UNIVERSITY of OTAGO
TE WHARE WĀNANGA O OTĀGO
DUNEDIN NEW ZEALAND

Spatial-Temporal Adaptation in Evolving Fuzzy Neural Networks for On-line Adaptive Phoneme Recognition

Nikola Kasabov
Michael Watts

The Information Science Discussion Paper Series

Number 99/03
March 1999
ISSN 1177-455X

University of Otago

Department of Information Science

The Department of Information Science is one of six departments that make up the Division of Commerce at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Copyright

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: http://divcom.otago.ac.nz:800/COM/INFOSCI/Publctns/home.htm). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311
email: dps@infoscience.otago.ac.nz
www: http://divcom.otago.ac.nz:800/COM/INFOSCI/

Spatial-Temporal Adaptation in Evolving Fuzzy Neural Networks for On-line Adaptive Phoneme Recognition

Technical Report TR99/03 Department of Information science, University of Otago

Nikola Kasabov, *Member of IEEE*, Michael Watts
Department of Information Science
University of Otago, P.O Box 56, Dunedin, New Zealand
Phone: +64 3 479 8319, fax: +64 3 479 8311
nkasabov@otago.ac.nz, mjwatts@infoscience.otago.ac.nz

May 22, 1999

Abstract

The paper is a study on a new class of spatial-temporal evolving fuzzy neural network systems (EFuNNs) for on-line adaptive learning, and their applications for adaptive phoneme recognition. The systems evolve through incremental, hybrid (supervised / unsupervised) learning. They accommodate new input data, including new features, new classes, etc. through local element tuning. Both feature-based similarities and temporal dependencies, that are present in the input data, are learned and stored in the connections, and adjusted over time. This is an important requirement for the task of adaptive, speaker independent spoken language recognition, where new pronunciations and new accents need to be learned in an on-line, adaptive mode. Experiments with EFuNNs, and also with multi-layer perceptrons, and fuzzy neural networks (FuNNs), conducted on the whole set of New Zealand English phonemes, show the superiority and the potential of EFuNNs when used for the task. Spatial allocation of nodes and their aggregation in EFuNNs allow for similarity preserving and similarity observation within one phoneme data and across phonemes, while subtle temporal variations within one phoneme data can be learned and adjusted through temporal feedback connections. The experimental results support the claim that spatial-temporal organisation in EFuNNs can lead to a significant improvement in the recognition rate especially for the diphthong and the vowel phonemes in English, which in many cases are problematic for a system to learn and adjust in an adaptive way.

1 Introduction

The complexity and the dynamics of real-world problems, especially in engineering and manufacturing, require sophisticated methods and tools for building on-line, adaptive intelligent systems (IS). Such systems should be able to grow as they operate, to update their knowledge and refine the model through interaction with the environment. This is especially crucial when solving AI problems such as adaptive speech and image recognition, multi-modal information processing, adaptive prediction, adaptive on-line control, and intelligent agents on the WWW. Seven major requirements of the present IS as defined in [23, 25] are listed below:

1. IS should learn fast from a large amount of data (using fast training, e.g. one-pass training).

2. IS should be able to adapt incrementally in both real time, and in an off-line mode, where new data is accommodated as it becomes available.

3. IS should have an open structure where new features (relevant to the task) can be introduced at a later stage of the system's operation. IS should dynamically create new modules, new inputs and outputs, new

connections and nodes. This should occur either in a supervised, or in an unsupervised mode, using one modality or another, accommodating data, heuristic rules, text, images, etc. The system should tolerate and accommodate imprecise and uncertain facts or knowledge and refine its own knowledge.

4. IS should be memory-based, i.e. they should keep a reasonable track of information that has been used in the past and be able to retrieve some of it for the purpose of inner refinement, external visualisation, or for answering queries.

5. IS should improve continuously (possibly in a life-long mode) through active interaction with other IS and with the environment they operate in.

6. IS should be able to analyse themselves in terms of behaviour, global error and success; to extract rules that explain what has been learned by the system; to make decisions about its own improvement; to manifest introspection.

7. IS should adequately represent space and time in their different scales; should have parameters to represent such concepts as spatial distance, short-term and long-term memory, age, forgetting, etc.

Several investigations [46, 10, 48]showed that the most popular neural network models and algorithms are not suitable for adaptive, on-line learning. This includes multilayer perceptrons trained with the backpropagation algorithm, radial basis function networks, self-organising maps SOMs [34, 35] and those NN models which were not designed for on-line learning in the first instance. At the same time some of the seven issues above have been acknowledged and addressed in the development of several NN models for adaptive learning and for structure and knowledge manipulation as discussed below.

Adaptive learning is aimed at solving the well-known stability / plasticity dilemma [4, 3, 5]. Several methods for adaptive learning are related to the work presented here, namely incremental learning [12, 11, 40], lifelong learning [24], and on-line learning [9, 48, 47, 15]. Incremental learning is the ability of a NN to learn new data without destroying (or at least fully destroying) the learned patterns from old data, and without a need to be trained on the entirety of the old and new data. On-line learning is concerned with learning data as the system operates (usually in real time) and where the data might exist only for a short time.

Some of the on-line learning methods deal with a fixed structure of the NN [1, 14], other exploit dynamically changing structure of NN through either a structural growth (constructivism) [8, 6, 12], or a structural pruning (selectivism) [45, 13, 16, 21, 36, 37, 42, 53], or both growing and shrinking [2, 24, 49, 39].

The paper further develops and explores spatial-temporal adaptation in evolving fuzzy neural networks (EFuNNs) for the task of on-line adaptive speech recognition. This task is explained and illustrated in section 2. The major principles of spatial temporal EFuNNs are presented in section 3. EFuNNs are used in section 4 for adaptive learning and classification of all the phonemes in New Zealand English. They are compared with other connectionist models. Section 5 introduces a framework for phoneme-based adaptive speech recognition systems. Section 6 suggests directions for further development.

2 Why should the seven major issues be addressed when building adaptive speech recognition systems?

Indeed, why are the above seven issues important for making a further progress is the area of adaptive speech recognition and language acquisition?

Building adaptive speech recognition systems is an important task in the area of spoken language processing [7, 28, 50]. Adaptive speech recognition is concerned with the development of speech recognition systems that:

1. can adapt to new speakers, new pronunciations and new accents;
2. can enlarge their vocabulary of words in an on-line mode;
3. can acquire new languages.

It is well known that there is much variation in the pronunciation of phones of the same phonemes, even when spoken by the same individual. At the same time there are similarities in the pronunciation of phones of different phonemes pronounced by different speakers. This makes the recognition of phonemes a very difficult task. Some of the variability within and similarity between examples of speech data that make the process of adaptive speech recognition difficult are illustrated in the next figures.

In this paper we use data collected from twenty one speakers (eleven females and ten males) of NZ English each pronouncing three times 135 words. The pronounced words, and the forty three English phonemes extracted from them, form the Otago Speech Corpus which is part of a Repository for Intelligent Connectionist-based Information Systems (http://divcom.otago.ac.nz/-infoscience/kel/CBIIS.html). First, we give an illustration of the variability problem within a phone data. Then we illustrate the variability problem of different pronunciation of same speakers and different speakers on the same phoneme. We then illustrate the similarity between different phoneme data. In the paper, data from one male and one female speaker (speakers 17 and 21 from the data base), their first pronunciation, is denoted as set A, their second pronunciation as set B, while data from a second male and female speakers is denoted as set C.

Figure 1 shows thirty two vectors, each of them consisting of three time-lags of twenty six mel-scale coefficients of the phoneme `/e/` data taken from one pronunciation of the word "get" by speaker 17. The speech data is processed as follows: after 512-point FFT is applied, each time frame of 11.6 msec is transferred into twenty six mel-scale elements with a 50% of overlap between consecutive frames. It can be seen that while there is similarity in the mel-scale vector patterns, there are no two vectors that are the same which may cause problems when trying to classify this phoneme data based on these 78-element vector classification.

In Figure 2 ten consecutive frames each of them of twenty six element mel vectors, taken from the phoneme `/e/` data are plotted as lines. This shows a certain pattern similarity across the consecutive time frames, but also a significant difference in the values of the main mel-scale coefficients.

To further stress the fact that the mel-scale coefficients may vary significantly within the same phones, Figure 3 shows this variation on the same phoneme `/e/`data, where ten consecutive values of the main mel-coefficient 2 are taken through their membership degrees to a fuzzy membership function denoting "high" value (when just three triangular membership functions denoting "low", "medium" and "high" are used on the domain values for the second mel-coefficient).

While there are variations within same phoneme data, there is a significant similarity across the phonemes pronounced by the same speaker that makes their recognition difficult. Figure 4 illustrates the "spatial" ambiguity of the monothong and and the diphthong data taken from the set A.

Successful adaptive speech recognition sets the same requirements to the IS as discussed in section 2. In the experiments presented in this paper different connectionist models and systems are experimented with in the same way, i.e.: A system is trained on the set A data and tested on the set A (for a training error), and on sets B and C (for a generalisation error). The system is further trained on set B and tested on set A (forgetting rate), on set B (adaptation error), and on set C (generalisation error). The system is further trained on set C (adaptation error), and recalled on set B and A (forgetting rate). Through this small experiment on three data sets A,B and C we demonstrate how adaptation to new pronunciations and new speakers can be achieved through a spatial-temporal organisation in a new connectionist model called EFuNN, and how it compares on the same task with EFuNNs without spatial-temporal organisation and with other NN models.

We can see from the experimental results that the spatial-temporal EFuNNs, in which the seven major issues have been addressed, are superior to the rest of the connectionist models which address only some of the issues. That gives a clear indication on the future directions for the development of the connectionist models for adaptive speech recognition. The next section gives a brief introduction to the main principles of EFuNNs, while section 4 presents experimental results of using different models for adaptive phoneme recognition on sets A,B and C of phoneme data. Section 5 presents a general framework for building adaptive spatial-temporal EFuNN-based phoneme recognition systems.

3 Spatial-Temporal Evolving Fuzzy Neural Networks EFuNNs

3.1 General principles of EFuNNs

Fuzzy neural networks are connectionist structures that implement fuzzy rules and fuzzy inference [54, 38, 17, 20, 19, 22, 27]. FuNNs represent a class of them [27]. EFuNNs are FuNNs that evolve according to the principles of evolving connectionist systems ECOS [23]. EFuNNs were introduced in [25] where preliminary results were given. Here EFuNNs are further developed in terms of their spatial-temporal rule node allocation and adaptation.

EFuNNs have a five-layer structure, similar to the structure of FuNNs (see Figure 5). But here nodes and connections are created / connected as data comes starting with no nodes in the beginning. An optional short-term memory layer can be used through a feedback connection from the rule (also called, case) node layer. The layer of feedback connections could be used if temporal relationships between input data are to be memorised structurally.

The input layer represents input variables. The second layer of nodes (fuzzy input neurons, or fuzzy inputs) represents fuzzy quantization of each input variable space. For example, two fuzzy input neurons can be used to represent "small" and "large" fuzzy values. Different membership functions (MF) can be attached to these neurons (triangular, Gaussian, etc.). The number and the type of MF can be dynamically modified in an EFuNN which is explained in [26, 30, 32]. New neurons can evolve in this layer if, for a given input vector, the corresponding variable value does not belong to any of the existing MF to a degree greater than a membership threshold. A new fuzzy input neuron, or an input neuron, can be created during the adaptation phase of an EFuNN. The task of the fuzzy input nodes is to transfer the input values into membership degrees to which they belong to the MF.

The third layer contains rule (case) nodes that evolve through supervised / unsupervised learning. The rule nodes represent prototypes (exemplars, clusters) of input-output data associations, graphically represented as an association of hyper-spheres from the fuzzy input and fuzzy output spaces. Each rule node r is defined by two vectors of connection weights - $W_1(r)$ and $W_2(r)$, the latter being adjusted through supervised learning based on the output error, and the former being adjusted through unsupervised learning based on similarity measure within a local area of the problem space. The fourth layer of neurons represents fuzzy quantization of the output variables, similar to the input fuzzy neurons representation. The fifth layer represents the real values of the output variables.

The evolving process can be based on two assumptions:

1. no rule nodes exist prior to learning and all of them are created (generated) during the evolving process; or

2. all the rule nodes exist from the very beginning but they are not connected to the input and output nodes and become connected through the learning (evolving) process.

The latter case is more biologically plausible [44, 18]. The EFuNN evolving algorithm [26] not differentiate between these two cases.

Each rule node r_j represents an association between a hyper-sphere from the fuzzy input space and a hyper-sphere from the fuzzy output space, the $W_1(r_j)$ connection weights representing the co-ordinates of the centre of the sphere in the fuzzy input space, and the $W_2(r_j)$ - the co-ordinates in the fuzzy output space. The radius of an input hyper-sphere of a rule node is defined as $(1 - Sthr)$, where $Sthr$ is the sensitivity threshold parameter defining the minimum activation of a rule node (e.g., r_1) to an input vector (e.g., (Xd_2, Yd_2)) in order for the new input vector to be associated to this rule node. Two pairs of fuzzy input-output data vectors $d_1 = (Xd_1, Yd_1)$ and $d_2 = (Xd_2, Yd_2)$ will be allocated to the first rule node r_1 if they fall into the r_1 input sphere and in the r_1 output sphere, i.e. the local normalised fuzzy difference between Xd_1 and Xd_2 is smaller than the radius r and the local normalised fuzzy difference between Yd_1 and Yd_2 is smaller than an error threshold $Errthr$. The local normalised fuzzy difference between two fuzzy membership vectors d_1f and d_2f that represent the membership degrees to which two real values d_1 and d_2 data belong to the pre-defined MF are calculated as follows:

$$D(d_1f, d_2f) = \frac{\sum \mid d_1f - d_2f \mid}{\sum \mid d_1f + d_2f \mid} \tag{1}$$

For example, if $d_1f = (0,0,1,0,0,0)$ and $d_2f = (0,1,0,0,0,0)$, then $D(d_1, d_2) = \frac{1+1}{2} = 1$ which is the maximum value for the local normalised fuzzy difference.

If data example $d_1 = (Xd_1, Yd_1)$, where Xd_1 and Yd_1 are correspondingly the input and the output fuzzy membership degree vectors, is associated with a rule node r_1 with a centre r_1^1, then a new data point $d_2 = (Xd_2, Yd_2)$, which falls in the r1 input and output hyper-sphere, will be associated with this rule node too. Through the process of associating (learning) of new data points to a rule node, the centres of this node hyper-spheres adjust in the fuzzy input space depending on a learning rate lr_1 and in the fuzzy output space depending on the learning rate lr_2. The adjustment of the centre r_1^1 to its new position r_1^2 can be represented mathematically by the change in the connection weights of the rule node r_1 from $W_1(r_1^1)$ and $W_2(r_1^1)$ to $W_1(r_1^2)$ and $W_2(r_1^2)$ as it is presented as follows:

$$W_2(r_1^2) = W_2(r_1^1) + lr_2 \cdot Err(Yd_1, Yd_2) \cdot A_1(r_1^1),$$
$$W_1(r_1^2) = W_1(r_1^1) + lr_1 \cdot Ds(Xd_1, Xd_2),$$

where: $Err(Yd_1, Yd_2) = Ds(Yd_1, Yd_2) = Yd_1 - Yd_2$ is the signed value rather than the absolute value difference vector; $A_1(r_1^1)$ is the activation of the rule node r_1^1 for the input vector Xd_2.

While the connection weights from W_1 and W_2 capture spatial characteristics of the learned data (centres of hyper-spheres), the temporal layer of connection weights W_3 from Figure 6 captures temporal dependencies between consecutive data examples. If the winning rule node at the moment $(t-1)$ (to which the input data vector at the moment $(t-1)$ was associated) was $r_1 = inda1(t-1)$, and the winning node at the moment t is $r_2 = inda1(t)$, then a link between the two nodes is established as follows:

$$W_3(r_1, r_2)^{(t)} = W_3(r_1, r_2)^{(t-1)} + lr_3 \cdot A_1(r_1)^{(t-1)} A_1(r_2),$$

where: $A_1(r)(t)$ denotes the activation of a rule node r at a time moment (t) lr_3 defines the degree to which the EFuNN associates links between rules (clusters, prototypes) that include consecutive data examples (if $lr_3 = 0$, no temporal associations are learned in an EFuNN).

The learned temporal associations can be used to support the activation of rule nodes based on temporal, pattern similarity. Here, temporal dependencies are learned through establishing structural links. These dependencies can be further investigated and enhanced through synaptic analysis (at the synaptic memory level) rather than through neuronal activation analysis (at the behavioural level). The ratio spatial-similarity / temporal-correlation can be balanced for different applications through two parameters S_s and T_c such that the activation of a rule node r for a new data example d_{new} is defined as the following vector operations:

$$A_1(r) = f(S_s \cdot D(r, d_{new}) + T_c \cdot W_3(r^{(t-1)}, r))$$

where: f is the activation function of the rule node r, $D(r, d_{new})$ is the normalised fuzzy distance value and $r^{(t-1)}$ is the winning neuron at the previous time moment. Figure 7 shows a schematic diagram of the process of evolving of four rule nodes and setting the temporal links between the consecutively evolved nodes based on consecutive frames of phoneme `/e/` data.

3.2 Spatial location of rule nodes in the rule node space

There are different ways to locate rule nodes in an EFuNN rule node space as it is explained here. The type selected depends on the type of the problem the EFuNN is designed to solve. Here five major strategies are explained as experimented with in section 4.

(a) Linear clustering. With this strategy, new nodes are appended to the end of the rule layer, with no heed paid to the spatial position of the example the new node is to represent (see Figure 8). This strategy is useful when there are no spatial relationships in the data, when the training examples for a particular class are to be presented sequentially, or when rule node aggregation is not going to be applied to the trained network.

(b) Maximum Weight clustering: In this strategy, the new node is inserted adjacent to the existing node that most supports the activation of the desired action node (see Figure 9). This is simply determined by finding which rule node has the highest valued weight leading to the desired action node. This strategy is useful for classification tasks, where one action node must be activated above the others. The strategy preserves the spatial characteristics of the data, by inserting new nodes next to those that represent the same class as the new example. It is also useful when aggregation is going to be applied to the network, as clusters of spatially close nodes will be formed, and when training examples for a class are not presented sequentially.

(c) Maximum Node clustering: With this strategy, the node is inserted next to the most highly activated existing node. This strategy is based upon the assumption that the most highly activated rule node will be the one that is the closest spatially to the current example. While this may yet prove to be useful for such tasks as time series prediction, it is no longer used for classification, as experiments have shown that it is not as effective as the Maximum Weight strategy.

(d) Spatial-temporal allocation strategy: as in (b) but temporal feedback connections are set as well (see Figure 10). New connections are set that link consecutively activated rule nodes through using the short term memory and the links established through the W_3 weight matrix; that will allow for the evolving system to repeat a sequence of data points starting from a certain point and not necessarily from the beginning.

(e) Spatial-temporal links are set within, and across, evolved modules: The same as above, but in addition, new connections are established between rule nodes from different EFuNN modules that become activated simultaneously (at the same time moment) (Figure 11). This would make it possible for an EFuNN-based system to learn a correlation between conceptually different variables, e.g. correlation between speech sound and lip movement.

3.3 Learning modes in EFuNN. Rule node aggregation

Different learning, adaptation and optimisation strategies and algorithms can be applied on an EFuNN structure for the purpose of its evolving. These include:

- Active learning , e.g. the EFuNN algorithm;
- Passive learning (i.e., cascade-eco, and sleep-eco learning) as explained in [23].
- Rule insertion into EFuNNs
- Rule extraction
- Rule node pruning
- Rule node aggregation.

Here the rule node aggregation strategy is explained as it is the main structural optimisation strategy exploited in section 4 on the adaptive phoneme recognition task. Each rule node, which represents a prototype, rule, exemplar from the problem space, can be described by its connection weights $W_1(r)$ and $W_2(r)$ that define the association of the two corresponding hyper-spheres from the fuzzy input and the fuzzy output problem spaces. The association is expressed as a fuzzy rule, for example:

```
IF x1 is Small 0.85 and x1 is Medium 0.15
   and x2 is Small 0.7 and x2 is Medium 0.3
THEN y is Small 0.2 and y is Large 0.8
```

The numbers attached to the fuzzy labels denote the degree to which the centres of the input and the output hyper-spheres belong to the respective MF.

3.3.1 Rule Node Aggregation

The process of aggregation of several rule nodes into a larger hyper-sphere is shown in Figure 12 on an example of three rule nodes r_1, r_2 and r_3 (only the input space is shown there). Although several aggregation strategies are in use, each is based upon measuring the spatial distances between the incoming and outgoing weight vectors of the rule two nodes n and m. The distances D_1 (distance between incoming weights) and D_2 (distance between outgoing weights) are calculated by:

$$D_{1n,m} = \frac{\sum_{i}^{C} \mid W_{1i,n} - W_{1i,m} \mid}{\sum_{i}^{C} (W_{1i,n} + W_{1i,m})} \tag{2}$$

and

$$D_{2n,m} = \frac{\sum_{i}^{A} \mid W_{2i,n} - W_{2i,m} \mid}{\sum_{i}^{A} (W_{2i,n} + W_{1i,m})} \tag{3}$$

where C is the number of condition nodes in the network, and A is the number of action nodes.

3.3.2 Aggregation Strategies

Pair Wise Aggregation The most conservative of the strategies investigated, pair wise aggregation will at most halve the number of rule nodes in an EFuNN. It will, for each pair of nodes in a network, examine the distances between them, and aggregate them into one if both D_1 and D_2 are below the set thresholds. If either distance is above the threshold, then both nodes will be preserved unchanged.

Group Wise Aggregation In this strategy, distances between adjacent nodes are measured. While both D_1 and D_2 are below the thresholds, the nodes are added to the aggregation set A_n. When a distance is found that exceeds the threshold (i.e. either D_1 or D_2 are greater than the threshold) all nodes in A_n are aggregated into one node, and the process begun again, starting at the last node not added to A_n.

3.4 The Aggregation Process

The centre of the new node r_{agg} created from the aggregation set A_n is calculated as:

$W_1(r_{agg}) = \overline{(W_1(A_n))}$

$W_2(r_{agg}) = \overline{(W_2(A_n))}$

Here the geometrical centre between two points in a fuzzy problem space is calculated with the use of an average vector operation over the two fuzzy vectors. This is based on a presumed piece-wise linear function between two points within their input and output hyper-spheres of a chosen radius (Figure 12).

Through node creation and consecutive aggregation an EFuNN systems can adjust over time to changes in the data stream. Rule nodes which represent phoneme data clusters, would shift in the phoneme data space with new speakers of different accents talking to the system over time and the system adapting to them. With the learning and pruning operations as part of the EFuNN learning algorithm, and with some additional adaptation techniques, an EFuNN can dynamically organise its structure to learn from data in an adaptive, continuous, incremental, life-long learning mode.

4 Spatial-temporal adaptation in EFuNNs for adaptive phoneme recognition

4.1 EFuNNs for adaptive phoneme recognition

In this section data collected from two groups of one male and one female speaker from the Otago Speech Corpus (see section 1) is used. The first pronunciation of the first two speakers (17 and 21) is denoted as set A, their second pronunciation - as set B, while data from the second group of one male and one female speakers, is denoted as set C. There were 10175 examples in set A, 4955 examples in set B and 7058 rows in set C. The data values were linearly normalised to lie between 0 and 1. Several experiments were carried out as explained below.

4.1.1 Experiment One

Initially forty three EFuNNs, one for each phoneme in the data set, were evolved over a single pass through data set A. The initial EFuNNs each had seventy eight inputs, with each input having three membership functions (representing low, medium and high activation of that input) attached for a total of 234 condition nodes. Each network had a single output with two action nodes (representing positive or negative classification of an example) attached. No temporal connections were present. The following training parameters were used: linear rule node allocation strategy; linear activation functions; $SThr =$

0.5; $lr_1 = lr_2 = 0.75$; $lr_3 = 0$; no pruning; $Errthr = 0.01$. The classification rate was evaluated on data set A (to evaluate the training error), on data set B (to evaluate the generalisation of the EFuNNs over a new articulation by the same speakers), and on data set C (to evaluate the generalisation over new speakers). Then all EFuNNs were further trained for one pass on set B to adapt them to the new articulation data. After the additional training the EFuNNs were tested again on sets A, B and C. The classification rate significantly improved on the sets A, B and C. This experiment shows that EFuNNs can successfully adapt to new pronunciations without forgetting previous ones (on the contrary - the recognition rates over previous examples improves). The EFuNNs were then further adapted to set C and tested again on sets A,B and C. The number of rule nodes after each training session are shown in Table 1. The positive classification accuracies for the vowels `/I/ /e/ /&/` and `/i/` are shown in Table 2. The mean positive classification accuracies across the affricate, approximant, monothong and diphthong phoneme classes are displayed in Table 3. In both cases the results show a successful adaptation of the initially trained EFuNNs on new articulation data and a new set of speakers.

4.1.2 Experiment Two

EFuNNs with temporal connections were evolved over the set A and tested with the set B elements for the phonemes `/p/ /f/ /ch/ /m/ /l/ /I/` and `/el/`. These phonemes are representative of each of the seven classes of phonemes that exist in New Zealand English. There were 1675 examples present in set A and 815 examples in set B. The training parameters were similiar to those in experiment one, with the exception of lr_3 being set to 0.01. The positive classification accuracies across set A and B after training are displayed in table 4. The number of rule nodes evolved for each phoneme are displayed in table 5. Here the initial positive accuracies over both set A and set B are higher than for EFuNN lacking the temporal connections. These results are to be expected after seeing the temporal variations in phonemes in Figure 1.

4.1.3 Experiment Three

As with experiment one, forty three EFuNNs were trained and recalled with each of the three data sets. However, after training each network had the groupwise aggregation algorithm from section 3.3.1 applied. The resulting reduced networks were also recalled with each data set. These aggregated networks were then used as the starting points for further training. The data sets and training parameters were the same as in experiment one. The incoming and outgoing weight distance aggregation thresholds were both set to 0.6

The positive classification accuracies for the vowels `/I/` `/e/` `/&/` and `/i/` are shown in Table 7. The table displays for each recall data set the accuracy across that set for both the trained and aggregated networks. The number of rule nodes present in each of these networks before and after aggregation are displayed in Table 6. Table 8 displays the mean pre- and post-aggregation accuracies for the affricate, approximant, monothong and diphthong phoneme classes.

Figure 13 shows the winning rule nodes (Y-axis) for the `/e/` network after training with set A and recall with the `/I/` `/e/` and `/&/` elements of set B. While definite spatial clustering is visible, the scattering about the main clusters indicates the potential confusion between the three phonemes.

4.2 Experiment Four

The setup for this experiment was the same as for experiment three, with the exception that the Maximum Weight rule node clustering algorithm was used during training. The positive classification accuracies for the same four vowels as before are displayed in Table 10, with the number of rule nodes before and after aggregation shown in Table 9. Mean pre- and post-aggregation accuracies for the four phoneme classes are in Table 11.

Comparison with the results of experiment three shows that maximum weight rule clustering not only reduces the size of aggregated networks (each network is consistently reduced to only two nodes) but also increases the positive classification accuracy.

Figure 14 shows the winning rule nodes (Y-axis) for the `/e/` network after training with set A and recall with the /I/ /e/ and /&/ elements of set B. The degree of seperation between the nodes representing the target phoneme /e/ and it's adjacent (from the data set) phonemes is much greater than with linear node allocation, and may account for the superior performance over set B of this network as compared with experiment three.

4.3 Comparison of EFuNNs with Other Connectionist Models

So that the performance of EFuNN could be compared with more traditional models, several experiments were carried out with data sets A and B. The models investigated were three and four neuron layer MLPs, four neuron layer FuNNs (FuNNs that lack fuzzy outputs), conventional FuNNs, and FuNNs structurally optimised by genetic algorithms. As with the EFuNNs, each manually designed network had seventy eight inputs and one output. The MLPs and FuNNs each had ten neurons in each hidden layer. Bootstrapped backpropagation training was used. Each network was trained for 1000 epochs, with the training set consisting of positive and negative data in a three to one ratio. The training set was rebuilt with fresh examples every ten epochs, and the learning rate and momentum were both set to 0.5.

The algorithm by which FuNNs are structurally optimised by GA is detailed in [31] and [52]. For these experiments, a population size of fifty individuals was used, with a mutation rate of 0.001 and a total run of fifty generations. At the end of each run, the most fit network was extracted and trained with the same parameters as above.

Although these experiments were run across all forty three phonemes, only the positive classification accuracy across data set B for the phonemes /I/ /e/ /&/ and /i/ are displayed here in Table 12.

The mean positive classification accuracy across the affricate, approximant, monothong and diphthong phoneme classes are presented in Table 13.

5 A general framework of an adaptive phoneme-based speech recognition system

The experiments explained in the previous section and the characteristics of the spatial-temporal EFuNNs makes it possible to build adaptive phoneme-based speech recognition systems that adapt to every new accent and new speaker if they fail to recognize them. Many EFuNNs evolve all the time in a life-long learning, on-line, adaptive mode to learn to classify phonemes and other elementary phones. They grow and shrink in an self-organized way reflecting the data structures and patterns in the speech data presented. A block diagram of a system is given in Figure (15). This framework is currently being used for the development of multilingual adaptive speech recognition systems that utilise some principles from [28].

6 Conclusions and directions for further development

The paper presents a theoretical and experimental results that proof the power of the evolving connectionist systems paradigm for wide spread applications in adaptive systems and especially adaptive speech and language systems.

Further development in this area includes building EFuNN-based system for evolving spoken languages and building multi-modal spoken language processing systems [41, 29]. The cortical areas of the human brain that are responsible for the speech and the language abilities of humans evolve through the whole development of an individual [33, 51, 43]. Computer modeling of this process, before its biological, physiological and psychological aspects are made completely known, is an extremely difficult task. It requires flexible techniques for adaptive learning trough an active interaction with a teaching environment. Spatial-temporal EFuNNs with an aggregation procedure could be a good choice in this respect.

7 Acknowledgements

This research is partially supported from a research programme funded by the New Zealand Foundation for Research Science and Technology, contract UOO808.

References

[1] J.S. Albus. A new approach to manipulator control: The cerebellar model articulation controller (cmac). *Transactions of the ASME: Journal of Dyanmic System, Measurement, and Control*, pages 220–227, September 1975.

[2] E. Alpaydin. Gal: networks that grow when they learn and shrink when they forget. Technical Report TR91-032, International Computer Science Institute, Berkley, CA, 1991.

[3] G. Carpenter and S. Grossberg. Art3: Hierarchical search using chemical transmitters in self-organising pattern-recognition architectures. *Neural Networks*, 3(2):129–152, 1990.

[4] G. Carpenter and S. Grossberg. *Pattern recognition by self-organizing neural networks.* MIT Press, 1991.

[5] G. Carpenter, S. Grossberg, J.H. Markuzon, and D.B. Rosen. Fuzzyartmap: A neural network architecture for incremental supervised learning of analog multi-dimensional amps. *IEEE Transations of Neural Networks*, 3(5):698–713, 1991.

[6] H. DeGaris. Circuits of production rule gennets - the genetic programming of artificial nervous systems. In R. Albrecht, Reeves. C., and N. Steele, editors, *Artificial Neural Networks and Genetic Algorithms.* Springer Verlag, 1993.

[7] J. Elman, E. Bates, M. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness (A Connectionist Perspective of Development.* MIT Press, 1997.

[8] C. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D Turetzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532. Morgan Kaufmann, 1990.

[9] J. Freeman and D. Saad. On-line learning in radial basis function networks. *Neural Computation*, 9(7), 1997.

[10] French. Semi-destructive representations and catastrophic forgetting in connectionist networks. *Connection Science*, 1:365–377, 1995.

[11] B. Fritzke. Vector quantization with growing and splitting elasitc net. In *ICANN'93: Proceedings of the Intern. Conf on aritifical neural networks, Amsterdam*, 1993.

[12] B. Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems*, volume 7. Morgan Kaufmann, 1995.

[13] Hassibi and Stork. Second order derivatives for network pruning: Optimal brainsurgeon. In *Advances in Neural Information Processing Systems*, volume 4, pages 164–171. Morgan Kaufmann, 1992.

[14] R. Hecht-Nielson. Counter-propagation networks. In *IEEE First International conference on neural networks, San Diego*, volume 2, pages 19–31, 1987.

[15] T.M. Heskes and B. Kappen. On-line learning processes in artificial neural networks. In *Math. foundation of neural networks*. Elsevier, Amsterdam, 1993.

[16] M. Ishikawa. Structural learning with forgetting. *Neural Networks*, 9:501–521, 1996.

[17] R. Jang. Anfis: adaptive network-based fuzzy inference system. *IEEE Trans. on SYst., Man, Cybernetics*, 23(3):665–685, 1993.

[18] S.R.H. Joseph. *Theories of adaptive neural growth*. PhD thesis, University of Edinburgh, 1998.

[19] N. Kasabov. Adaptable connectionist production systems. *Neurocomputing*, 13(2-4):95–117, 1996.

[20] N. Kasabov. *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. MIT Press, 1996.

[21] N. Kasabov. Investigating the adaptation and forgetting in fuzzy neural networks by using the method of training and zeroing. In *Proceedins of the International Conference on Neural Networks ICNN'96*, volume Plenary, Panel and Special Sessions volume, pages 118–123, 1996.

[22] N. Kasabov. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. *Fuzzy Sets and Systems*, 82(2):2–20, 1996.

[23] N. Kasabov. Ecos: A framework for evolving connectionist systems and the eco learning paradigm. In *Pro. of ICONIP'98, Kitakyushu, Japan, Oct. 1998*, pages 1222–1235. IOS Press, 1998.

[24] N. Kasabov. The ecos framework and the eco learning method for evolving connectionist systems. *Journal of Advanced Computational Intelligence*, 2(6):195–202, 1998.

[25] N. Kasabov. Evolving fuzzy neural networks - algorithms, applications and biological motivation. In Yamakawa and Matsumoto, editors, *Methodologies for the Conception, Design and Application of Soft Computing*, pages 271–274. World Scientific, 1998.

[26] N. Kasabov. Evolving connectionist systems for on-line, knowledge-based learning: principles and applictions. Technical Report TR99/02, Department of Information Science, University of Otago, New Zealand, 1999.

[27] N. Kasabov, J.S. Kim, M. Watts, and A. Gray. Funn/2 - a fuzzy neural network architecutre for adaptive learning and knowledge acquisition. *Information Sciences - Applications*, 101(3-4):155–175, 1997.

[28] N. Kasabov, R. Kozma, R. Kilgour, M. Laws, J. Taylor, M. Watts, and A. Gray. A methodology for speech data analysis and a framework for adaptive speech recognition using fuzzy neural networks and self organising maps. In N. