Analyzing the Internal Representations of Trained Neural Networks

John A. Bullinaria
Centre for Speech and Language, Department of Psychology
Birkbeck College, Malet Street, London WC1E 7HX, UK
j.bullinaria@physics.org

1 Introduction

Although it may sometimes be sufficient to view a trained neural network as a 'black box' that solves or performs a particular task, we would often like to have some idea of what exactly the network is doing. It is clearly something of a problem not to understand how a network operates when it has been designed as a model of some cognitive function. It is equally problematic to have artificial systems solving real world problems in ways that are not easily understandable by their human creators. Such problems range from matters of legality (e.g. in some countries it is illegal to refuse credit without giving a reason and 'because our neural network said so' is not generally considered to be a good enough reason) to matters of reliability (e.g. in safety critical applications, can we trust a system whose working we do not fully understand?).

In this chapter we shall be concerned with analysing the internal representations that are learnt by simple feedforward networks and using the resultant information to understand how the networks are operating. This will also allow us to investigate how the networks are likely to respond to damage, which is important both from the point of view of neuropsychological modelling and for the more practical problem of estimating the robustness of implemented real world systems. A useful introduction to this area of study is provided by Hanson and Burr (1990).

We shall begin by reviewing the traditional techniques of hierarchical cluster analysis, principal component analysis, multi-dimensional scaling and discriminant analysis and illustrate their use by investigating the internal representations learnt by a recent connectionist model of reading aloud. We shall see that there are limitations to all these approaches and then show how simple output weight projections may lead to a clearer picture of what is happening. In the reading model, the learning trajectories of these projections may help us understand reading development in children and the results of naming latency experiments in adults. We shall then discuss contribution analysis and its use in predicting the effect of various types of damage to the networks. In our reading model, studying the effects of network damage seems to provide insight into the mechanisms underlying acquired surface dyslexia.

Before embarking on our general analysis of internal network representations, we first describe the network model that we shall use to illustrate our analysis. The NETtalk (Sejnowski and Rosenberg, 1987) model of reading aloud (i.e. text to phoneme conversion) was one of the earliest applications made possible by the re-invention of training techniques suitable for multi-layer neural networks (Rumelhart et al., 1986). It has recently been shown (Bullinaria, 1994a, 1995a) how this original NETtalk model can 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 to earlier reading models trained on the same words (e.g. Seidenberg and 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 how it operates under normal conditions and after damage.

Numerous possible variations of the original NETtalk model were discussed in Bullinaria (1994a). We shall only be concerned here with a fairly standard version consisting of a fully connected simple feedforward network with sigmoidal activation functions and one hidden layer of 300 units (as shown in Figure 1). 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[1]. 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 in the context of the other letters in the input window. Usually the output consists of one phoneme and one phonemic null (e.g. 't' → /t_/ in 'hot'), occasionally it consists of two phonemes (e.g. 'x' → /ks/ in 'box') and for silent letters we get two phonemic nulls (e.g. 'e' → /__/ in 'cake'). The fact that we have three possibilities causes the so-called *alignment problem*, because it is not obvious from individual words in 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, 1995a) allows the network to *learn* the appropriate alignments during the training process.

[1] We use the phonemic notation and conventions and words of Seidenberg and McClelland (1989) throughout this chapter. Apart from the standard consonants, this has: D = 'th' in 'than', T = 'th' in 'thin', S = 'sh' in 'shot', C = 'ch' in 'chat', N = 'ng' in 'fang', Z = 'z' in 'azure', a = 'a' in 'hat', A = 'ai' in 'mail, e = 'e' in 'met', E = 'ee' in 'deed', i = 'i' in 'hit', I = 'i' in 'pint', o = 'o' in 'hot', O = 'oa' in 'goal', ∧= 'u' in 'hut', U = 'oo' in 'boot', u = 'oo' in 'book', * = 'aw' in 'saw', W = 'ow' in 'cow'.

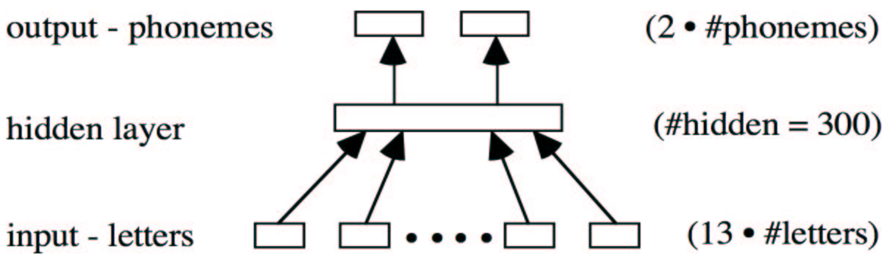

Figure 1: The NETtalk style model we shall use to illustrate our discussion.

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, given a sufficiently representative training set, the sensible regular alignments will tend to over-power the others, so the network eventually settles down to using only the optimal set of alignments (e.g. /hy U_ dZ __/). Once trained, this network achieves 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 provides simulated reaction times that correlate well with various naming latency experiments and allows several possible accounts of developmental and acquired surface dyslexia. Most important for current purposes, however, is that we have a simple network successfully trained to perform a relatively complex mapping.

2 Analysis techniques

In the reading model, different regions of hidden unit activation space are selected by the output weights to activate different output phonemes. The learning process consists of judiciously choosing these regions and mapping from each central input letter to an appropriate region 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. This manifests itself here, and in other multi-layer feed-forward networks, in the formation of well structured internal representations, i.e. patterns of hidden unit activation. Each network input pattern maps onto a particular point in hidden unit activation space and by understanding how these points stand in relation to each other we can gain an understanding of how the network is operating. In this section we shall critically review several techniques that have previously been employed to study the internal representations learnt by connectionist systems and illustrate their use with our reading model. We shall also look at some less traditional techniques which appear to cast more light on what is happening.

2.1 Hierarchical cluster cnalysis

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 input pattern (e.g. Everitt, 1975). The basic idea of HCA is that we define some distance measure on the hidden layer activation space (such as simple Euclidean distance) and then construct a hierarchy of clusters of points based on that measure. If the network is operating efficiently, we can expect input-output patterns to be more closely clustered when they are more related. This approach was found to result in sensible clustering of the letter-to-phoneme correspondences in the original NETtalk model (Sejnowski and Rosenberg, 1987) and also of the lexical categories in a simple sentence prediction network (Elman, 1990). We might therefore expect this approach to identify useful relations in other situations where they might not be so obvious.

Rather than attempting to look at all the 12744 points representing the training data of our reading model (given by 2998 words with an average of 4.25 letters per word), we begin by looking at the mean activations for each of the main 65 letter to phoneme mappings. A simple Euclidean clustering results in Figure 2 and we obtain a similar picture using an L1 norm. The overall pattern is largely as one might expect: vowels together, silent letters together, consonants together and so on down to the likes of /dZ/ sounds together; though there are a few anomalies (such as 'k' → /k/ being grouped with the silent letters) that we shall discuss later. It is also worth noting that the words are sometimes initially clustered according to their input letters (e.g. the 'a' → /o/ instances are clustered with the other 'a' instances rather than the 'o' → /o/ instances) and sometimes according to their output phonemes (e.g. the 'i' and 'y' words are clustered according to the output /i/ and /I/ sounds).

Such large scale clustering is interesting, but we also need to check that the good clustering persists right down to the level of individual words. To illustrate this, Figure 3 shows the clustering of a representative set of 72 instances of the single vowel 'i' (and the words containing them). We see that there appears to be a clear distinction between the long /I/ sound and the short /i/ sound. However, a closer inspection shows that the irregular words (such as 'give' → /giv/ and 'pint' → /pInt/) are clustered with their regular counterparts ('gibe' → /dZIb/ and 'tint' → /tint/) rather than with 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. It seems that the HCA is representing the well known linguistic rule that a final 'e' lengthens the preceding vowel, but it is not picking up the fact that the network has also managed to learn the exceptions to that rule. In many situations the whole object is for the network to identify any regularities in the training data and to ignore the exceptions (that commonly constitute 'noise'), so HCA may well still be a useful tool in these circumstances, but it gives a misleading picture of the networks' performance.

