In: *Proceedings of the 16th Annual Conference of the Cognitive Science Society,* pp. 84-89. Hillsdale, NJ: Erlbaum, 1994.

Internal Representations of a Connectionist Model of Reading Aloud

John A. Bullinaria
Department of Psychology, University of Edinburgh
7 George Square, Edinburgh EH8 9JZ, U.K.
johnbull@uk.ac.ed

Abstract

We use hierarchical cluster analysis, principal component analysis, multi-dimensional scaling and discriminant analysis to investigate the internal representations learnt by a recent connectionist model of reading aloud. The learning trajectories of these representations may help us understand reading development in children and the results of naming latency experiments in adults. Studying the effects of network damage on these representations seems to provide insight into the mechanisms underlying acquired surface dyslexia. The discussion of the various techniques used may also prove useful in analysing the functioning of other connectionist systems.

Introduction

Connectionist models (e.g. Seidenberg & McClelland, 1989) have begun to play an important role in the long running debate concerning the processes underlying the act of reading aloud (e.g. Coltheart et al., 1992). It was recently shown (Bullinaria, 1993, 1994) how the NETtalk model of reading (Sejnowski & Rosenberg, 1987) could be modified to work without the need for pre-processing of the training data to align the letters and phonemes prior to training. This modified model not only has superior learning and generalization performance than other models trained on the same words (e.g. Seidenberg & McClelland, 1989), but also has the advantage that it does not require the use of complicated input and output representations. Consequently, it has become feasible to analyse the internal representations of this model with view to better understanding the working of the model under normal conditions (e.g. has it learnt 'rules' or is it merely operating by analogy with particular words occurring in the training data) and after damage (e.g. can we model acquired dyslexias without the need for separate sub-systems for the regular and exception words).

There are numerous possible variations of the original NETtalk model discussed in Bullinaria (1994). The 'standard' extension to be investigated here is a fully connected feedforward network with sigmoidal activation functions and one hidden layer of 300 units. The input layer consists of a window of 13 sets of units, each set having one unit for each letter occurring in the training data (i.e. 26 for English). The output layer consists of two sets of units, each set having one unit for each phoneme occurring in the training data (i.e. 38 units). The network was trained using back-propagation on a standard set of 2998 monosyllabic words with the corresponding pronunciations - see Seidenberg & McClelland (1989) for details and notation. The input words slide through the input window, 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, with each letter activating a single input unit. The output phonemes correspond to the letter in the centre of the input window. Usually the output consists of one phoneme and one phonemic null (e.g. 't' → /t–/), occasionally it consists of two phonemes (e.g. 'x' → /ks/) and for silent letters we get two phonemic nulls (e.g. 'e' → /––/). The three possibilities cause the so-called *alignment problem* because it is not obvious from the training data how the letters and phonemes should line up. The advantage of this model over the original NETtalk is that, rather than doing the alignment by hand prior to training, a multi-target approach (Bullinaria, 1993) allows the network to *learn* the appropriate alignments during the training process. Given a word such as 'huge' → /hyUdZ/, the network considers all possible output target alignments (e.g. /hy Ud — Z–/) and trains only on the one that already gives the smallest total output activation error. Even if we start from random weights, the sensible regular alignments will tend to over-power the others, so that eventually the network settles down to using only the optimal set of alignments (e.g. /hy U– dZ —/).

This network achieved perfect performance on the training data (including many irregular words) and 98.8% on a standard set of 166 non-words used to test generalization. It also correlates well with various naming latency experiments and provides several possible accounts of developmental and acquired surface dyslexia.

Internal Representations

In this model, different regions of hidden unit activation space are selected by the output weights to produce different output phonemes and the learning process consists of judiciously choosing these regions and mapping the input letters to appropriate regions depending on the context information (i.e. surrounding letters). Since the consistent weight changes corresponding to regularities will tend to reinforce whereas others will tend to cancel, the network tends to learn the most regular mapping possible and hence we also get good generalization performance.

