem that it cannot distinguish TOP from POT or SALT from SLAT. However, it turns out that a scheme that involves only a small amount of duplication is sufficient for unique representation of virtually all uninflected monosyllables. By definition, a monosyllable contains only a single vowel, so we only need one set of vowel units. A monosyllable may contain both an initial and a final consonant cluster, and almost every consonant can occur in either the initial or the final cluster, so we need separate sets of consonant units for the initial and final consonant clusters.

The remarkable thing is that this is nearly all that we need. The reason is that within an initial or final consonant cluster there are strong phonotactic constraints. At both ends of the syllable, each phoneme can only occur once, and the order of phonemes is strongly constrained. Given this, if we know which phonemes are present in the initial consonant cluster, which phoneme is the vowel, and which phonemes are present in the final cluster, then we can use the phonotactic constraints to uniquely determine the order in which these phonemes occur. These constraints can be captured by simply listing the units from left to right to correspond to the left-to-right ordering constraints that exist in initial and final consonant clusters. Once we do this, readout occurs by simply reading out the phonemes that are active in sequence from left to right (see Figure 1).

There are a small number of cases in which two phonemes can occur in either order within a consonant cluster. One such pair is /s/ and /p/. The /s/ can precede the /p/ as in CLASP; or the /s/ can follow the /p/ as in LAPSE. To allow handling of such cases, it is necessary to add units disambiguating the order. To do this we include a unit for /ps/. The convention is that when /s/ and /p/ are both active the order is /s/ then /p/, unless the /ps/ unit is also active, in which case the order is taken to be /ps/. To cover the pronunciations of the items in the SM corpus, we needed only 3 such units, for /ps/, /ts/, /ks/. Interestingly, some of these combinations are sometimes treated as single phonemes called affricates and each is written with a single character in at least one Western orthography (Greek ψ , German Z, English X).

We use a similar scheme to represent the spellings of words. Because English is an alphabetic language, this scheme works almost as well for spelling as it does for sound. There is one complication, however. As Venezky (1970) pointed out, the spelling units that correspond to sounds in English are not necessarily single letters. Rather, they are relational units sometimes called graphemes, consisting of one, two or sometimes even three or four letters. Because the spelling-sound regularities of English are grapheme-phoneme correspondences, the regularities in the system are most elegantly captured if the units represent the graphemes present in the string rather than simply the letters that make up the word. Unfortunately, it

Phonology

(s) (b) (p) (d) (t) (g) (k) (f) (v) (z) (T) (D) (S) (Z) (l) (r) (w) (m) (n) (h) (y)
 (a) (@) (e) (i) (o) (u) (A) (E) (I) (O) (U) (W) (Y) (^)
 (r) (l) (m) (n) (N) (b) (g) (d) (ps) (ks) (ts) (s) (f) (v) (p) (k) (t) (z) (S) (Z) (T) (D)

Orthography

(Y) (S) (P) (T) (K) (Q) (C) (B) (D) (G) (F) (V) (J) (Z) (L) (M) (N) (R) (W) (H) (U)
 (CH) (GH) (GN) (GU) (PH) (PS) (QU) (RH) (SH) (TH) (TS) (WH)
 (E) (I) (O) (U) (A) (Y)
 (AI) (AU) (AW) (AY) (EA) (EE) (EI) (EU) (EW) (EY) (IE) (OA) (OE) (OI) (OO) (OU) (OW) (OY) (UE) (UI) (UY)
 (H) (R) (L) (M) (N) (B) (D) (G) (C) (X) (F) (V) (J) (S) (Z) (P) (T) (K)
 (BB) (CH) (CK) (DD) (DG) (FF) (GG) (GH) (GN) (GU) (KS) (LL) (NG) (NN) (PH) (PP) (PS) (QU) (RR) (SH) (SL) (SS)
 (TCH) (TH) (TS) (TT) (ZZ) (E) (ES) (ED)

