Proceedings of 1993 International Joint Conference on Neural Networks

Neural Network Models of Reading Multi-Syllabic Words

John A. Bullinaria

Department of Psychology, University of Edinburgh
7 George Square , Edinburgh EH8 9JZ, U.K.

This paper presents the results of simulations of a new class of artificial neural network models of reading. Unlike previous models, they are not restricted to mono-syllabic words, require no complicated input-output representations such as Wickelfeatures and, although based on the NETtalk system of Sejnowski & Rosenberg (1987), require no pre-processing to align the letters and phonemes in the training data. The best cases are able to achieve 100% performance on the Seidenberg & McClelland (1989) training corpus, in excess of 90% on pronounceable non-words and on damage exhibit symptoms similar to acquired surface dyslexia.

Introduction

The world of cognitive science has recently enjoyed a lively debate concerning the processes underlying the act of reading aloud, i.e. the human act of converting strings of letters into strings of phonemes. One camp (recently championed by Coltheart, Curtis & Atkins, 1992) argues that the process can only be described by a dual route model, with one route consisting of a series of letters to phonemes rules (which are necessary in order to be able to read new words or pronounceable non-words) and another route that acts as a lexicon (necessary to deal with irregular/exception words which do not follow the rules). The other camp (as exemplified by Plaut, Seidenberg & McClelland, 1992) believe that a single route is sufficient and have constructed explicit neural network models of reading that are able to learn the words (including exception words) in their training data and also read new non-words with accuracies comparable to human subjects. This paper will describe another class of neural network models of reading and compare the performance of the simplest cases with that of previous models and humans.

The first thing to be decided for any model of reading is the representation to use for the inputs (letters) and outputs (phonemes). If there were a one-to-one correspondence between the letters and phonemes of every word, it would be fairly easy to set up a neural network to map from letter strings to phoneme strings. Unfortunately, however, the mapping is many-to-one (up to four letters can map to one phoneme in English, e.g. 'ough' $\rightarrow$ /O/ in 'though'), so more complicated models are necessary. We use the phoneme notation of Seidenberg & McClelland (1989) throughout.

One of the first successful neural network systems to get round this problem was NETtalk by Sejnowski & Rosenberg (1987) who simply pre-processed the training data by inserting special continuation (i.e. no output) characters into the phoneme strings to align the letters and phonemes. For many, this degree of pre-processing is considered unacceptable.

A more sophisticated model by Seidenberg & McClelland (1989) used a system of distributed Wickelfeatures in which each letter and phoneme string is split into sets of triples of characters (Rumelhart & McClelland, 1986). This certainly bypasses the problem of aligning the letters and phonemes, but makes the interpretation of the networks output difficult and presents difficulties in understanding the nature of the internal representations. This model is also restricted to mono-syllabic words and performs poorly on non-words.

A more recent neural network model by Plaut et al. (1992) uses 108 orthographic input units (one for each of the Venezky graphemes occurring in the initial consonant, vowel and final consonant clusters) and 57 phonological output units. This model does very well at learning the training data and at reading non-words but is still restricted to mono-syllabic words.

Meanwhile, Coltheart et al. (1992), as part of their dual route model, developed a rule based non-neural network system which had good success in reading non-words, but (by construction) was poor at reading the non-regular words in the original training set.

The class of neural network models presented in this paper might be considered to be a neural network implementation of this Grapheme Phoneme Conversion (GPC) Rule system of Coltheart et al. (1992). However, given that an exception word mapping can be thought of as a very low frequency high powered rule (i.e. a rule that is activated only for one specific word and over-rides all other potentially useful rules) such a model should be able to handle exceptional words as well. Regular words will be pronounced according to simple rules, exception words will be pronounced according to complicated special purpose rules (effectively a lexicon) that must over-rule the simpler rules. There will clearly be a continuous spectrum between these two classes of words and since there are very few (if any) 'exception' words that do not contain any regular features at all, the need for true lexical entries will be minimal. The success of the model depends on the network maximizing its use of simple rules whilst minimizing its use of special purpose rules. In this way, when presented with new words or non-words, none of the special purpose rules will fire and the network will output phonemes according to a full set of regular

(GPC) rules, yet it will still be able to pronounce the exceptional words it has been trained on.