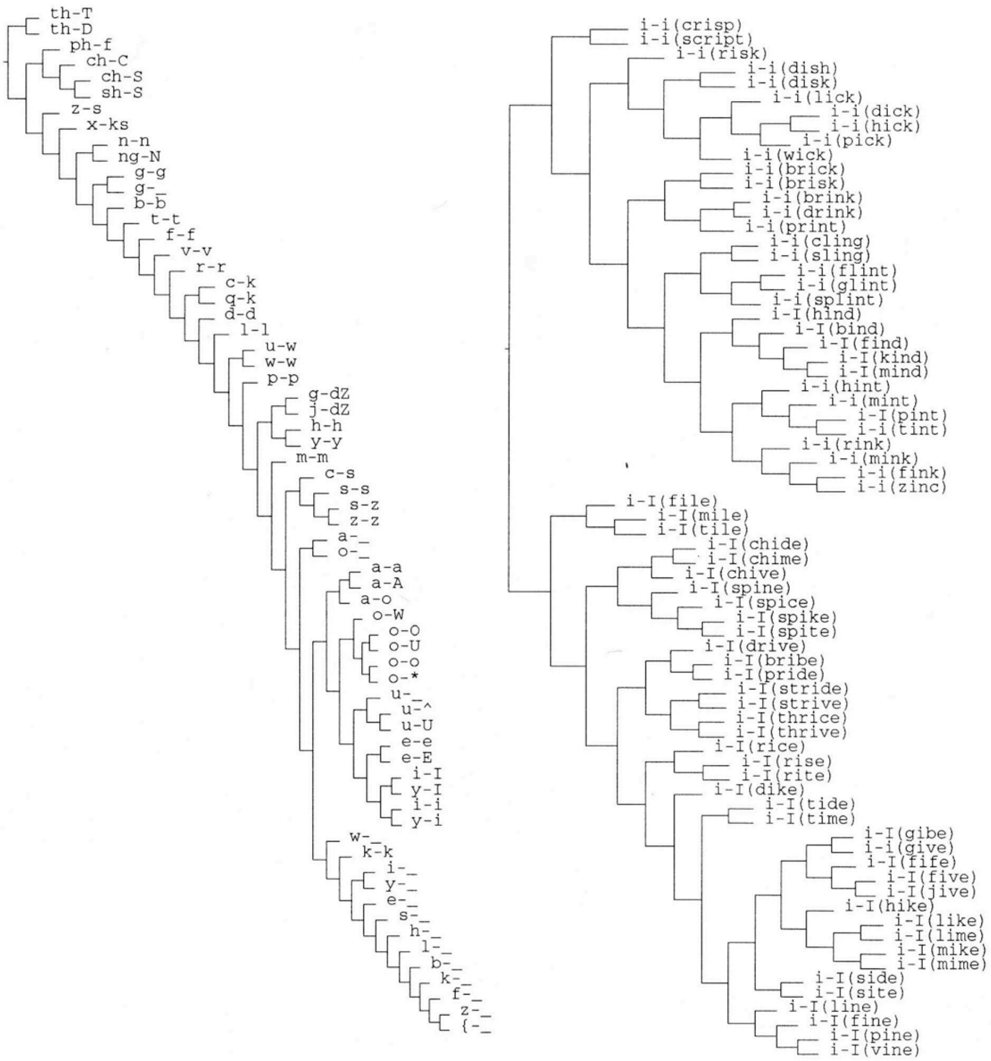

Figure 1: Hierarchical Cluster Analysis of (a) phoneme means, (b) sample words.

For our purposes, the main advantage of connectionist models over humans is that we can relatively easily examine their internal representations with view to understanding how they (and possibly humans) operate under normal and abnormal conditions. In the following sections we will consider several techniques that have previously been used to study the internal representations learnt by connectionist systems and then apply the best to various aspects of our model.

Hierarchical Cluster Analysis

One way to map out what is going on in hidden layer activation space is to perform a Hierarchical Cluster Analysis (HCA) of the points corresponding to each presentation of each word (Sejnowski & Rosenberg, 1987; Elman, 1989). Rather than trying to look at all the 12744 points representing our training data, we begin by looking at the mean activations for each of the main 65 letter to phoneme mappings. A simple Euclidean clustering gives Figure 1a and we get a similar picture using an L1 norm. The overall pattern is as expected - vowels together, silent letters together, consonants together and so on down to the likes of /dZ/ sounds together.

