Proceedings of The Institute of Acoustics

SELF ORGANISING ARRAYS FOR SPEECH RECOGNITION

G.D.Tattersall and R.D.Johnston

British Telecom Research Laboratories, Martlesham Heath,
Ipswich, IP7 5RE

Abstract:

Over the last twenty five years many attempts have been made to electronically simulate the action of neurons in the brain with the aim of producing a pattern recognising machine with some of the characteristics of a human. However, until recently it has not been appreciated that neural models incorporating lateral excitation and inhibition can exhibit self organising properties which allow the unsupervised extraction of features from patterns applied to the system.e.g. Kohonen (1),Hirai(2).

Recently it has been demonstrated that this type of self organising neural model can be used for the recognition of phonemes of one speaker to whose speech the system has been exposed (1),(4). However,these experiments have been limited by computational speed to arrays containing about 400 " neurons" and input pattern vectors of up to about 30 dimensions.

At British Telecom Research Laboratories a high speed hardware self organising neural array has been built and will be used for an investigation into the characteristics of speech and perhaps ultimately as the basis of a speech recogniser.

This paper describes the principles of the self organising system, the types of investigation which seem appropriate and notes some of the rules found to be necessary for the successful operation of the system.

Feature Extraction for Speech Recognition:

The central problem in automatic speech recognition is to extract suitable features of an utterance from its time domain representation.The rationale for this is twofold:

First,the time waveforms of the same speech utterance spoken by different speakers or by the same speaker on different occasions are very different and this makes recognition of the utterance by direct comparison of time domain waveforms with stored templates unreliable. The solution is to try and extract features from the time domain representation of the speech which are invariant over all speakers and which at the same time have different values for each type of utterance in the vocabulary.

The second reason for extracting features from the speech for recognition is to reduce the amount of data which has to be used in the recogniser. However ,this is no longer a very strong reason with currently available semiconductor memory sizes and computational power.

Finding Good Features:

The most commonly used features used in speech recognisers to the present have been some form of spectral coefficient set coupled with time warping. The reasons for this are largely historical: The absence of digital processing techniques, meant that in the past the only practical data compression system was by means of an analogue filter bank. Another reason is the belief that the ear acts as filter bank. This is undoubtedly true in some sense , but it is not certain that phase information is thrown away by the ear as is done in the filter bank approach to speech recognition. Overall ,the use of spectral coefficients as features appears rather arbitary.

How can suitable features be found in a more rigorous way? To answer this ,the nature of a feature must be investigated. Consider the three dimensional pattern space shown in figure 1. , points corresponding to some imaginary class of patterns are plotted in this space. The obvious feature of this data is that although the patterns are described in a three dimensional space they all lie within a plane in the pattern space. The best feature extractor which could be found for this type of data would project the data points onto a two dimensional subspace (surface). This would reduce the amount of information required to pin point each pattern and also remove the apparently insignificant information about the variability of the pattern along a normal to the plane.

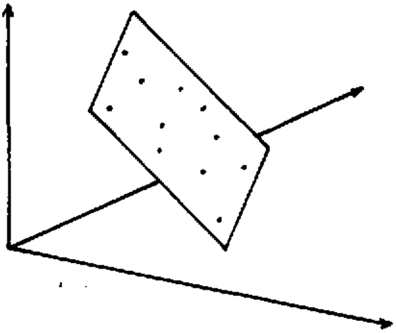

Fig1. Data lying within a 2-D plane in a 3-D pattern space.

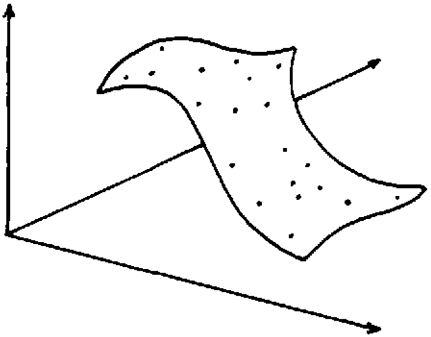

Fig2. 2-D data lying on a complex surface in a 3-D pattern space.

The eigen vectors of the subspace onto which data can be projected for feature extraction can be found analytically using the

Karhunen-Loeve Transform and this has been used in speech recognisers,e.g (3),although the original pattern space was described in terms of spectral coefficients rather than time domain samples of speech.