Figure 1: The phonological and orthographic representations. The meanings of the symbols in phonology are as follows: /a/ in POT, /@/ in CAT, /e/ in BED, /i/ in HIT, /o/ in DOG, /u/ in GOOD, /A/ in MAKE, /E/ in KEEP, /I/ in BIKE, /O/ in HOPE, /U/ in BOOT, /W/ in NOW, /Y/ in BOY, /^/ in CUP, /N/ in RING, /S/ in SHE, /C/ in CHIN /Z/ in BEIGE, /T/ in THIN, /D/ in THIS. All other phonemes are represented in the conventional way (e.g., /b/ in BAT).

is not always clear what graphemes are present in a word. Consider the word COLTHEART. In this case, there is a T next to an H, so we might suppose that the word contains a TH grapheme, but in fact it does not; if it did it would be pronounced “col-thart”. Thus it is apparent that the input is ambiguous in such cases. Because of this, there is no simple procedure for translating letter strings into the correct sequence of graphemes. However, it is possible to translate a letter sequence into a pattern of activation representing all possible graphemes in the string, and this is what we did in our model. So in the case of COLTHEART, we would activate units for T, for H, and for TH.

2 Simulation 1: Feedforward Network

The network we used consisted of 108 orthographic input units, one for each of the Venezky graphemes that occurs in the initial consonant cluster, the vowel cluster, or the final consonant cluster. There were also 57 phonological output units, and 100 intermediate or hidden units. The grapheme units are fully connected to the hidden units which, in turn, are fully connected to the phoneme units. To encode any letter string forming an uninflected monosyllable into an input pattern, we parse the string into onset, vowel, and coda, and activate all of the units corresponding to graphemes contained in each cluster. To translate the output into a pronunciation, we simply scan the output units from left to right, outputting in order the phonemes with activation greater than 0.5.

We trained the network on the entire list of 2897 monosyllabic words in the SM89 corpus, augmented with 101 words missing from the corpus but used as word stimuli in various experiments. We also included training patterns consisting of each single grapheme and the corresponding phoneme. These were included because children are taught these correspondences in the process of learning to read, although equivalent networks not trained on these correspondences exhibit equivalent behavior (see Plaut, McClelland, Seidenberg, & Patterson, 1994). Furthermore, rather than present each word with a probability proportional to its frequency of occurrence (Kučera & Francis, 1967) and update the weights immediately, we accumulated the error derivatives for the training cases, each weighted by its frequency, before changing the weights. This enabled the learning rates on each connection to be adapted independently during training using the delta-bar-delta procedure (Jacobs, 1988, with an additive increment of 0.1 and a multiplicative decrement of 0.9). Finally, phoneme units within 0.2 of their target values (0 or 1) for a particular training item did not receive any error for that presentation.

Two instances of the same network, differing only in the random initial connection weights, were trained using back propagation for 300 epochs (with a global learning rate of 0.002 and momentum of 0.9). Performance steadily improved over time. Using the scoring procedure described above, the networks averaged 99.9% correct after training. One network made six and the other made two errors, all on very low-frequency exceptions.

Given this very high level of mastery of the training corpus, we are ready to consider our two competing hypotheses. One of these, due to Coltheart, Besner and their colleagues, is that no single network is capable of learning to read both exceptions and pronounceable nonwords. According to this hypothesis, we should expect our networks to fare poorly on pronounceable nonwords; either as poorly as the SM89 model did; or, if not that poorly, at least considerably more poorly than human subjects. The other hypothesis, the one that we have put forward above, is that in fact a single network can learn to read both exceptions and pronounceable nonwords, as long as the regularities embodied in the spelling sound system are not too dispersed. On this hypothesis, we expect a substantial improvement in performance on nonwords relative to the SM89 model.