The Models

The basic model consists of a standard fully connected feedforward network with one hidden layer set up in a similar manner to the NETtalk model of Sejnowski & Rosenberg (1987). The input layer consists of a window of *nchar* sets of units, each set consisting of one unit for each letter occurring in the training data (i.e. 26 for English). The output layer consists of one unit for each phoneme occurring in the training data (i.e. about 40 units). The input words slide through the input window which is *nchar* letters wide, starting with the first letter of the word at the central position of the window and ending with the final letter of the word at the central position. Each letter activates a single input unit. If there were a one-to-one correspondence between the letters and the phonemes, the activated output phoneme would then correspond to the letter occurring in the centre of the window. Since there can be a many-to-one correspondence between the letters and phonemes, some of the outputs must be blanks (i.e. no phoneme output). It is the problem of not knowing where to put the blanks in the training data that has hampered progress with this type of model in the past.

The solution proposed here is to allow the set of phonemes corresponding to each word in the training data to be padded out with blanks (to the same number of phonemes as there are letters in the word) in all possible ways. If there are nl letters and np phonemes, then there are $ntarg = nl\,!\,/\,np\,!\,(nl - np)\,!$ ways that the phonemes can be padded out. Clearly, we only want the network to train on one of these *ntarg* possible targets. The surprising thing is that by calculating for each input word the total error corresponding to each of the possible targets and only propagating back the error from the target with the least error, the network is (with a suitably diverse set of training words) able to learn which is the appropriate target for each word.

For example, consider the word 'ace' and the corresponding phonemes /As/. This training example will be presented $nl = 3$ times, each with $ntarg = 3$ possible target outputs:

presentation	inputs	target outputs
1.	- - - **a** c e -	A A -
2.	- - a **c** e - -	s - A
3.	- a c **e** - - -	- s s

For each of the three input presentations the error is calculated for each of the three target outputs. The sum of the errors for each target over the three input presentations is then computed and the target with the minimum total error is used to update the weights in the appropriate manner. With a realistic set of training patterns the regular correspondences will dominate and eventually the network learns that the appropriate target is /As-/ rather than /A-s/ or /-As/.

The networks were trained using the back-propagation algorithm (Rumelhart, Hinton & Williams, 1986) with the extended Seidenberg & McClelland training corpus of 2998 monosyllabic words consisting of the original Seidenberg & McClelland (1989) set plus 101 other words missing from that set. The small initial weights were chosen randomly with a rectangular distribution in the range –0.1 to 0.1. After some experimentation, the back-propagation learning rate was fixed at 0.05, the momentum factor at 0.9 and the sigmoid prime offset at 0.1. Over-learning was controlled by not propagating back the error signal for words that already had the correct phoneme outputs and a total error less than some threshold *errcrit*. Once trained the output phoneme of the network is simply defined to be the phoneme corresponding to the output unit with the highest activation.

Results from the Simulations

For each simulation, against the number of training epochs (on a logarithmic scale), were plotted the percentages learnt correctly and mean square errors for the full set of training data plus various interesting subsets (homographs, regular words, high/low frequency exceptions, etc.). To test the networks generalization ability (i.e. its success at learning the GPC rules) the percentages of three sets of non-words that were pronounced 'acceptably' and the corresponding errors were also plotted.

As in Plaut et al. (1992), the three sets of non-words used were the regular non-words and exception non-words of Glushko (1979, Experiment 1) and the control non-words of McCann & Besner (1987, Experiment 1). The allowable pronunciations of the non-words were derived from the training data-base by matching word segments (particularly rimes) in the non-words with the same segments in the training data and constructing possible non-word pronunciations by concatenating the pronunciations of the segments from the training data.

Due to the large amount of processing power required for these simulations, only 25 fairly small runs have so far been carried out and it is often difficult to distinguish real improvements caused by parameter or architecture changes from statistical fluctuations. A detailed analysis of the learning trajectories, how the models' performance varies with the window size, number of hidden units, *errcrit*, etc., what structures are actually being represented in the hidden units, the relationship between errors and naming latencies, developmental dyslexias and how the model responds to different types of damage will be presented in a necessarily longer paper elsewhere (Bullinaria, 1993). Here we will just make a few general comments and plot some of the results from one typical successful run.