Unfortunately, this analytic approach cannot deal with the type of data shown in figure 2. The inherent dimensionality of this data is clearly only two but it cannot be directly projected onto a flat surface or linear sub space. No linear transform such as the Fourier transform or Karhunen-Loeve transform can extract the feature of this data: A non-linear transform is required. Finding an appropriate transform for real speech data is virtually impossible using analytic techniques. However, it has recently been shown that a particular type of neural array is capable of performing this non-linear transformation. The following sections of this paper describe the neural array and its use in some experiments on speech . The results of these experiments will be presented at the conference.

Physiological Description of Neural Array:

A common type of neuron in the cortex is the pyramidal neuron shown in figure 3. Although the neurons superficially appear to be operating in a binary pulse mode it is thought that this is actually pulse density encoding of analogue signals.With this assumption, each neuron seems to behave like an analogue multi-input summing amplifier with modifiable gains on each of the inputs .The axon corresponds to the summing amplifier output, the afferent fibres to the summing inputs ,and the modifiable synapses to the variable weights on each input of the amplifier.

There is evidence that the neurons in the cortex are arranged with the following features:

a) Often neurons are densely interconnected within a plane and sucessive planes are more sparsely connected together. The interconnections within a plane are from the output of a neuron to the inputs of surrounding neurons.

b) Large numbers of neurons within a local plane appear to have the similar inputs.

c) If one neuron in the plane is excited then the excitation of physically adjacent neurons follows the "mexican hat function" shown in figure 4. which shows lateral excitation and inhibition.

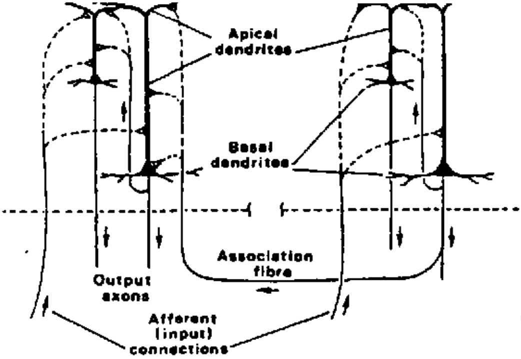

Fig3. Pyramidal neurons in the cortex.

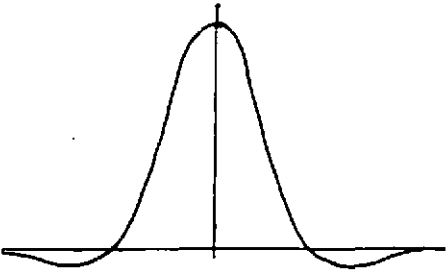

Fig4. Lateral excitation function between neurons in the array.

The operation of this type of neural array when exposed to input pattern vectors via the afferent fibres is largely speculative but if just a few apparently reasonable assumptions are made then the array is found to have powerful feature extraction properties.

The first assumption is that when an input vector is applied to the array each neuron will tend to be excited in proportion to the closeness of its synaptic weight vector to the input pattern vector.i.e Assuming the neurons to be weighted input summing amplifiers, the neuron output will be the scalar product of the input vector and that neuron's synaptic weight vector. The second assumption is that the outputs of each neuron are non-linearly scaled such that the output of just one neuron will dominate in a particular locality. The final assumption is that the synaptic weight vectors of each neuron are moved towards or away from the input pattern vector depending on whether the overall excitation of that neuron is positive or negative after all the effects of lateral excitation and inhibition are taken into account.

With these assumptions a simple computational algorithm for the neural array can be formulated as follows:

1) Set up a two dimensional array of elements(neurons),each with storage for an initially random synaptic weight vector order k.

2) Take an input pattern vector of order k and find the neuron with the closest stored synaptic weight vector.

3) Define an excitory and inhibitory neighbourhood around that neuron in the array and modify the synaptic weight vectors of each neuron in the neighbourhood such that they move towards or away from the input pattern vector depending on whether it lies in an excitory or inhibitory part of the neighbourhood.

i.e $$Wn = Wn + Kn * (X - Wn) \quad ....1$$

Where Wn is the weight vector of the nth neuron in the array ,X is the input vector and Kn is the factor determined from the "mexican hat" lateral excitation function.