Figure 1b shows that the good clustering persists right down to the level of individual words. However, we see that irregular words (such as ‘give’ → /giv/ and ‘pint’ → /pInt/) are clustered with their regular counterparts (‘gibe’ → /gIb/ and ‘tint’ → /tint/) rather than the other words pronounced in the same way. Also, we find whole sub-rules (e.g. ‘ind’ → /Ind/) apparently in the wrong high level cluster. Since the network itself doesn't make use of Euclidean (or other) distance measures on the hidden unit space it is not surprising that HCA can lead to slightly misleading results. Indeed, all the distances are very similar here (mean 4.0, s.d. 0.6 for Figure 1a; mean 3.8, s.d. 0.8 for Figure 1b), so clustering doesn't make much sense anyway. The networks' output weights work by projecting out particular sub-spaces of the hidden unit space, so to get a better understanding of the internal representations we really

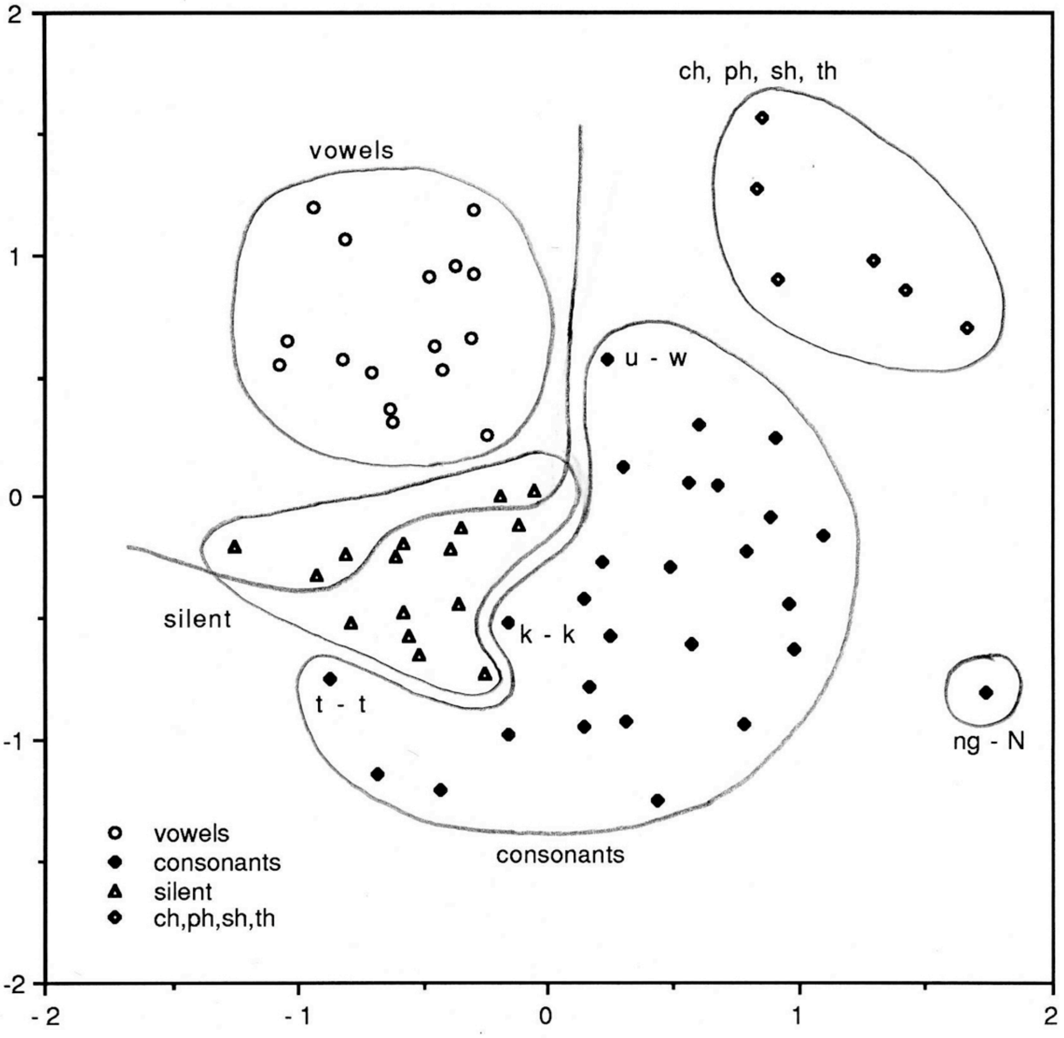

Figure 2: MDS plot of mean phoneme positions in hidden unit activation space.

need to see more accurately how the words are positioned in hidden unit activation space.

Principal Component Analysis

For our model we deliberately chose to use a large number of hidden units (i.e. 300), about ten times as many as actually needed to learn our training data. The reason for this was that we were particularly interested in modelling the effects of brain damage and acquired dyslexia. To do this realistically we needed a system that was fairly resilient and degraded gracefully when damaged. This required a highly distributed internal representation for which the removal of any single hidden unit or connection has very little effect on the network′s performance.

We succeeded in this aim, but we now have the difficult problem of visualising points in a 300 dimensional space. We clearly need to reduce the number of dimensions to something more manageable, i.e. two or three. The easiest way to do this is to use Principal Component Analysis (PCA): We change the coordinate basis to diagonalise the covariance matrix S and then restrict ourselves to the dimensions which carry the most variance. This approach was used to good effect by Elman (1989), but in our network the variance is distributed over too many components (the first three normalized eigenvalues for the full set of training data are 0.096, 0.078, 0.067). Taking any two or three components on their own gives a very poor representation of what’s happening.

Multi-Dimensional Scaling

A useful non-metric approach to dimensional reduction is provided by Multi-Dimensional Scaling (MDS). A gradient descent algorithm is used to adjust iteratively the positions of the points in a low dimensional space until the rank order of the inter-point distances correspond as closely as possible to those in the original space (Kruskal, 1964).

For small numbers of points, MDS works quite well. The average phoneme data of Figure 1a resulted in the two dimensional MDS plot shown in Figure 2. The correlation with the original data is 0.82 compared with 0.50 for a 1D plot, 0.88 for a 3D plot and 0.56 for the first two principal components. We can see more clearly how the points are clustered and understand anomalies in the HCA, e.g. why the 'k-k' point was grouped with the silent letters. We can also get good plots for the individual words, with the exception words and sub-rules appearing at the edges of clusters as close as possible to the clusters of their regular counterparts. However, for larger numbers of points the correlations become weaker and often words that we know from cluster analysis should be close together do not appear together on the MDS plots. In these cases it is clearly dangerous to make detailed predictions from MDS plots, since it is difficult to know which lost information is responsible for the breakdown in correlation. What we really need is a procedure for plotting true distances.

Canonical Discriminant Analysis

Clearly, we cannot represent the whole of our network's internal representation in two dimensions. What should be feasible and more useful, however, is to plot a small subset of the representation, for example the distinction between the long and short 'i' sounds (/I/ and /i/). This approach, in the form of Canonical Discriminant Analysis (CDA), was successfully applied by Wiles & Ollila (1992) to study combinatorial structure in hidden unit space. If we know which of g groups each point in hidden unit space belongs to (i.e. which output phoneme it corresponds to) we can partition the total covariance matrix $S = B + W$ into the between groups covariance B and the within groups covariance W. Then, by solving the eigenvalue problem for S^{-1} B, we obtain a matrix M which projects our space onto a rank $B \leq g - 1$ dimensional subspace that maximises the ratio $|B| / |S|$, i.e. clusters the points into groups with the maximum between group separations and minimum within groups dispersion.

Since we know our network performs a similar clustering (Gallinari et al., 1991), it is tempting to assume that this procedure will give a good representation of what is happening in hidden unit space. Consider our /I/ versus /i/ case again. We can separate the words into two groups and use CDA to obtain a projection vector in hidden unit space that best discriminates between the long and short sounds. The quality of the discrimination depends on the number of data points we use. If we have many less points than the number of hidden units, then the discrimination is essentially perfect (B/S = 1.0000 for 160 points). In fact, we get equally good discrimination even if we assign the points to groups at random (B/S = 1.0000). It is clear that we are not getting a good picture of the true internal representation. If we use all the points in the training data (239 /I/'s and 272 /i/'s) we do better : B/S = 0.98 for the true groups and B/S = 0.61 for random groups (and, as we should expect, for random groups we fail to get good clusters at all and have many overlaps). However, testing the projection on new non-words fails to classify them properly (even when the network itself does). To define the projection more accurately we clearly need more data points, particularly for the borderline region between the two groups. To this end we generated a set of 14766 words and non-words of the form 'C_1 V C_2' and 'C_1 V C_2 e' where C_1 was one of a set of 58 initial consonant clusters, V was one of the set {i, ia, ie, y} and C_2 was one of a set of 58 final consonant clusters. The CDA gave us a projection with B/S = 0.83, but there was a large overlap between the two groups: $\max_i = -0.15$, $\min_i = -0.38$, $\max_I = -0.23$, $\min_I = -0.46$ with 2454 /I/ words greater than $\min_i$ and 3828 /i/ words less than $\max_I$.

It is clear that our CDA is *not* giving us a good representation of the true internal representation. When the network learns, it certainly maximizes the between group distances $\min_i - \max_I$, but it has no need to minimize the within group dispersions. A simple iterative gradient descent procedure was employed to adjust the projection vector so that it correctly classified all 14766 data points – this gave B/S = 0.73. Unfortunately this was still not good enough. For a good representation, we would expect the borderline cases in the projections to correspond to borderline output phonemes - in fact, the correlation was very poor. Moreover, the same iterative procedure even managed to find a projection vector that could classify the set of training data points into our assigned random groups (B/S = 0.49).

Output Weight Projections

Actually, for our simple one hidden layer architecture, it is easy to make the projections correlate with the outputs: We just use the projections the network itself has learnt – namely the output weights. If we project the hidden unit activations using the output weights $w_o(h,p)$ and redefine the zero points using the output thresholds $\theta_o(p)$, our projection is then simply the output before passing through the sigmoid. We are guaranteed correlation. We thus have a suitable projection vector for each phoneme and these 38 vectors turn out to be nearly orthogonal (mean angle 84°, s.d. 5°). These can be viewed in pairs to examine the relation-ships between the clusters, or the above techniques may be used to study any interesting sub-spaces of this 38 dimensional space (e.g. the four dimensional /i/, /I/, /e/, /E/ subspace may be studied to investigate the various pronunciations of 'ie'). Figure 3 shows the resultant discrimination for our /I/ and /i/ phonemes. Projection on to the /i/ – /I/ diagonal gives us B/S = 0.72. On this graph, points corresponding to other phonemes would appear in the bottom left quadrant. Each point has a positive projection onto the line in hidden unit space corresponding to that phoneme and a negative projection onto the lines corresponding to all the other phonemes. Plotting trajectories on such graphs may help us understand both developmental effects (e.g. developmental dyslexia) and how the network performs after damage.

Learning Trajectories

With small random initial weights, all points start near $A(p) = {}^1/_2 \sum_h w_o(h,p) - \theta_o(p)$ and step towards the appropriate quadrant. There are several effects that determine the final position of each word presentation. First, as is clear from our HCA and MDS plots, similar

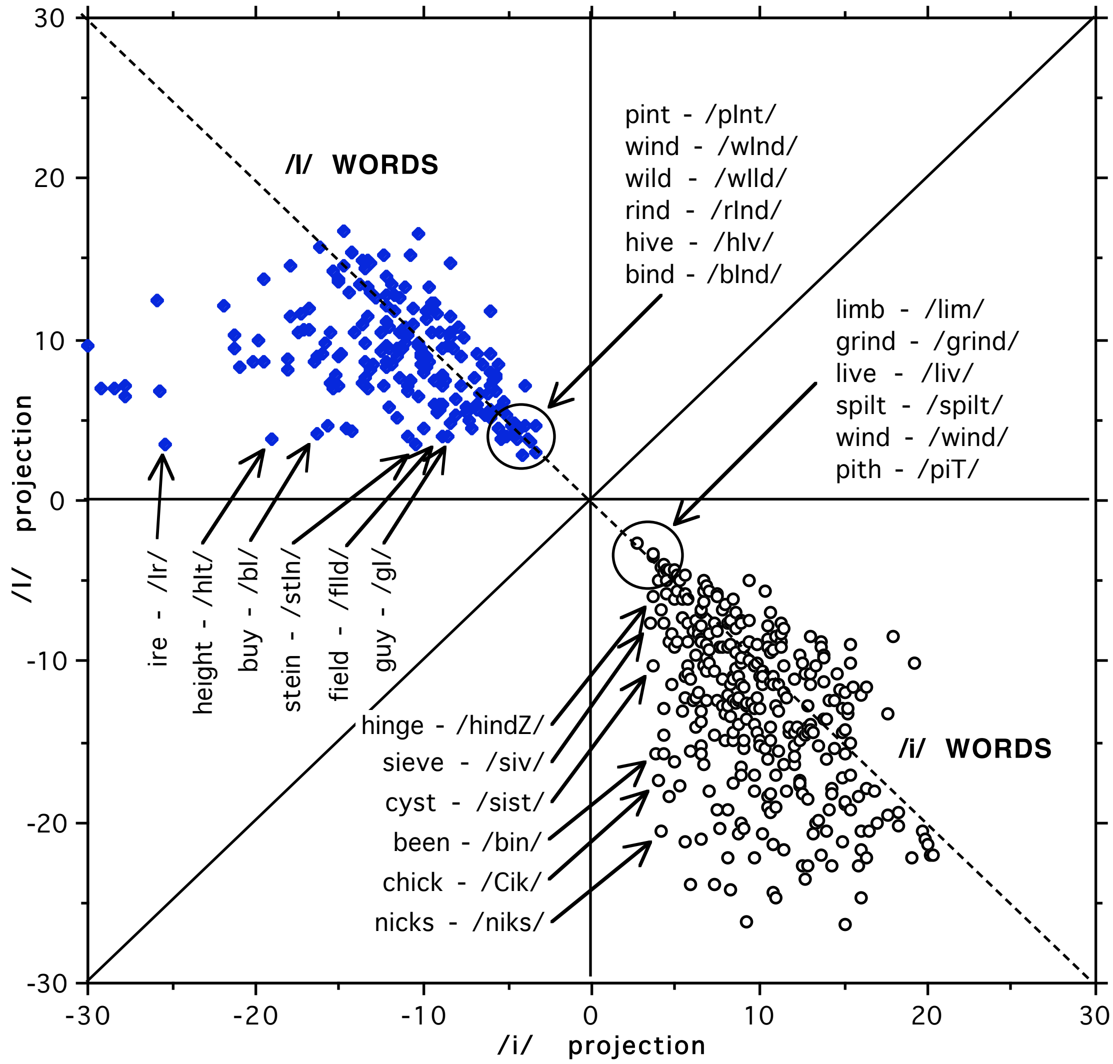

Figure 3: The 2D hidden unit sub-space corresponding to the /i/ and /I/ phonemes.

words will tend to follow similar trajectories and end up in similar regions of hidden unit space. High frequency words of all types will tend to have had time to get well into the right quadrant. The positions for the lower frequency words will be more variable. Words containing no ambiguity will head directly to the correct quadrants. Ambiguous phonemes in exception words and closely related regular words (often referred to as regular inconsistent words), will be pulled towards two (or more) different quadrants with strengths proportional to their relative frequencies. Although the network eventually learns to use the context information to resolve these ambiguities, these points will still be the last to cross into the right segments and be the ones left closest to the axes. Strange words (e.g. 'sieve'), that have very rare spelling patterns, may also be left near the axes depending on their word frequency.

It has been argued that there should be a correlation between network output error scores and naming latencies in humans (Seidenberg & McClelland, 1989). Thus, since the closeness of each point to the axes is a measure of the output error score we can read off from our graphs the model's predictions for naming latency experiments: High frequency words will not show a type effect, low frequency exception words will be slower than regular inconsistent words which will be slower than consistent regular words and strange words will also have an increased latency effect. These predictions turn out to be fairly accurate, (for a detailed discussion see Bullinaria, 1994).

Damage Trajectories

The main reason for wanting to investigate the internal representations was to gain insight into how various forms of acquired dyslexia may occur in the model. Connectionist models that can deal with regular and exception words in a single system have cast doubt on the traditional dual route models of reading with their separate phonemic and semantic routes. However, a minimum requirement for them to replace the dual route model completely is for them to be able to exhibit both surface dyslexia (lost exceptions) and phonological dyslexia (lost non-words) when damaged appropriately (e.g. Coltheart et al., 1992). We will consider six forms of damage and examine the kinds of output errors they produce (see Bullinaria, 1994, for more details). In each case the degree of damage is increased from zero to a level where the network fails to produce any correct outputs at all. Patients with varying degrees of dyslexia will correspond to particular intermediate stages.