Since the Seidenberg & McClelland corpus contains 13 pairs of homographs it is clear that the network can never achieve total success at learning

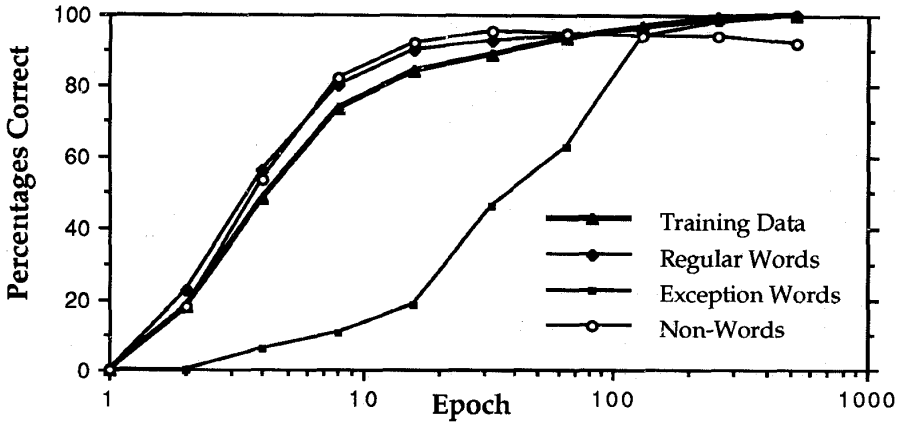

Figure 1. Typical Learning Curves.

this training data. Experiments were therefore carried out on the use of context flags to resolve these ambiguities. As a preliminary investigation, this was implemented by introducing an extra character into the input alphabet and appending that character to the least regular input word of each pair of homographs. This not only allowed the network to achieve 100% success rate on the homographs (compared with a maximum of 50% before) but also seemed to improve its performance on certain non-homographs as well. That such a simple flag works so well also gives us hope that similar flags could be used to flip the network between accents and languages as effortlessly as in humans.

Networks with a window size of 9 characters and as few as 40 hidden units were able to learn all but one of the training examples, namely 'though' → /DO/. The reason the network fails on this word is that the training data also includes the word 'thought' → /T*t/ in which the sub-word 'though' has to be pronounced as /T*/ and unless the input window is large enough to have the final 't' in the window while the initial 't' is in the centre of the window, the network has no way of resolving the ambiguity. By increasing the window size to 13 this long range dependency can be handled and the network achieves 100% success rate on its training data. To confirm the networks' capability of handling long range dependencies and also its ability to deal with more complex words, some runs with the words 'photographic' → /fotOgrafik/ and 'photography' → /fot*grafE/ incorporated into the training data were carried out. With a window size of 13 the network was able to learn both words without any difficulty. With a window size of 11 (for which the crucial 'i' and 'y' fall outside the window while the problematic 'o' is in the central position) the network failed to learn the two words.

Figure 1. shows the learning curves for a typical network with 120 hidden units, a window size of 13 characters, *errcrit* = 0.01 and a context flag to resolve homograph ambiguities. It achieved 100% performance on the training data and for non-words plateaued at 95.3% for regulars, 93.0% for exceptions and 92.5% for controls. Comparisons with other models are complicated by different authors using different non-word sets and scoring criteria, so bearing this in mind, the Seidenberg & McClelland model achieved 97.3% on the training data and about 65% on Glushko non-words, Plaut et al. (1992) achieved 99.9% and 97.7%, Coltheart et al. (1992) achieved about 77% and 98% and for human subjects we would typically have about 100% and 96%.

Although our networks generally performed fairly well, and many of the non-word errors would be acceptable under more generous criteria of acceptability (e.g. 'wuff' → /wuf/ is counted as wrong by the above rules), there still remain a few errors of the kind humans would never make (e.g. 'zute' → /hyt/). Whether these problems can be removed by the introduction of recurrent connections, clean up units, different learning algorithms/parameters is not clear at present. However, for small networks it was found that increasing the number of hidden units improved generalization. It was also noticed that if smaller training sets were used, then obviously incorrect grapheme-phoneme correspondences could be learnt (e.g. 'ace' → /-As/ instead of /As-/) without affecting the output performance on that training set. It is likely, therefore, that simply using larger networks and training sets could further improve our performance on non-words.