Another potential problem we face is that some parts of the training data may cluster better than others. Figure 4 shows the cluster plot for 68 instances of the letters 'o' or 'a' in our reading model plus the relevant eight mean mapping

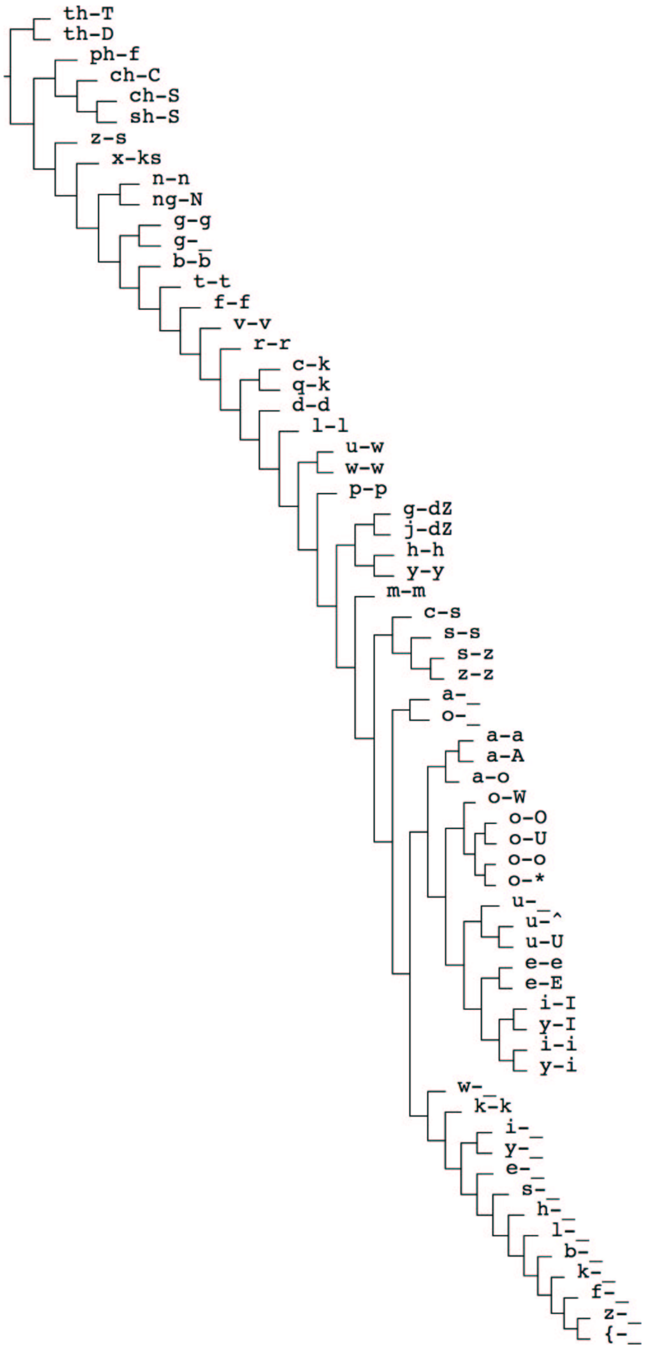

Figure 2: HCA of the mean letter-phoneme points in hidden unit space.

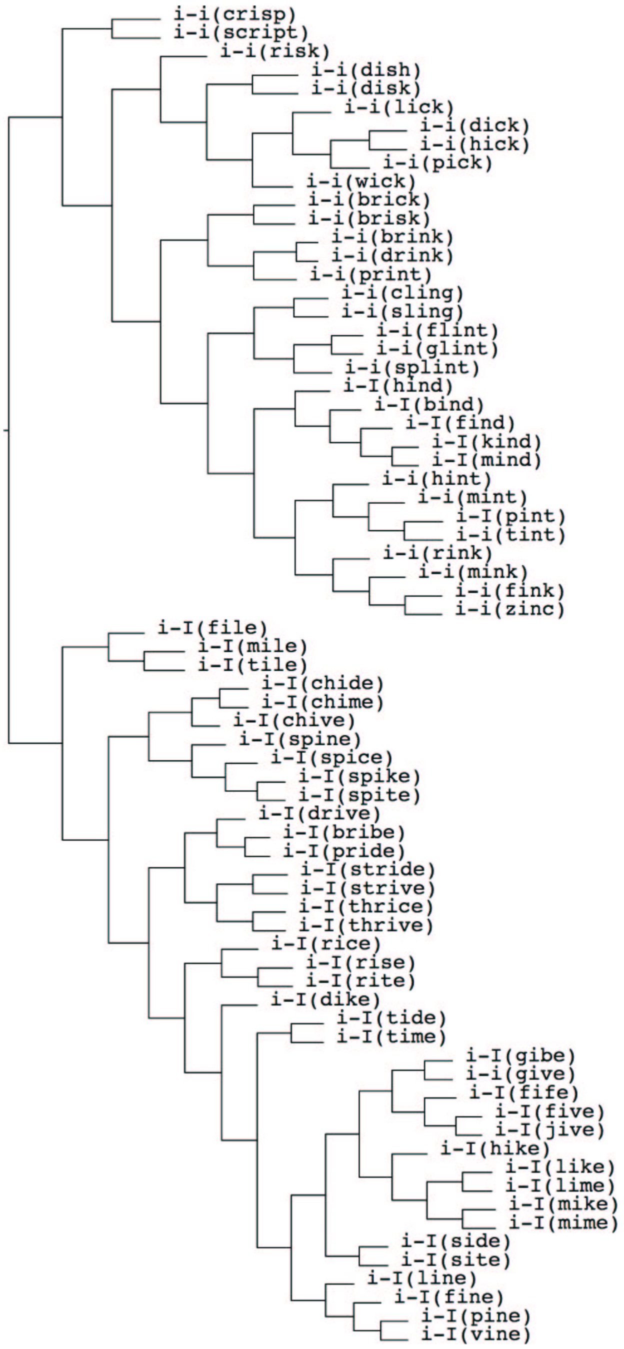

Figure 3: HCA of representative ‘i’ words in hidden unit space.

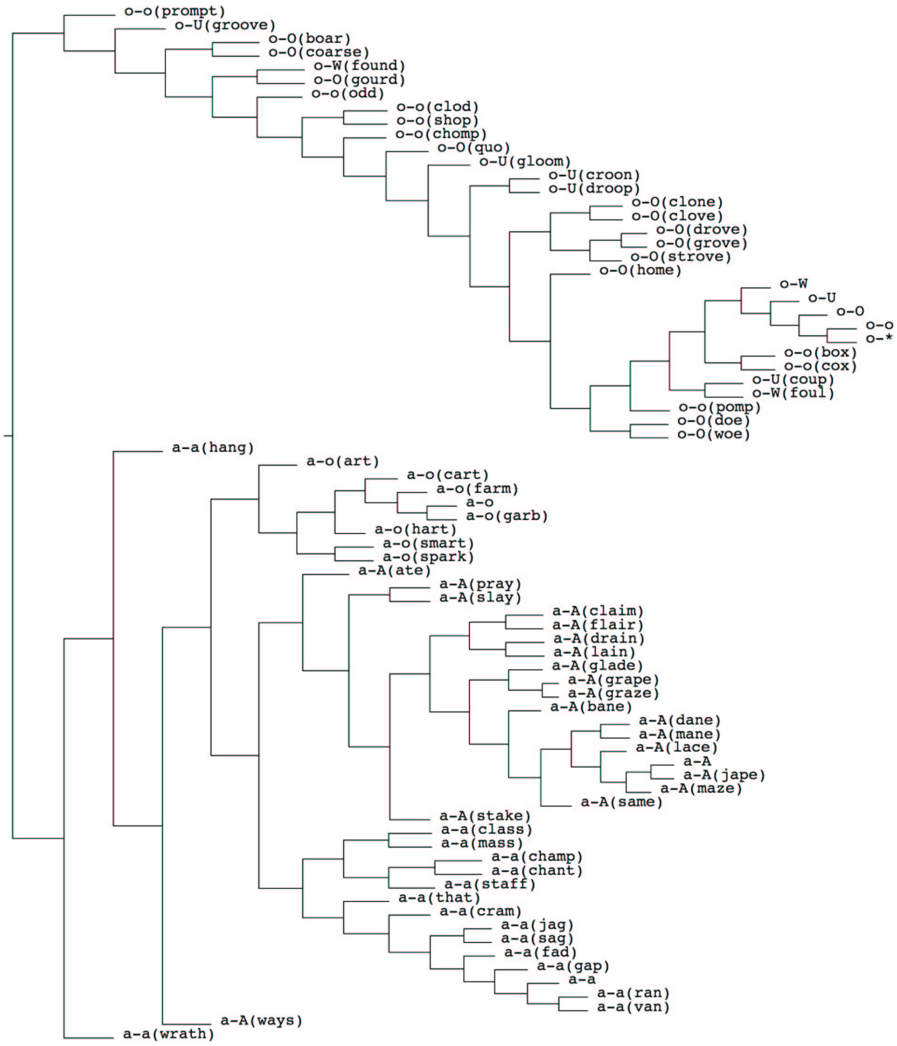

Figure 4: HCA of representative 'a' and 'o' words and their letter-phoneme means.

points from Figure 2. We have a clear distinction between the 'o' and 'a' words, and the 'a' words split reasonably well into the /o/, /A/ and /a/ clusters with the mean values fairly central to each cluster. The 'o' words, however, show poor clustering with the means apparently closer to each other than they are to the words they represent.