First, let's begin with the results of experiments reporting nonword reading performance. We consider here two such experiments. Experiment 1 of Glushko (1979), in which he tested subjects' reading of two sets of nonwords: one set derived from regular words such as DEAN and another set derived from exception

Table 2: Percent of Regular Pronunciations of Nonwords

| | Glushko nonwords | | McCann-Besner nonwords |
|----------------------|------------------|-----------|------------------------|
| | Regular | Exception | Control |
| Human Subjects | 93.8 | 78.3 | 88.6 |
| SM89 network | 60.5 | 53.5 | 51.0 |
| Feedforward networks | 94.2 | 69.8 | 88.8 |

words such as DEAF. Corresponding example nonwords would be HEAN and HEAF. The second experiment is one by McCann and Besner (1987), in which they tested a set of pseudohomophone pseudowords like BRANE and a set of control pseudowords like FRANE. For present purposes, we concentrate on the control items since we believe pseudohomophone effects are mediated by aspects of the reading system that are not implemented in our simulation (e.g., semantics and/or articulatory processes; see Seidenberg, Petersen, MacDonald, & Plaut, in press).

Table 2 presents the percentage of times subjects used “regular” pronunciations, for both sets of Glushko stimuli and for the McCann and Besner nonwords. Also included is the performance of the earlier, SM89 model, and the new model. We can see that the original SM89 model fared very poorly relative to human subjects. The new model, by contrast, produces a dramatic improvement. It produces the correct, regular response as often as human subjects in two conditions: the Glushko regular nonwords and the McCann and Besner control nonwords. For the the Glushko exception nonwords, however, both the model and the human subjects fail to give the regular response on a substantial portion of items.

To delve more deeply into the data, it is important to consider how one determines whether a response is regular or not. This is not a simple matter, since it is far from clear *a priori* how to define what counts as a regular response. Consider the nonwords GROOK and WEAD. These items involve cases in which what counts as regular depends on whether we consider the specific context in which the vowel occurs. /U/ as in BOOT is the most common pronunciation of OO, so we might be tempted to treat /grUk/ as regular. But final OOK is pronounced /uk/ as in TOOK in 11 of the 12 words ending in OOK. Which one counts as regular depends on the scoring procedure we choose to employ. For the simulation results we have shown thus far, responses were considered regular only when they conformed to abstract spelling to sound correspondences. Thus /grUk/ was considered to be the regular pronunciation for GROOK and /wEd/ as in WEED was considered to be the correct pronunciation for WEAD. We believe the criterion is at least as strict as the one used by Glushko, and is probably somewhat stricter than the one McCann and Besner used.

Let us now consider in more detail the performance of both Glushko’s subjects and the model on his exception nonword stimuli (see Figure 2). Starting with the subjects, Glushko considered how many of the non-regular responses were consistent with any of the different pronunciations of the word’s body that occur in the Kuçera and Francis (1967) corpus. For example, given GROOK, /gruk/ would now be considered correct, and for WEAD, the response /wed/ as in DEAD would become acceptable. This criterion accounted for 80% of the non-regular responses, leaving only 4.1% of the responses unaccounted for. We applied the same criterion to the model’s performance, and found that it accounted for 24 out of the 26 non-regular responses, leaving an error rate of only 2.3%. Both our model, and human subjects, read nonwords derived from exceptions in a way that is not always consistent with abstract GPC rules, but is consistent with at least one of the possible pronunciations of the word’s body, almost all of the time.

On the data considered thus far, the model appears a little more likely than Glushko’s subjects to use a pronunciation other than the one dictated by abstract GPC rules. What is it doing in these cases? It appears that it has a strong tendency to choose the most frequent vowel correspondence found among the words in