1. Weight Scaling. The simplest form of damage we can inflict is to scale all the weights and thresholds by a constant scale factor $0 < \alpha < 1$. The effect of decreasing α is to flatten all the sigmoids and, since the winning output phoneme is independent of the flatness of the output sigmoids, all the effect can be seen at the hidden units. As the hidden unit sigmoids are flattened, all the hidden unit activations tend to 0.5 and all the projections head back to the A(p) defined above. It turns out that all the A(p) are large and negative (mean -50.3, s.d. 10.2) so all the points drift more or less parallel to the bottom left diagonal. The flow is fairly laminar, so the first points to cross the phoneme borders tend to be those that started off nearest to the borders. Thus the errors are predominantly on low frequency exceptions rather than regular words and the errors for small amounts of damage tend to be regularisations. This is precisely the pattern of errors commonly found in surface dyslexics.

2. Weight Reduction. Reducing each weight by constant amounts also causes each point to head for A(p). The flow is less laminar than with scaling, and the output layer also gets affected, but there is still a strong tendency for the borderline cases to cross over first. Again we get symptoms similar to surface dyslexia.

3. Adding Noise. We can consider adding Gaussian noise to all the weights. In the extreme limit all the weights become random, the hidden unit activations tend to 0.5 and again each point tends towards A(p). The random walk we produce by adding more and more noise is not laminar but on average more borderline points have crossed over than others, so again we model surface dyslexia.

4. Removing Connections. As connections are removed at random, the hidden unit activations will tend towards the sigmoids of the thresholds. In these networks we find the hidden unit thresholds are predominantly negative (mean -1.3, s.d. 0.7) so most of the activations end up in the range ~0.12 to ~0.38. The points in hidden unit space thus end up scattered around ~A(p)/2. Again the flow is into the bottom left quadrant and again the random walk will lead to surface dyslexic effects.

5. Removing Hidden Units. As the hidden units are removed, the projections tend towards the output thresholds (mean 0.9, s.d. 1.4). For large numbers of hidden units we can expect a reasonably ordered drift with the borderline cases crossing first, otherwise we get a random walk with similar effects. We thus have more surface dyslexia.

6. Weight Clipping. To show that damage can occur without resulting in surface dyslexia we consider weight clipping, i.e. imposing a maximum value that the weights may take. The effect of clipping depends crucially on the input units activated and varies radically between different words and phonemes. The flow in hidden unit space, although again towards A(p), is extremely non-uniform and words that begin nearest the borderlines are not necessarily the ones that cross first. Consequently we get no rule/exception effect.

Discussion

We have seen that using output weight projections is the best way to view the internal representations of our reading model and that several other traditional techniques (such as HCA, PCA, MDS and CDA) cannot be reliably used. Similar conclusions can be expected even if we replace the moving window system by a more psychologically plausible system of recurrent connections (e.g. Jordan, 1986). We can see how the developmental and naming latency effects arise. We can also understand how five different kinds of damage all lead to symptoms similar to surface dyslexia. Moreover, it is reasonable to expect that, as we increase the number of hidden units, the different kinds will better approximate to the cleanest example of weight scaling. However, there appears to be no way for the model to exhibit acquired phonological dyslexia - so the dual route model is not dead yet!

References

Bullinaria, J.A. (1993). Neural Network Models of Reading Without Wickelfeatures. *Proceedings of the 2nd Neural Computation and Psychology Workshop*, Edinburgh.

Bullinaria, J.A. (1994). Representation, Learning, Generalization and Damage in Neural Network Models of Reading Aloud. Submitted to *Psychological Review*.

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

Elman, J. L. (1989). Representation and structure in connectionist models. CRL Technical Report 8903, University of California, San Diego.

Gallinari, P., Thiria, S., Badran, F. & Fogelman-Soulie, F. (1991). On the Relations Between Discriminant Analysis and Multilayer Perceptrons. *Neural Networks*, **4**, 349-360.

Jordan, M.I. (1986). Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 531-536). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kruskal, J.B. (1964). Multidimensional scaling by optimizing goodness to fit to a non-metric hypothesis. *Psychometrika*, **29**, 1-27.

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.

Wiles, J. & Ollila, M. (1992). Intersecting Regions: The key to combinatorial structure in hidden unit space. In S. J. Hanson, J. D. Cowan & C. L. Giles (Eds.) *Advances in Neural Information Processing Systems 5* (pp. 27-33). San Mateo, CA: Morgan Kauffmann.