Discussion

An important method of constraining cognitive models is to examine their performance after damage (e.g., Coltheart et al., 1992). Of particular importance for models of reading are two forms of acquired dyslexia: Patients with phonological dyslexia exhibit a dissociation between word and non-word naming - there can be complete failure to read non-words whilst maintaining around 90% success on words (Funnell, 1983). Patients with surface dyslexia exhibit a dissociation between regular and exception word naming - for example 80% success on regular words against 35% on very irregular words (Shallice,

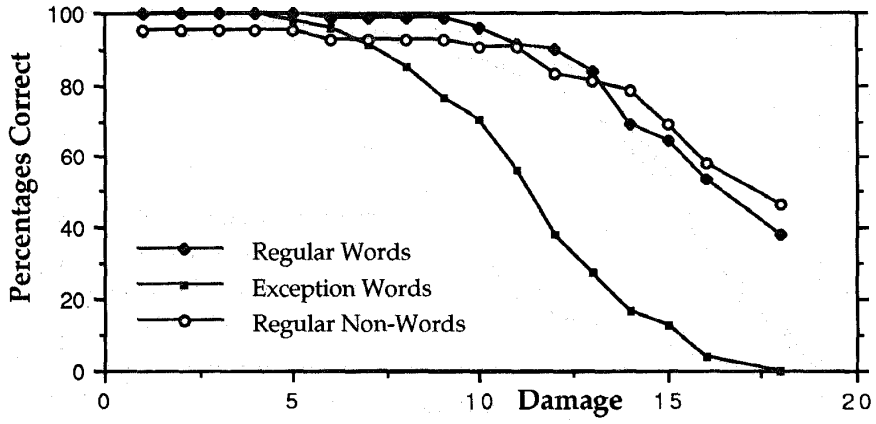

Figure 2. Typical Damage Curves.

Warrington & McCarthy, 1983).

Since no form of damage to our model seems to be able to produce the loss of rules (i.e. non-words) but not the words (nor even the high frequency exceptions) we should not consider it to be a realistic single route model. Phonological dyslexia must presumably still be explained by losing the GPC route, but not the lexical/semantic route, of a dual route model (Coltheart et al., 1992). It would seem appropriate, therefore, to consider our model to be an implementation of the GPC route of a dual route model. However, given our models inherent success with non-words, losing the lexical/semantic route but not the GPC route is not enough to explain surface dyslexia. We must, at the same time, lose the exceptions but not the rules in our GPC route. Fortunately, a form of damage whereby all the weights (i.e. synaptic strengths) are globally reduced in magnitude seems to do just that. In Figure 2. is plotted (for the same network as Figure 1.) the network performance as its weights are reduced by successive amounts of 0.05. We see that the exception words are preferentially lost over the regular words, so that at the point where the weights have been reduced by a total of 0.6 we have regular word performance at 90% compared with exception word performance at 37%. The exact percentages seem to vary somewhat with different network parameters, in particular the number of hidden units, but they are generally not far from those found in human patients.

In conclusion then, a class of neural network models have been presented which, given their simplicity, performance and room for improvement, seem to be promising candidates for the GPC route of a dual route model of reading.

References

Bullinaria, J.A., (1993), Representation, Learning, Generalization and Damage in Neural Network Models of Reading, in preparation.

Coltheart, M., Curtis, B. & Atkins, P., (1992), Models of Reading Aloud: Dual-Route and Parallel-Distributed-Processing Approaches, submitted to *Psychological Review*.

Funnell, E., (1983), Phonological processes in reading: new evidence from acquired dyslexia, *British Journal of Psychology*, **74**, 159-180.

Glushko, R.J., (1979), The Organization and Activation of Orthographic Knowledge in Reading Aloud, *Journal of Experimental Sciences: Human Perception and Performance*, **5**, 674-691.

McCann, R.S. & Besner, D., (1987), Reading Pseudohomophones: Implications for Models of Pronunciation Assembly and the Locus of Word-Frequency Effects in Naming, *Journal of Experimental Psychology: Human Perception and Performance*, **13**, 14-24.

Plaut, D.C., McClelland, J.L. & Seidenberg, M.S. (1992), Reading Exception Words and Pseudowords: Are Two Routes Really Necessary?, presented at the *Annual Meeting of the Psychonomic Society*, St. Louis, MO, November 1992.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J., (1986), Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing*, Volume 2 (eds. D.E. Rumelhart & J.L. McClelland) MIT Press, Cambridge, Mass.

Rumelhart, D.E. & McClelland, J.L., (1986), On learning the past tenses of English verbs, in *Parallel Distributed Processing*, Volume 2 (eds. D.E. Rumelhart & J.L. McClelland) MIT Press, Cambridge, Mass.

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

Sejnowski, T.J. & Rosenberg, C.R., (1987), Parallel Networks that Learn to Pronounce English Text, *Complex Systems*, **1**, 145-168.

Shallice, T., Warrington, E.K. & McCarthy, R., (1983), Reading without semantics, *Quarterly Journal of Experimental Psychology*, **35A**, 111-138.