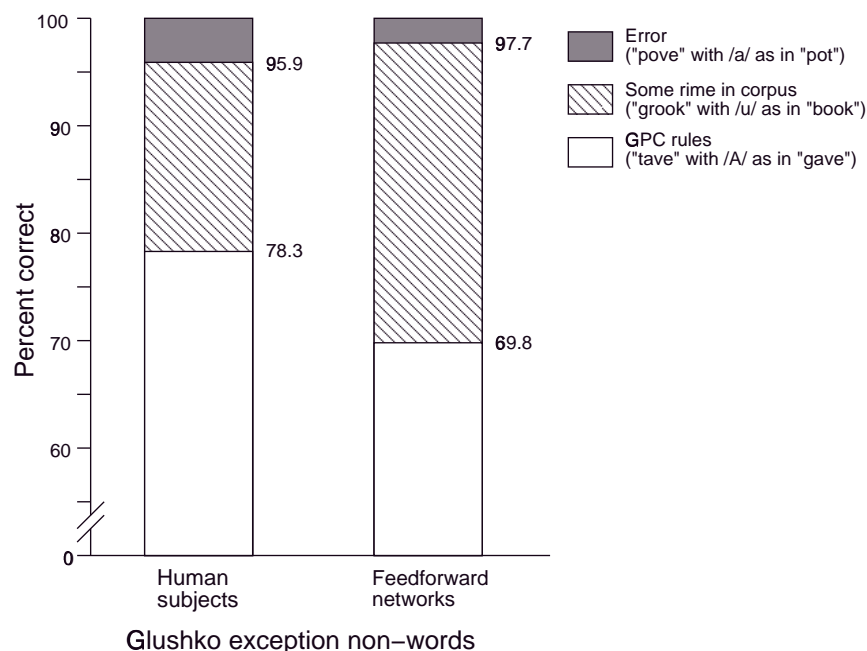


Figure 2: Percent correct on nonword reading for human subjects and the feedforward networks as scored according to GPC rules or according to whether the pronunciation matches some rime for the same body in the corpus.

the training corpus that share the same body. Thus, /uk/ is the most frequent pronunciation of OOK in the corpus, and both runs of the network pronounced GROOK as /gruk/. Overall, in 14 of the 24 cases in which a correspondence other than the abstract GPC correspondence was used, the response actually agreed with the most frequent correspondence for that word body in the model's training corpus.

We are now in a position to consider whether the model's performance on the Glushko exception nonwords represents a meaningful deviation from human performance. We believe it does not. If anything, it represents a slight relative difference in the tendency to rely on the pronunciation of word body neighbors, but this tendency is obviously present both in the model and in human subjects, and any difference is only a minor matter of degree. Furthermore, many low-frequency words in English are absent for the training corpus. These words tend to be regular by abstract GPC rules, and if they were included they would increase the extent to which the most frequent correspondence of a word body corresponds to the one that is regular by GPC rules. Therefore we would expect an increase in the tendency of the model to adhere to GPC rules, if it were trained on a more complete sample of the actual words in the English language.

It must be acknowledged that the model is not absolutely perfect in its reading of nonwords—it occasionally makes a frank mistake, such as leaving out a sound that should be pronounced, or pronouncing PH as /p/ instead of /f/. But in our view, the model comes close enough to mimicking human performance to seriously threaten the Coltheart-Besner hypothesis. This hypothesis claims that no single system can read both exception words and pronounceable nonwords. Our model comes exceedingly close to doing exactly that.

There is one last escape for the Coltheart-Besner hypothesis. This is the possibility that the network has partitioned itself into two sub-networks, one that reads regular words, and another that reads exceptions. If this were the case, we would expect some of the hidden units to contribute to exception words and not nonwords, and others to contribute to nonwords but not exceptions. To investigate this possibility, we considered each hidden unit to make a significant contribution to pronouncing a given item if, when that unit

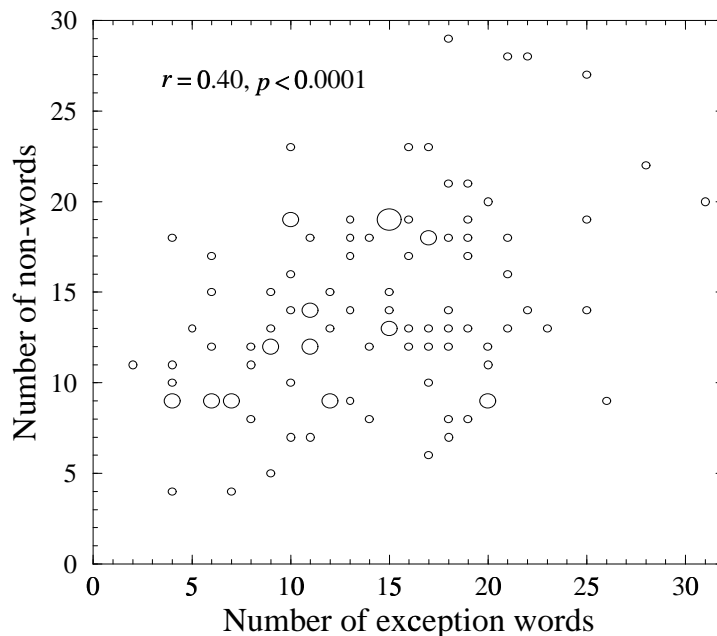


Figure 3: The numbers of exception words and nonwords ($n = 48$ for each) to which each hidden unit makes a significant contribution, as indicated by an increase in error of at least 0.025 when the unit is removed from the network. Each circle represents one or more hidden units, with the size of the circle proportional to the number of hidden units making significant contributions to the numbers of exception words and nonwords indicated by the coordinates of the circle.

is removed from the network, the overall error increases by some criterion (e.g., 0.025). We then counted the number of Taraban and McClelland’s (1987) 48 exception words and the number of 48 body-matched nonwords to which each hidden unit makes a significant contribution. Contrary to the hypotheses that hidden units had specialized for exception words versus nonwords, almost all hidden units make substantial contributions to both exception words and nonwords. In fact, there tends to be a positive correlation between the number of exception words a hidden unit contributes to and the number of nonwords ($r = 0.40$, $p < 0.0001$; see Figure 3). Thus it appears that the network has not in fact partitioned itself into a subset that learns the rules of English and a separate system that learns the exceptions.

3 Simulation 2: Refixation Network

The network developed in Simulation 1 succeeds at reading nonwords because its representations condense the regularities present within and between orthography and phonology. Although we designed these representations ourselves, based on phonotactic and orthotactic constraints, a network itself can learn to condense the regularities that are dispersed in a position-specific letter representation by refixating the input string. To demonstrate this, we trained a simple recurrent network (Elman, 1990) to produce sequences of single phonemes as output when given position-specific letters as input.

The architecture of the network is shown in Figure 4. There are 26 letter units and a “blank” unit at each of 10 positions. The third position from the left, indicated by the dark rectangle in the Figure, corresponds to the point of fixation. These 270 letter units are fully connected to 100 hidden units which, in turn, are fully connected to 36 phoneme units. The hidden units also receive input from the previous states of phoneme units. In addition, there is a fourth group of *position* units that the network uses to keep track of where it is in the letter string as it is producing the appropriate sequence of phonemes, analogous to a focus of

attention. Two copies of this group are shown in the Figure simply to illustrate their behavior over time. These position units have connections both to and from the hidden units.

In understanding how the network is trained, it will help to consider first its operation after it has achieved a reasonable level of proficiency at its task. A letter string is presented with its first letter at fixation, by activating the appropriate letter unit at each corresponding position, and the blank unit at all other positions. We assume that position information for internal letters is somewhat inaccurate, so that the same letter units at neighboring internal positions are also activated slightly (to 0.3). In Figure 4, the grey regions for letter units are intended to indicate the activations for the word BAY when fixating the B. Initially, the position unit at fixation (numbered 0 by convention) is active and all others are inactive, and all phoneme units are inactive. (The Figure shows the network after generating B \Rightarrow /b/ and attempting AY \Rightarrow /A/.) Hidden unit states are initialized to 0.2.

The network then computes new states for the hidden units, phoneme units, and position units. The network has two tasks: 1) to activate the phoneme corresponding to the currently-attended grapheme (as indicated by the position units), and 2) to activate the position of the *next* grapheme in the string (or, if the end of the string is reached, the position of the adjacent blank). For example, when attending to the letter B at fixation in BAY, the network must activate the /b/ unit and position unit 1 (the position of AY in the input). To the extent that the activations over the phoneme and position units are inaccurate (i.e., not within 0.2 of their correct values), error is injected and back-propagated through the network.