The above examples illustrate what the HCA can miss and consequently, even when the clustering appears to make sense, we must be careful about what conclusions we draw from such an analysis. Since the network itself doesn't make use of Euclidean (or other) distance measures on the hidden unit activation space, it is not surprising that HCA can sometimes produce slightly misleading results. Indeed, all the inter-point distances are actually very similar in the examples shown above (mean 4.0, s.d. 0.6 for Figure 2; mean 3.8, s.d. 0.8 for Figure 3; mean 4.4, s.d. 0.9 for Figure 4), so clustering doesn't make much sense anyway. Given a large enough dimensional space, the points will tend to spread themselves out as uniformly as they can. Reducing the number of hidden units closer to the minimum number required to learn the mapping may help slightly, but even if we train our reading network with only 30 hidden units we still find the exception words falling in the wrong clusters.

Since the networks' output weights operate by projecting out particular subspaces of the hidden unit activation space, to get a better understanding of the internal representations we really need to see more directly how the words are positioned in the hidden unit activation space.

2.2 Principal component analysis

For the reading 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 the 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 had very little effect on the network's performance.

We succeeded in this aim, but are now left facing the difficult problem of visualising points in a 300 dimensional space. Even if we had used a more minimal network, with only around 30 hidden units, we would still have far too many dimensions to visualise easily. We clearly need to reduce the number of dimensions to something more manageable, i.e. two or three. One conventional way to do this is to use Principal Component Analysis (PCA). If $\{P_{i\alpha} : i = 1, \ldots, d;\ \alpha = 1, \ldots, n\}$ are the vector components of a set of n points in our d dimensional hidden unit activation space and $\langle P_i \rangle$ denotes the mean $P_{i\alpha}$ over all values of α, then the standard covariance matrix S_{ij} is defined by

$$S_{ij} = \sum_{\alpha} (P_{i\alpha} - \langle P_i \rangle)(P_{j\alpha} - \langle P_j \rangle) \tag{1}$$

It then follows that, since S_{ij} is symmetric, the matrix Λ_{ij} of its eigenvectors given by

$$\sum_k S_{jk} \Lambda_{ki} = \lambda_i \Lambda_{ji} \tag{2}$$

is orthogonal. This means that Λ_{ij} can be used to perform a change of basis, i.e. an axis rotation, as given by

$$P^{\Lambda}_{i\alpha} = \sum_j \Lambda^{-1}_{ij} P_{j\alpha} \tag{3}$$

such that the covariance matrix is diagonalised as in

$$S^{\Lambda}_{il} = \sum_j \sum_k \Lambda^{-1}_{ij} S_{jk} \Lambda_{kl} = \lambda_i I_{il} \tag{4}$$

and the total variance is unchanged and given by

$$var(P) = trace(S) = trace(\Lambda S^{\Lambda} \Lambda^{-1}) = trace(S^{\Lambda}) = \sum_i \lambda_i \tag{5}$$

The p new coordinates $\{P^{\Lambda}_{i\alpha} : i = 1, \ldots, p;\ \alpha = 1, \ldots, n\}$ corresponding to the p largest eigenvalues λ_i are called the first p principal components and provide the best possible account for the variance in p dimensions.

PCA thus provides a convenient procedure for dimensional reduction with the minimum loss of information. We simply project our points onto the p dimensional sub-space spanned by the first p eigenvectors of the covariance matrix. This approach was used to good effect by Elman (1993) to analyse his sentence processing network. We see in Figure 5 that the first two principal components alone are able to separate the vowels, consonants and silent letters in our reading model. However, at the level of individual words, we see in Figure 6 that many exception words (e.g. 'give') and sub-rules (e.g. the '–ind' words) still find themselves clustered incorrectly. The problem is that, 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). Moreover, the situation is only slightly improved if our network has only 30 hidden units (eigenvalues 0.174, 0.117, 0.086). Clearly, taking only the first two or three components on their own is bound to give a very poor representation of what is happening.

2.3 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 corresponds as closely as possible to those in the original space (Kruskal, 1964a, b; Shephard, 1962a, b, 1980; Young, 1987). Since we can start this iterative procedure with the positions given by PCA, we are guaranteed to end up with at least as good a representation of the inter-point distances as provided by PCA on its own. Though in practice, since the procedure can settle into local minima rather than the optimal configuration, it is usually sensible to start the procedure from a number of different configurations and select the best final configuration.

For small numbers of points, MDS works quite well. The average phoneme data of Figures 2 and 5 results in the two dimensional MDS plot shown in Figure 7. The rank correlation with distances in 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. In addition to the vowel, consonant and silent clusters evident

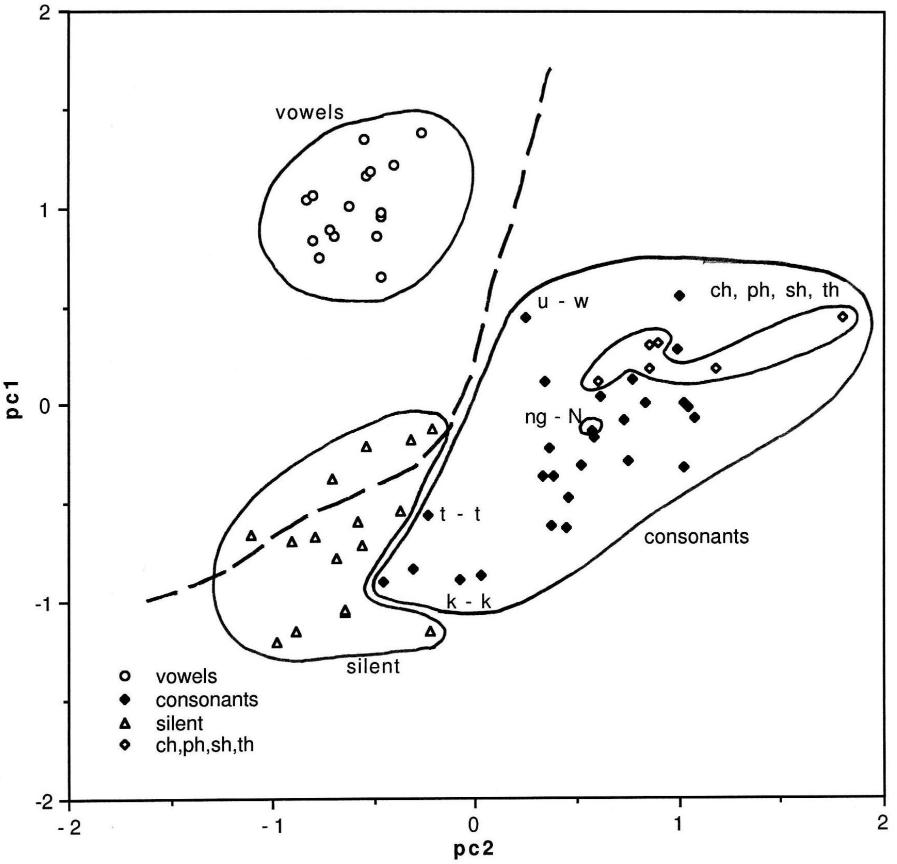

Figure 5: The first two principal components of the mean letter-phoneme positions.

in the PCA plot, a 'ch,ph, sh,th' cluster has separated itself as it did in the HCA. We can not only now see more clearly how the points are clustered but also better understand the anomalies in the HCA, e.g. why the 'k-k' point was grouped with the silent letters. Figure 8 shows that we can also get good plots for the individual words. We still have problems with the exception words and sub-rules, but there is a stronger tendency for them to appear at the edges of clusters as close as possible to the clusters of their regular counterparts. The rank correlation with the original data here is now 0.90 compared with 0.83 for the PCA. 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 not clear which lost information is responsible for the breakdown in correlation.