Properties of the Neural Array Model:

The properties of the neural array model are most easily demonstrated by generating an artificial data set of random two dimensional vectors having a uniform probability distribution at any radius from the centre of their pattern space and a Gaussian distribution along a radius. The scatter plot of such points is shown in figure 5. If vectors are drawn at random from this distribution and applied to a one dimensional neural array it is found that the synaptic weight vectors associated with each neuron start to cluster along the ridge of the data's probability distribution. More startling, neurons which are adjacent to each other in the array take on synaptic weight values which are adjacent in the pattern space. In other words ,the array becomes topologically related to the data as is shown in figure 6. and the data is projected through a complex non-linear transform onto the array.

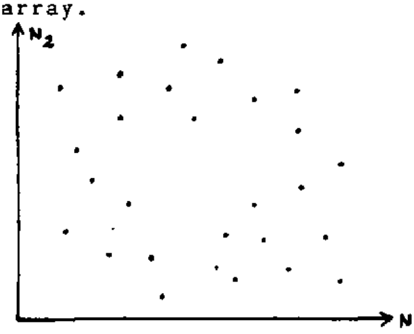

Fig5. Scatter plot of 2-D data.

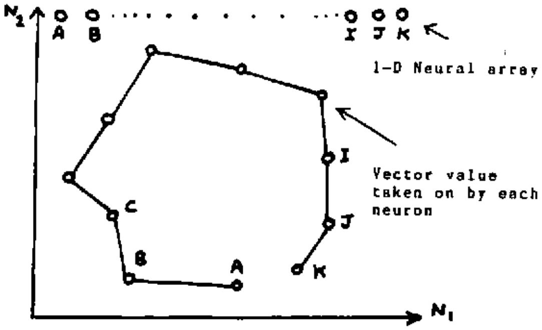

Fig6. 1-D neural array and its associated synaptic weight vector after exposure to the data of Fig5.

In general data which is embedded in a very high dimensional space can be projected onto a neural array of low dimensionality as long as the inherent dimensionality of the data is not greater than the dimensionality of the array.This is feature extraction. If the inherent dimensionality of the data exceeds that of the array then the array will fold itself so as to fill the subspace occupied by the data. Of course, when this happens ,the topological ordering of the map is disturbed. However,this in itself is a useful property, because it means that the inherent dimensionality of a set of data can be determined by increasing the dimensionality of the array onto which it is projected until the array is seen to be topologically ordered.

Neural Arrays and Speech Recognition:

A two dimensional neural array has been used by Kohonen (1) for the recognition of Finnish phonemes described in a pattern space of thirty DFT coefficients .Such a system could form the basis of a speech recogniser. However it is perhaps more interesting at this stage to use the array as a tool for investigating the properties of speech sounds. To this end a high speed hardware neural array model has been built at BTRL .In the long term it is intended to expose the array directly to sequences of time domain samples of speech to see if speaker independent features can be found which do not depend on spectral analysis. However,in the short term an attempt is being made to set a bench mark by applying spectral coefficients of speech to the array in the following experiments:

Single Speaker Clusters:

A phrase from a single speaker, "Why were you away a year Roy?" will be segmented into blocks of 256 samples and spectrally analyzed to yield sixteen spectral coefficients equally spaced in frequency. Each of the 16 dimensional vectors will be applied many times in random order to a 20*20 neural array such that the total number of "training passes " is 20000. The values of the synaptic weight vectors in the array will then be analyzed to see if the array is topologically ordered and also to measure the proximity of adjacent neuron's synaptic weight vectors over the entire array.This should give a measure of cluster density and cluster separation in the original 16 dimensional pattern space. It is of course expected that the clusters will correspond to particular speech primitives. The cluster density and separation should indicate how reliably a recogniser working on these types of spectral analysis could operate.

Multi Speaker Clustering :

The same experiment will be repeated except that the speech will be taken from several different speakers. The values taken on by the neural array will be analyzed to see if discernible clusters still exist and if so, how their separation has changed.

All the previous tests will be repeated for spectral analysis block lengths ranging from 4ms to 32ms and using an enlarged speech test set containing nasals and fricatives as well as vowel sounds.