Assuming that the network succeeds at generating the correct phoneme and position, this information is then used to guide the production of the next phoneme. As shown in Figure 4 for BAY, position unit 1 and the phoneme /b/ are now active, the letter input remains the same, and the network must activate /A/, the phoneme corresponding to the indicated grapheme AY, as well as position unit 3 (corresponding to the blank following the string, thereby indicating a complete pronunciation). In general, when pronouncing a letter string of arbitrary length, the network activates the sequence of phonemes corresponding to its pronunciation, while simultaneously keeping track of the position of the grapheme it is currently working on. If, throughout the process, every phoneme and position is generated correctly, the activations over the letter units remains fixed.

If, however, the network fails at generating the correct phoneme or position at some point in pronouncing a word, it *refixates* the input string so that the problematic grapheme falls at fixation. The network has information sufficient to do this because it generated the position of this grapheme (relative to fixation) on the previous time step. Although the network thus indicates *where* to fixate, the actual refixation is carried out by an external procedure that shifts the letter activations to the left (for a rightward saccade) the indicated number of positions and resets position unit 0 (at fixation) to be active. At this point the network tries again to pronounce the (now fixated) grapheme. Thus, for more difficult words and nonwords, and when the network is only moderately trained, it pronounces as much of the input as it can until it runs into trouble, then refixates to that part of the input and continues.

The only remaining issue is what to do when the network fails on the first grapheme of a string, or immediately after refixating, because in these cases there is no accurate position information to drive refixation. This is solved during training by using the target for the position units as the location of the next fixation.

When the fully-trained network is tested on an input string, its operation is slightly different. In this context, it cannot base its decision to refixate on whether its output is correct because it isn't provided with this information. Rather, it refixates when its output is not *clean*. An output is defined to be clean when exactly one phoneme unit and one position unit are above 0.8 and the remaining phoneme and position units are all below 0.2. If it fails to generate clean output at fixation, it simply includes the most active phoneme unit in its pronunciation, and uses the most active position unit to guide refixation.

To summarize, as the network is trained to produce the appropriate sequence of phonemes for a letter string, it is also trained to maintain a representation of its current position within the string. The network