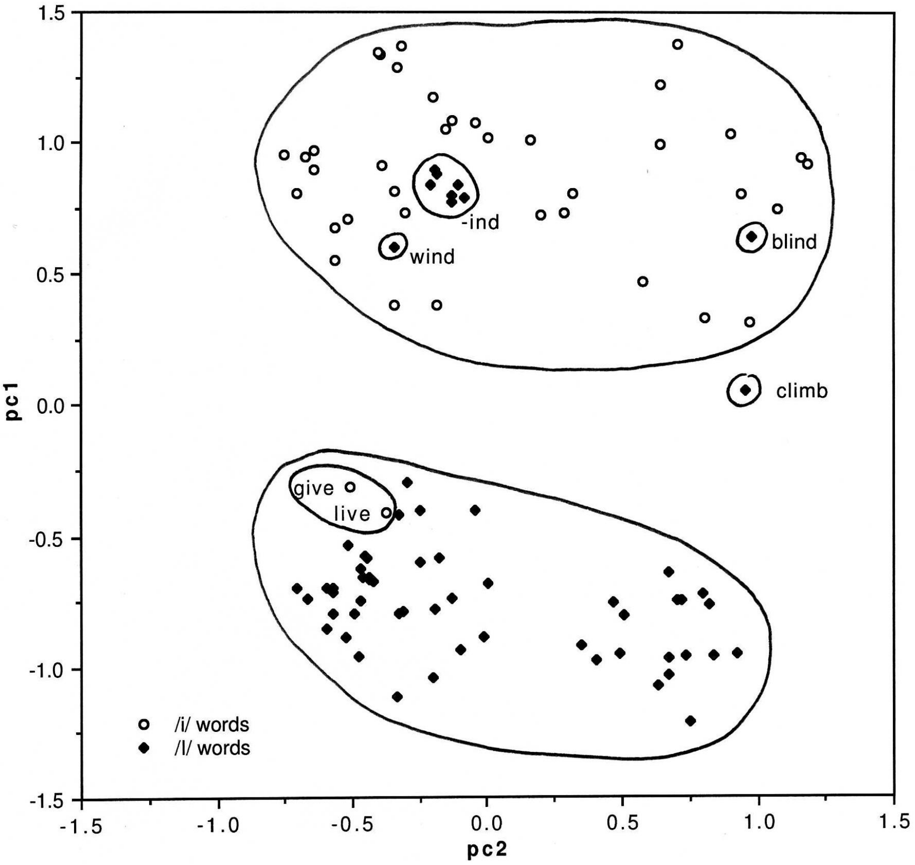

Figure 6: The first two principal components of our typical word set.

2.4 Discriminant analysis

The problem with both PCA and MDS is that they fail to take into account the fact that some hidden units are more important than others, i.e. they do not take into account the network's output weights. It is also becoming clear that we cannot expect to represent reliably the whole of our network's internal representation in only two dimensions. What should be feasible and more useful, however, is to plot a series of small useful subsets of the full representation based on the knowledge that we already have about the networks operation. For example, we could attempt to elucidate the network's distinction between the long /I/ and short /i/ sounds. We can do such a thing using Discriminant Analysis, which is a general procedure for projecting onto sub-spaces that optimise particular conditions (e.g. Devijner and Kittler, 1982). This approach, in the form of Canonical Discriminant Analysis (CDA), was successfully applied by Wiles and Ollila (1992) to study combinatorial structure in hidden unit space.

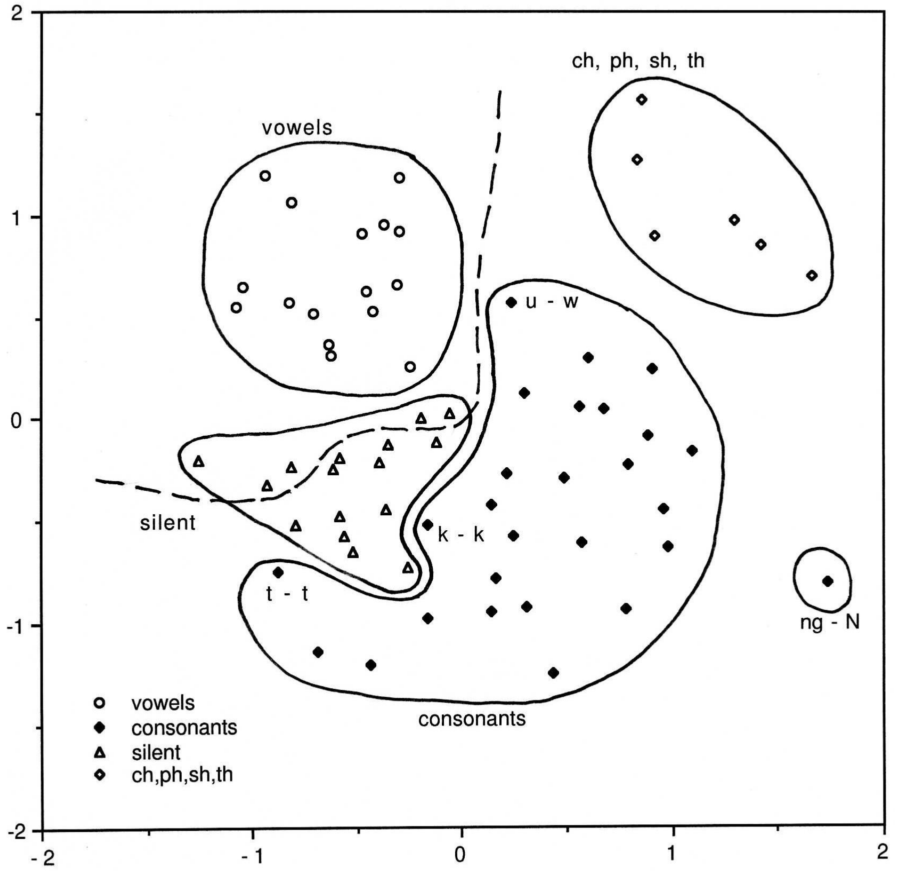

Figure 7: A two dimensional MDS plot of the mean letter-phoneme positions.

We shall attempt to use it to identify the hidden unit sub-spaces responsible for particular output patterns in our reading model.

If we know which of G groups each point in hidden unit space belongs to (e.g. which output phoneme it corresponds to), we can partition the total covariance matrix $S = W + B$ into the within groups covariance given by

$$W_{ij} = \sum_g \left(\sum_{\alpha \in g} (P_{i\alpha} - \langle P_i \rangle_g)(P_{j\alpha} - \langle P_j \rangle_g) \right) \tag{6}$$

and the between groups covariance given by

$$B_{ij} = \sum_g n_g (\langle P_i \rangle_g - \langle P_i \rangle)(\langle P_j \rangle_g - \langle P_j \rangle) \tag{7}$$

where the groups are labelled by g, contain n_g points and have mean components $\langle P_i \rangle_g$. The aim of discriminant analysis is to find a low dimensional

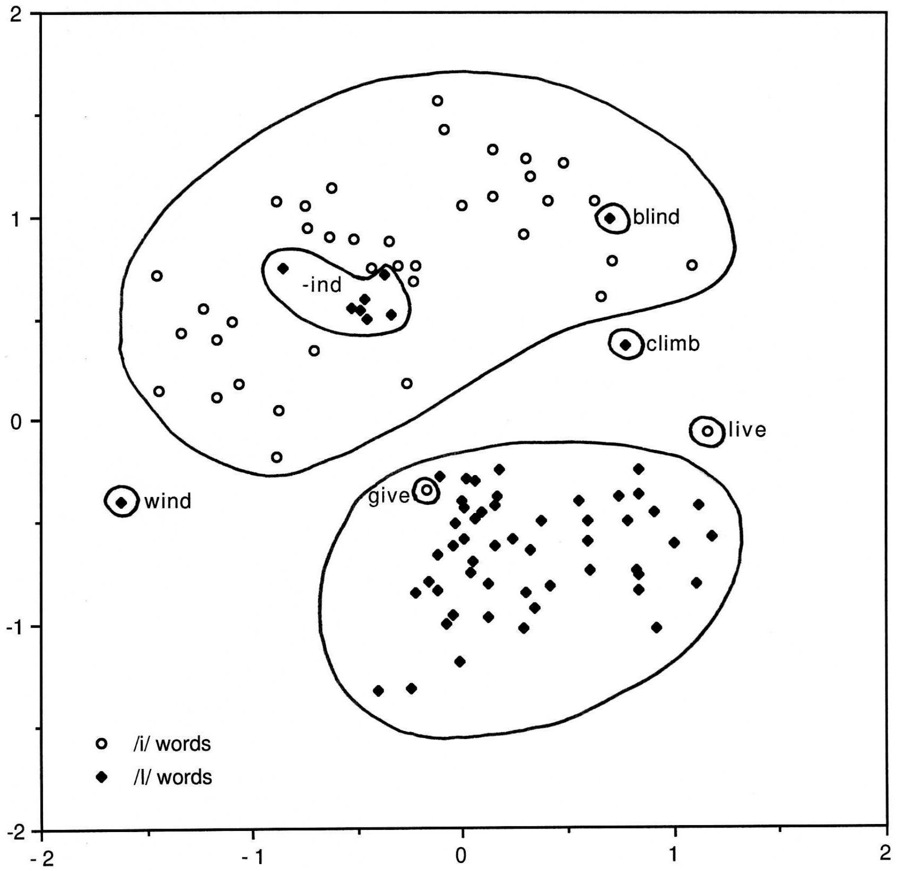

Figure 8: A two dimensional MDS plot of our typical word set.