Inherent Dimensionality of Speech:

In this experiment arrays of various dimensionalities will be exposed to the sets of spectral coefficients and the minimum array dimensionality required for a topologically ordered map determined. This test would provide an interesting piece of circumstantial evidence for or against the speculative idea that the ear produces spectral features which are initially projected onto a two dimensional neural array.

Hardware for Neural Array Model:

In order to train the neural array ,it must be exposed to very large numbers of input patterns .For each input the neuron with the nearest synaptic weight vector must be found and then all the neuron vectors in the array updated using equation(1). This is a computationally time consuming task when done in software on a mini computer and so a the neural array has been implemented in hardware form under the control of a micro computer.

The hardware consists of thirty identical rack mounted cards each communicating with the controlling micro computer via a common bus.Each card consists of 32kbytes of memory to store synaptic weight vectors and their corresponding difference vectors (Xn - Wn) along with logic to determine the position of the neuron in the array with the smallest difference vector and logic to update all neuron values in the array according to equation (1). The definition of the lateral exctitation function is software controlled.

The total memory capacity available for storing synaptic weight vectors is 240 kbytes and this can be partitioned under control of the microcomputer between array size and vector order. For example ,a 32*32 array could be set up which could deal with input vectors of order 2048.

Observations from using a Neural Array:

a) Nearest Neuron Metric:

Computationally ,the simplest metric for finding which synaptic weight vector is nearest to the current input vector is "city block". This metric does not actually match the Euclidean space in which we wish to generate the array map,but it has been found that the measure will enable the array to become roughly ordered. At this stage the vector distances are so small that there is very little difference between Euclidean distance and city block distance, and the system will continue to full convergence.

b) Neighbourhood Metric:

Since it is required that the map be topologically ordered, the metric which is used to determine the neighbourhood of neurons which are updated must be consistent with the spatial distances between neurons in the physical array. Thus if the array is rectangular, a Euclidean metric should be used. If the array is based upon a diamond shape ,then the city block metric should operate.

c)Neighbourhood as a Function of Time:

At the start of training the neighbourhood must be set wide in order that topological ordering can occur. However,if it is not reduced as time progresses it is difficult for the synaptic weight vectors to converge to values which accurately mirror the statistics of the input patterns .A typical result is that all the synaptic weight vectors are pulled towards the average of all the pattern vectors to which the system has been exposed. The result is a shrunken map. The solution is to linearly decrease the neighbourhood size as training progresses.

d) Lateral Excitation Function:

The original software simulations of the system done at BTRL showed that very low levels of inhibition aided rapid ordering of the array and also gave convergence without reducing the neighbourhood size. The necessary ratio between excitation and inhibition values being about 100 to 1 while the excitory neighbourhood size was about one eigth of the pattern space dimension and the inhibitory neighbourhood about one half. However ,it has been found that when using 8 bit integer arithmetic in the hardware ,inhibition leads to instability and has therefore been abandoned.

It has also been found that a computational simplification can be made at the cost of increased ordering time: The expression for updating the neuron values (equation (1)) can be modified so that the synaptic weight vector is moved by an incremental amount away or towards the current input vector.

$$Wn = Wn + (X - Wn)/|X - Wn| \ldots\ldots\ldots\ldots 2$$

In the computation this is implemented merely by adding the sign of the difference between the ith element of X and the ith element of Wn to the value of the ith element of Wn.

Acknowledgement is made to the Director of British Telecom Research Laboratories for permission to publish this paper.

References:

1) T.Kohonen, Clustering Taxonomy, and Topological Maps of Patterns, Proc. 6th International Conf. on Pattern Recognition, IEEE, October

1982.

2) Hirai, A Template Matching Model for Pattern Recognition: Self Organisation of Templates and Template Matching by a Disinhibitory Neural Network, Biol Cybernetics 38,91-101,1980.

3) T.Kohonen et al, A Thousand Word Recognition System Based on The Learning Subspace Method and Redundant Hash Addressing, Proc. IEEE 5th International Conf. on Pattern Recognition, December 1980.

4) M.J.Carey, The classification of phonemes by a self ordering network, Institute of Acoustics Autumn Conf. ,1984.