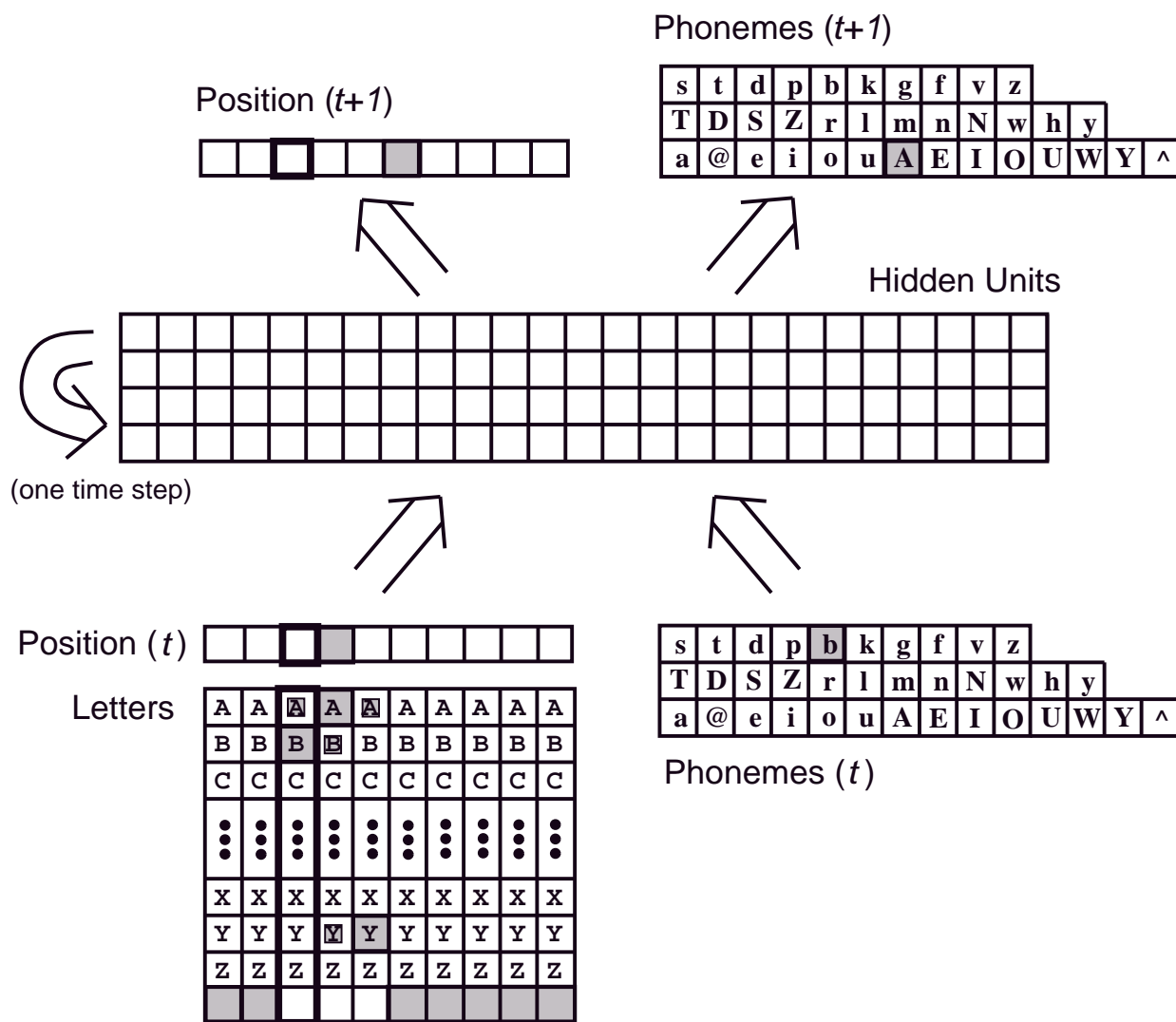


Figure 4: The network architecture for the refixation network. The arrows indicated full connectivity between groups of units. The recurrent connections among the hidden units only convey information about the last time step. The grey areas in the input and output units are intended to depict their activities at an intermediate point in processing the word BAY, after the B \Rightarrow /b/ has been pronounced (with no refixation) and the AY \Rightarrow /A/ is being attempted.

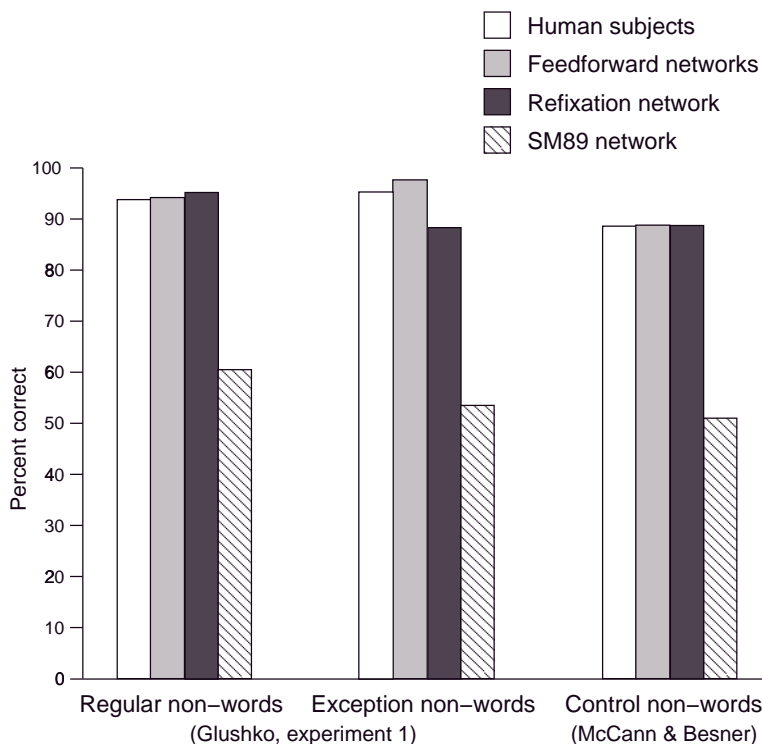


Figure 5: Percent correct on nonword reading for subjects and all networks, where a correct response is one that is regular or is consistent with the pronunciation of a word with the same body.

uses this position signal to refixate a peripheral portion of the input when it finds that portion difficult to pronounce. This repositions the input string so that the peripheral portion falls right at the point of fixation, where the network has had more experience in generating pronunciations. In this way, the network can apply the knowledge tied to the units at the point of fixation to any portion of the string that is difficult for the network to read.

The network was given extensive training on the extended SM89 corpus (900,000 word presentations, sampled according to frequency, with a learning rate of 0.01 and momentum of 0.9). Early on in training, the network requires multiple fixations to read words, but as the network becomes more competent it eventually reads most words in a single fixation. At the end of training, the network reads correctly all but 9 of the words (99.7%), with an average of 1.18 fixations per word.

When tested on nonwords, the network gives regular pronunciations to 71/80 (88.8%) of McCann and Besner's control nonwords, and 41/43 (95.3%) of Glushko's regular nonwords. Although only 26/43 (60.5%) of Glushko's exception nonwords are pronounced regularly, 12 of the 17 remaining pronunciations match some word in the training corpus, resulting in an overall level of performance of 38/43 (88.4%). Thus, while the refixation network shows a somewhat stronger tendency to give non-regular pronunciations to exception nonwords than does the feedforward network or human subjects, nonetheless the network demonstrates that good nonword reading can co-exist with good (exception) word reading in a network that must learn phonotactics itself. For comparison purposes, Figure 5 shows the percent of acceptable pronunciations given to the different sets of nonwords by subjects, by the feedforward and refixation networks, and by the SM89 network.

4 Conclusions

Coltheart et al. (1993) list several points beyond the ones considered here in support of their argument that two separate systems are needed to read pronounceable nonwords and exception words. The present article addresses only one of their arguments, and so we cannot claim to have produced a single route system that provides a complete account of reading of regular and exception words (although see Plaut et al., 1994, for additional simulations and discussion). However, we have addressed what we take to be the most central argument against a single mechanism account. Coltheart and colleagues state in their article that dual route theorists claim that there cannot be a single procedure that can correctly translate pseudowords and exception words from spelling to sound. We have shown that this claim is false.

References

- Besner, D., Twilley, L., McCann, R. S., & Seergobin, K. (1990). On the connection between connectionism and data: Are a few words necessary? *Psychological Review*, *97*(3), 432–446.
- Coltheart, M., Curtis, B., Atkins, P., & Haller, M. (1993). Models of reading aloud: Dual-route and parallel-distributed-processing approaches. *Psychological Review*, *100*(4), 589–608.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.
- Glushko, R. J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance*, *5*(4), 674–691.
- Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, *1*, 295–307.
- Kuçera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English*. Providence, RI: Brown University Press.
- McCann, R. S., & Besner, D. (1987). Reading pseudohomophones: Implications for models of pronunciation and the locus of the word-frequency effects in word naming. *Journal of Experimental Psychology: Human Perception and Performance*, *13*, 14–24.
- Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1994). *Understanding normal and impaired word reading: Computational principles in quasi-regular domains* (Technical Report PDP.CNS.94.5). Pittsburgh, PA: Department of Psychology, Carnegie Mellon University.
- Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96*, 523–568.
- Seidenberg, M. S., & McClelland, J. L. (1990). More words but still no lexicon: Reply to Besner et al. (1990). *Psychological Review*, *97*(3), 477–452.
- Seidenberg, M. S., Petersen, A., MacDonald, M. C., & Plaut, D. C. (in press). Pseudohomophone effects and models of word recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.
- Taraban, R., & McClelland, J. L. (1987). Conspiracy effects in word recognition. *Journal of Memory and Language*, *26*, 608–631.
- Venezky, R. L. (1970). *The structure of English orthography*. The Hague: Mouton.
- Waters, G. S., & Seidenberg, M. S. (1985). Spelling-sound effects in reading: Time course and decision criteria. *Memory and Cognition*, *13*, 557–572.
- Wickelgren, W. A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, *76*, 1–15.