subspace which best discriminates between the given groups. Since, roughly speaking, the determinant of a covariance matrix is a measure of the dispersion, a convenient fitness function to maximize is the ratio $|B|/|S|$ since it is well known (e.g. Healy, 1986) that this maximization can be achieved by solving the eigenvalue problem for $S^{-1}B$ to give a matrix M which projects our points onto a $rank(B) \leq G - 1$ dimensional subspace. In this sub-space the points will be clustered into groups with the maximum between group separations and minimum within groups dispersion. Unlike with PCA, the projection directions are not necessarily orthogonal, but still their eigenvalues provide a useful measure of the importance of each direction.

Since we know that our network performs a similar clustering (e.g. Gallinari et al., 1988; Webb and Lowe, 1990; 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 long /I/ versus short /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 two 'i' sounds. The quality of the discrimination will clearly depend 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 can get equally good discrimination even if we assign the points to groups at random ($B/S = 1.0000$). It is clear that this is not giving us 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. We then obtain 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, if we test this procedure on words or non-words not in the training data, the projection vector fails to classify them properly even when the network itself does. To define the projection more accurately we clearly need many more data points, particularly for the borderline region between the two groups. To this end a set of 14766 words and non-words of the form 'C_1VC_2' and 'C_1VC_2e' were generated, 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. Using these, the CDA then gave us a projection with $B/S = 0.83$, but there was now 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. Figure 9 illustrates the problem with a more manageable intermediate set of 942 words and non-words. As usual, the exception words (e.g. 'give' and 'pint') are at the forefront of the problems.

It is clear that standard CDA does not necessarily give us a good representation of the true internal representation. The problem is that, 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. This is simply because, for our non-linear networks, once the projections fall in the tails of the output sigmoids, very large differences can make relatively little difference to the networks' actual outputs (cf.the networks discussed in Gallinari et al., 1988; Webb and Lowe, 1990; Gallinari etal., 1991). One way we may attempt to get round the limitations of standard discriminant analysis is to start with CDA and then employ a simple iterative procedure to adjust the projection vectors so that all the points are correctly classified. There are many ways we can do this. Suppose, for example, $P_{i\alpha}$ is a point in hidden unit activation space and V_i is our projection vector, then the projection of each point is given by

$$D_\alpha = \sum_i V_i P_{i\alpha} \tag{8}$$

This projection can be increased or decreased by a standard gradient descent adjustment of V_i given by

$$\Delta V_i = \pm\varepsilon \frac{\partial D_\alpha}{\partial V_i} = \pm\varepsilon P_{i\alpha} \tag{9}$$

where ε is a small constant. If we keep ε sufficiently small and sum the ΔV_i

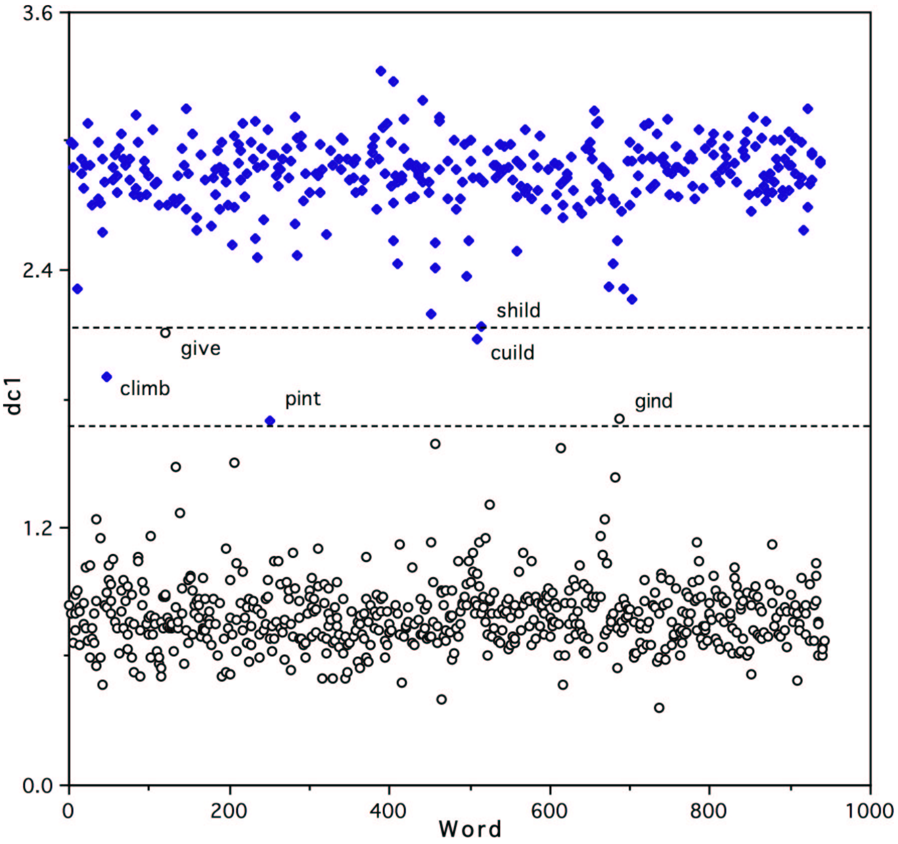

Figure 9: The canonical discriminant components for the /i/-/I/ distinction.

over an appropriate subset of points (e.g. all the misclassified points) we can iteratively decrease the overlap.

When this was done for our 14766 data points in the reading model it resulted in the correct classifications with a reduced $B/S = 0.73$. 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 data points into randomly assigned groups ($B/S = 0.49$).

Projection vectors provided by discriminant analysis are clearly not very useful if they can be found for groupings that bear little or no relation to what the network has actually learnt to do. Such spurious projections are possible because the procedure can make use of any noise that results from having a large number of extra hidden units that are not strictly necessary for performing the mapping (about 270 in the case of our reading model). If our network has very nearly the minimal number of hidden units (i.e. 30), the CDA still gives a projection vector that has the two groups overlapping ($B/S = 0.75$) and it is still possible to adjust the vector to separate the groups ($B/S = 0.69$). However, it is no longer possible to find a projection vector that correctly classifies the random groups.

We can conclude, therefore, that in general standard CDA does not reliably inform us how a network is operating. Using gradient descent procedures to find projection vectors that separate the groups can also lead to misleading results. The problem is that we can very easily end up with projection vectors that have little to do with the actual network outputs. Networks that have many hidden units more than actually required for the task in question will be particularly susceptible to these problems. At least, by attempting to use the networks' hidden unit activations to classify the points into random groups, it is possible to get some estimate of the reliability of the projection vectors in particular situations.

2.5 Output weight projections

For the simple one hidden layer architecture and localist output representation of our reading model, it is actually very easy to find projection vectors that correlate with the outputs. We can simply use the projections the network itself has learnt - namely the output weights. If we project the hidden unit activations using the output weights W_{ij} and redefine the zero points using the output thresholds θ_j, our projections are then simply the network outputs before being passed through the sigmoid. We are guaranteed rank correlation. We thus have a suitable projection vector for each phoneme and these 38 vectors actually turn out to be nearly orthogonal (mean angle 84°, s.d. 5°). These projections can then be viewed in pairs to examine the relationships between the clusters, or any of the above techniques may be used to study 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 10 shows the resultant discrimination for our /I/ and /i/ phonemes for our 300 hidden unit network. 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. Thus points corresponding to other phonemes would appear in the bottom left quadrant.

The projection vector that we were attempting to find by discriminant analysis now corresponds to the /i/-/I/ diagonal in Figure 10 . We can easily see the clear separation of the groups and the unrestricted within group dispersion that really corresponds to what the network has learnt ($B/S = 0.72$). We can also easily check the relation of these correct projection vectors to those provided by the CDA and gradient descent group separation. For the 300 hidden unit network the angles between the vectors are 113° and 63° respectively. For the 30 hidden unit case the corresponding angles are 24° and 3°, which is consistent with our intuition that we get more reliable results for networks with fewer spare hidden units.

Plotting trajectories on graphs such as Figure 10 can help us understand how

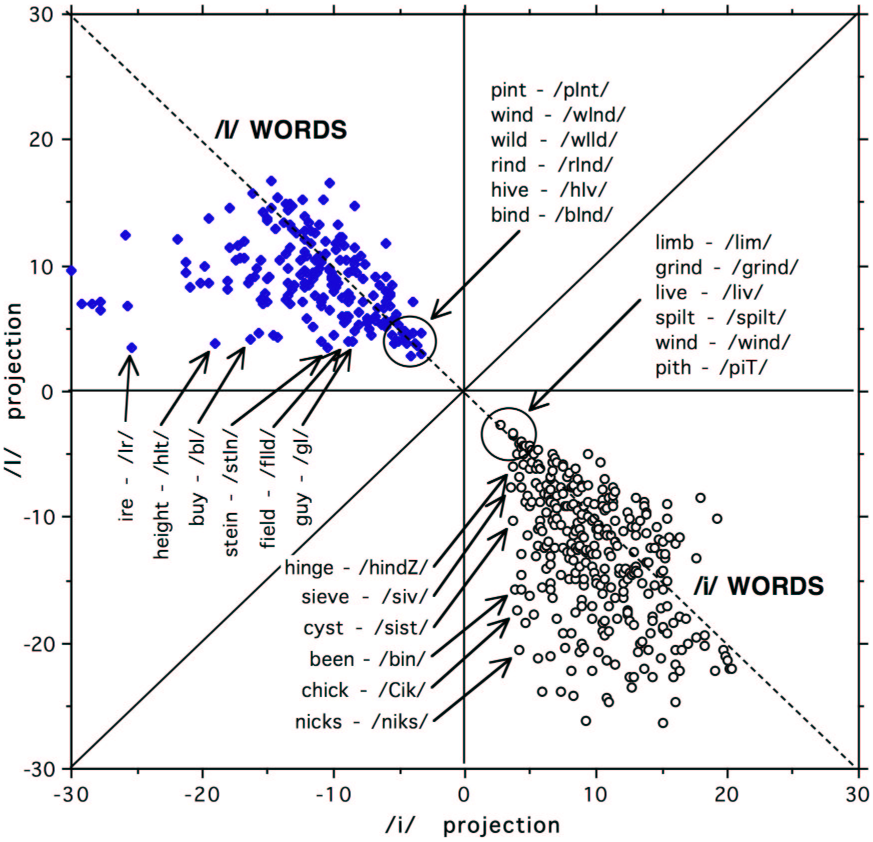

Figure 10: The 2D hidden unit sub-space corresponding to the /i/ and /I/ phonemes.

the final pattern of projections arises, as well as possible causes of developmental problems (such as developmental dyslexia) and how the network responds to damage. Of course, the output weights and thresholds change during the learning or damage process at the same time as the hidden unit patterns, so whilst the following description is broadly correct, we should be careful not to take it too literally.

If we begin training with small random initial weights, all hidden unit activation points start near coordinates A_j given by

$$A_j = \frac{1}{2}\sum_i W_{ij} - \theta_j \tag{10}$$

They then each step towards their appropriate quadrant. There are several effects that will determine the final position of each word presentation. First, as is clear from our HCA and MDS plots, similar 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 plenty of 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 hence be the ones left closest to the axes. Strange words (such as 'sieve'), that have very rare spelling patterns, may also be left near the axes depending on their word frequency. Such a pattern of learning is in broad agreement with that found in children (Backman et al., 1984). Also, the pattern of performance that will arise if there are problems in completing the later stages of learning is consistent with developmental surface dyslexia in children (Coltheart et al., 1993).

These effects are seen clearly in Figure 10 . The points (circled) nearest the group border lines tend to be exception words (e.g. 'pint'), regular inconsistents (e.g. 'hive') and homographs (e.g. 'wind'). The other points (arrowed) near the zero projection lines correspond to border line cases in orthogonal directions (e.g. 'been' is a borderline case in the /i/-/E/ plane) or orthographically strange words (e.g. 'sieve'). Similarly, Figure 11 shows the two dimensional sub-space corresponding to the /i/ and /f/ phonemes with the /i/, /I/ and /f/ words plotted. Not surprisingly, given the virtual absence of ambiguity between the /i/ and /f/ phonemes, the groups are much further apart than the /i/ and /I/ phonemes were in Figure 10. We can also see how the /i/-/I/ distinction looks from orthogonal directions.

It is often argued that there should be a correlation between network output activation error scores and the corresponding reaction times in humans (Seidenberg and McClelland, 1989). This follows because, in the more realistic cascaded approach to modelling reaction times (McClelland, 1979; Bullinaria, 1995b), the rate of output activation build-up is proportional to the appropriate projection. Thus, the closer each point falls to the axes of our projection graphs, the longer the corresponding reaction time. We can therefore easily read off from our graphs the model's predictions for naming latency experiments: There will be a basic frequency effect. 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 do turn out to be in broad agreement with experiment (for a detailed discussion see Bullinaria, 1994a, 1995b).

The original reason for wanting to investigate our networks 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 cast doubt on the traditional dual route models of reading with their separate phonemic and lexical 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

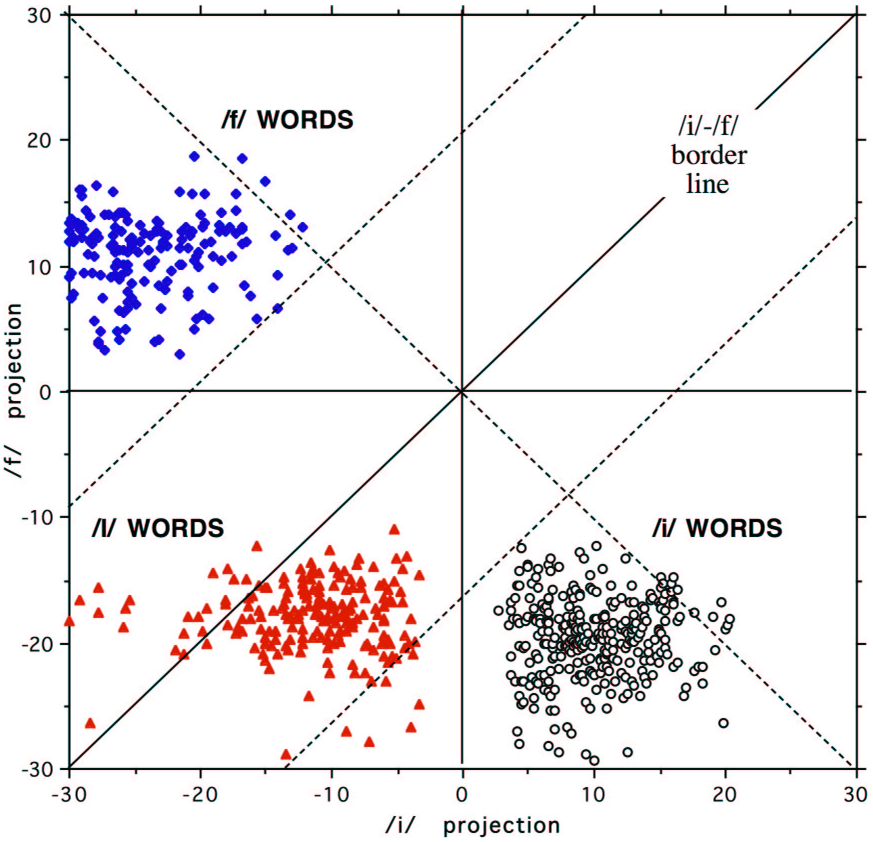

Figure 11: The 2D hidden unit sub-space corresponding to the /i/ and /f/ phonemes.

dyslexia (lost non-words) when damaged appropriately (Coltheart et al., 1993). A detailed study of six representative forms of network damage (Bullinaria, 1994b) showed that, for each form of damage where there was a significant type effect (namely weight scaling, weight reduction, addition of noise to weights, removal of random connections and removal of random hidden units), we always find symptoms similar to surface dyslexia but never anything like phonological dyslexia. In each case we increased the degree of damage from zero to a level where the network failed to produce any correct outputs at all and patients with varying degrees of dyslexia corresponded to particular intermediate stages of this process.

How do we understand these results in terms of our projection plots? Since we are primarily concerned here with the internal representations we shall illustrate the analysis for just one form of damage, namely weight scaling. It is this form of damage in small networks that seems to give the most reliable indication of what is likely to occur in more realistic networks (Bullinaria and Chater,

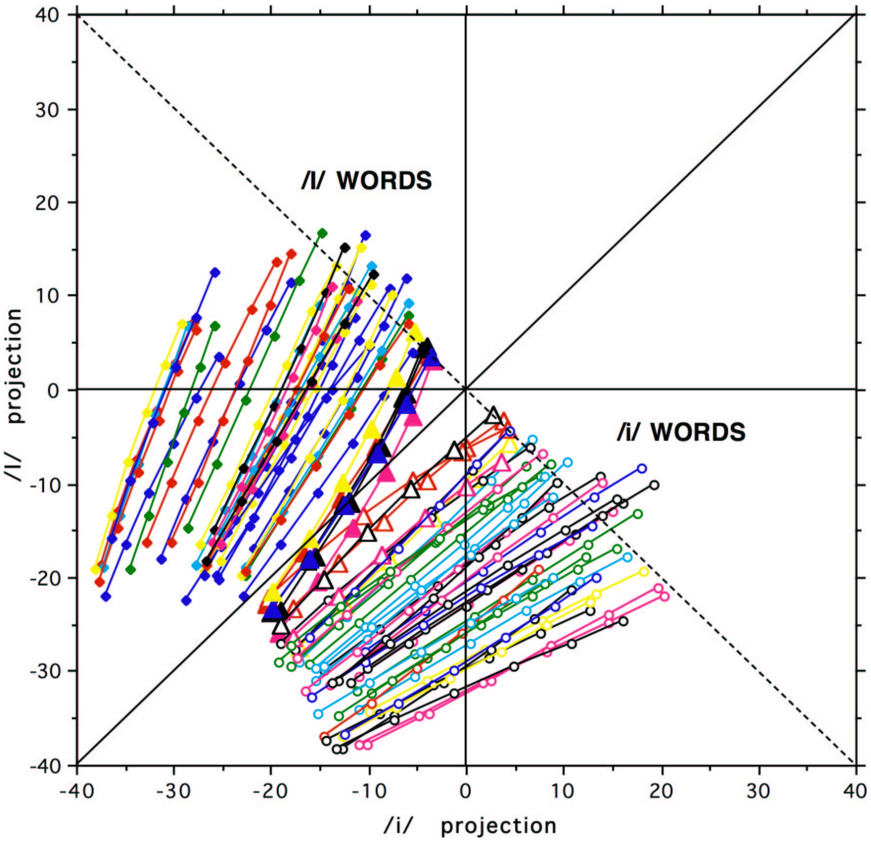

Figure 12: The effect of damage by weight scaling on our /i/-/I/ projections.

1995). We simply scale all the weights and thresholds by a constant scale factor $0 < \gamma < 1$. The effect of decreasing γ 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_j defined above. It turns out that all the A_j 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. We see this clearly in Figures 12 and 13. 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. Given this clear understanding of the effects of damage here we can be more confident in our claims about the effects of damage more generally (Bullinaria and Chater, 1995).

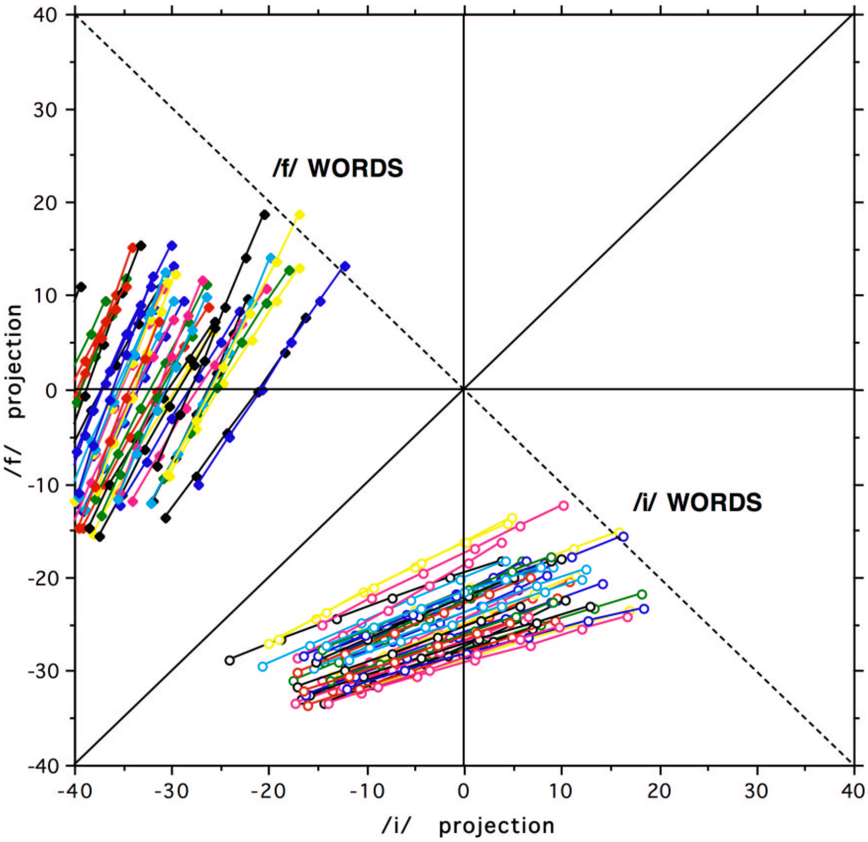

Figure 13: The effect of damage by weight scaling on our /i/-/f/ projections.

2.6 Contribution analysis

We have already mentioned briefly the important related questions of robustness and scaleability, i.e. are our networks sufficiently robust to damage and are our networks sufficiently large scale such that we can confidently extrapolate their performance to larger networks. In this section we discuss Contribution Analysis, which is a technique for analysing the importance of individual hidden units in connectionist networks (Sanger, 1989).

In terms of our hidden unit activation $P_{i\alpha}$ for input pattern α, the standard network output activation can be written as in

$$Out_{j\alpha} = Sigmoid\left(\sum_i W_{ij} P_{i\alpha} - \theta_j\right) \tag{11}$$

Hence the contribution of hidden unit i to output unit j can be defined by

$$C_{ij\alpha} = Sign\left(\sum_i W_{ij}P_{i\alpha} - \theta_j\right) \cdot W_{ij}P_{i\alpha} \qquad (12)$$

We introduce the $Sign$ function to ensure that positive contributions always enhance the accuracy of the output whether it be 0 or 1. Since the contributions $C_{ij\alpha}$ take into account the output weights, we should expect them to be more useful for analysing the network's performance than the $P_{i\alpha}$ themselves. However, the extra index j means that we now have many times the number of components to deal with and these clearly have to be reduced for visualization purposes. Perhaps the most convenient and illuminating way to proceed is simply to perform analyses of the form described above on the contributions for particular output units or particular hidden units. Sanger (1989) originally illustrated the advantages of this approach using PCA on a simplified reading model with one hidden layer. More recently, Shultze and Elman (1994) have shown how it can also be used to analyse multi-layer networks with cross-connections between hidden layers.

Another important use of contribution analysis, that we have not already discussed, is to examine the effect of single hidden units or connections on the output of the network. If individual contributions, or small numbers of contributions, have a significant effect on the networks outputs, then the network will not be robust with respect to damage and simulations of network damage will not necessarily scale up to more realistically sized networks. By definition, minimal networks will be highly dependent on individual contributions, whereas much larger networks performing the same task are likely to form more distributed internal representations with individual contributions insignificant with respect to the sum of the others.

This aspect of network analysis was discussed more fully by Bullinaria and Chater (1995) in the context of connectionist neuropsychology. The important variable here is the ratio $\mathcal{C}$ of each contribution compared with the total effect of all the contributions (including the threshold). If $\mathcal{C} > 1.0$, then removing that contribution will change the sign of the total and (assuming the network has been trained to produce near binary outputs) drastically change the output. Figure 14 shows (for a simple model described fully in Bullinaria and Chater, 1995) that the number of patterns affected by at least one such contribution does indeed decrease from 100% for minimal networks to zero for larger networks (e.g. with more than 150 hidden units in this case). If $\mathcal{C} > 1/N$ we run the risk that the removal of N random connections will have a significant affect on the outputs. Figure 14 also shows how the number of patterns affected by contributions with $\mathcal{C} > 0.5$ and $\mathcal{C} > 0.3$ falls with the number of hidden units. It is worrying that we need tens, if not hundreds, of times the minimal number of hidden units to provide a reasonably robust and distributed network.

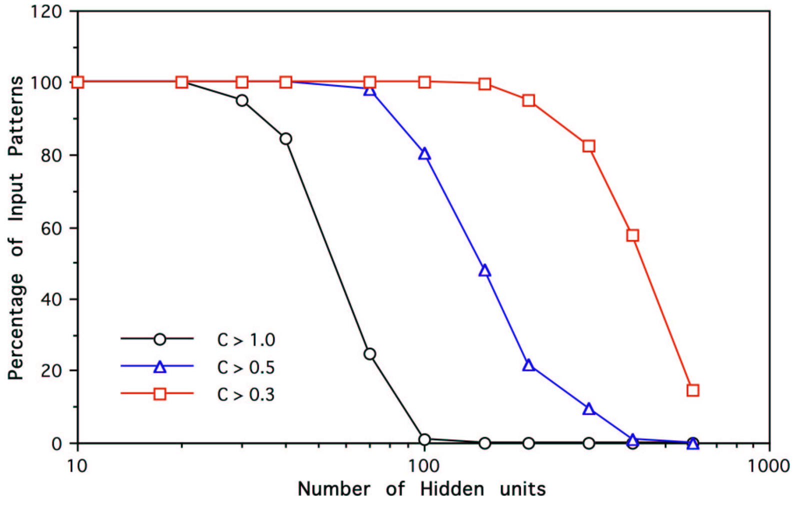

Figure 14: The number of patterns dependent on significant individual contributions.

3 Conclusions

We have surveyed a number of techniques for analysing the internal representations (i.e. patterns of hidden unit activation) of trained feedforward networks and seen how they may all provide useful information concerning the operation of what would otherwise be 'black box' systems. In doing so we have also identified a number of pitfalls in these approaches which may result in misleading information about how the networks are performing.

We began by seeing how cluster analysis can indicate patterns of hidden unit activation that agree well with how we might expect the network to operate, but the mis-classification of exceptional items (that the network itself could handle correctly) highlighted the fact that this approach was missing out on some crucial aspects of the networks performance. We then turned to various techniques for reducing the dimension of the hidden unit activation space to something that could be visualised more directly. Firstly we used standard principal component analysis to reduce the number of dimensions with minimal loss of information, but found that for non-trivial problems two or three dimensions simply cannot capture enough of the variance for more than the grossest features to be recognised. We then saw how the non-metric approach of multi-dimensional scaling could provide a slightly better picture of what was happening. The problem with both these approaches is that neither takes account of the fact that some hidden units are more important than others simply because they are connected with different weights to the output layer. They consequently tend to mis-represent what is happening for some items, the exceptional items in particular. To remedy this problem we then looked for particular directions in the hidden unit activation space that corresponded to known features of the network's mapping. Using discriminant analysis we were able to find projection vectors that best discriminated between various classes of network outputs. However, careful analysis showed that this approach could indicate modes of network operation that did not actually correspond to how the network was really operating, and that this difficulty was particularly problematic when the network employed many more degrees of freedom than were necessary to perform the given task. It was then noted that, for networks where it was possible, simple projections onto sub-spaces defined by the output weights provide a much better picture of the network's internal representations than any of the preceding techniques. This is unfortunate, since output weight projections will not be nearly so simple for systems that have more complicated output representations. We ended with a brief discussion of contribution analysis and how it may be used with the preceding techniques and how it can be used to investigate the robustness and scaleability of trained networks.

Throughout this chapter we have used a simple reading model to illustrate the various techniques. We have seen explicitly in this case how an analysis of the internal representations can not only provide an insight into how the network is operating, but can also lead to a better understanding of various human developmental effects and reaction times. Similarly, analysing the network's response to damage can lead to better models of acquired dyslexia. It seems likely that the simple techniques discussed in this chapter will be able to provide equally useful insights into the operation of a wide range of other connectionist systems.

4 References

Backman J, Bruck M, Hébert M and Seidenberg M 1984 Acquisition and use of spelling-sound information in reading *Journal of Experimental Child Psychology* **38** 114–33

Bullinaria J A 1994a Representation, learning, generalization and damage in neural network models of reading aloud. Edinburgh University Technical Report

Bullinaria J A 1994b Internal representations of a connectionist model of reading aloud *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (Hillsdale, NJ: Erlbaum) pp 84–9

Bullinaria J A 1995a Neural network learning from ambiguous training data *Connection Science* **7** 99–122

Bullinaria J A 1995b Modelling reaction times *Neural Computation and Psychology* ed L S Smith and P J B Hancock (New York: Springer Verlag) pp 34–48

Bullinaria J A and Chater N 1995 Connectionist modelling: Implications for cognitive neuropsychology *Language and Cognitive Processes* **10** 227–64

Coltheart M, Curtis B, Atkins P and Haller M 1993 Models of reading aloud:

Dual-route and parallel-distributed-processing approaches *Psychological Review* **100** 589–608

Devijver P and Kittler J 1982 *Statistical Pattern Recognition* (New Jersey: Prentice Hall)

Elman J L 1990 Finding structure in time *Cognitive Science* **14** 179–211

Elman J L 1993 Learning and development in neural networks: The importance of starting small *Cognition* **48** 71–99

Everitt B 1975 *Cluster Analysis* (London: Heinmann)

Gallinari P, Thiria S and Fogelman-Soulie F 1988 Multilayer perceptrons and data analysis *Proceedings of the Second International Joint Conference on Neural Networks* Vol 1 (San Diego: SOS Printing) pp 391–99

Gallinari P, Thiria S, Badran F and Fogelman-Soulie F 1991 On the relations between discriminant analysis and multilayer perceptrons *Neural Networks* **4** 349–60

Hanson S J and Burr D J 1990 What connectionist models learn: Learning and representation in connectionists networks *Behavioral and Brain Sciences* **13** 471–518

Healy M J R 1986 *Matrices for Statistics* (Oxford: Clarendon Press)

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

Kruskal J B 1964b Non-metric multidimensional scaling: A numerical method *Psychometrika* **29** 115–29

McClelland J L 1979 On the time relations of mental processes: An examination of systems of processing in cascade *Psychological Review* **86** 287–330

Rumelhart D E, Hinton G E and Williams R J 1986 Learning internal representations by error propagation *Parallel Distributed Processing* vol 1 ed D E Rumelhart and J L McClelland (Cambridge MA: MIT Press)

Sanger D 1989 Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks *Connection Science* **1** 115–38

Seidenberg M S and McClelland J L 1989 A distributed, developmental model of word recognition and naming *Psychological Review* **96** 523–68

Sejnowski T J and Rosenberg C R 1987 Parallel networks that learn to pronounce English text *Complex Systems* **1** 145–68

Shepard R N 1962a The analysis of proximities: Multidimensional scaling with an unknown distance function, I *Psychometrika* **27** 125–40

Shepard R N 1962b The analysis of proximities: Multidimensional scaling with an unknown distance function, II *Psychometrika* **27** 219–46

Shepard R N 1980 Multidimensional scaling, tree-fitting and clustering *Science* **210** 390–8

Shultz T R and Elman J L 1994 Analyzing cross connected networks *Advances in Neural Information Processing Systems* Vol 6 ed J D Cowan, G Tesauro and J Alspector (San Mateo, CA: Morgan Kauffmann) pp 1117–24

Webb A R and Lowe D 1990 The optimised internal representation of multilayer classifier networks performs nonlinear discriminant analysis *Neural Networks* **3** 367–75

Wiles J and Ollila M 1993 Intersecting regions: The key to combinatorial structure in hidden unit space *Advances in Neural Information Processing Systems* Vol 5 ed S J Hanson, J D Cowan and C L Giles (San Mateo, CA: Morgan Kauffmann) pp 27–33

Young F W 1987 *Multidimensional Scaling: History, Theory and Applications* (Hillsdale, NJ: Lawrence Erlbaum)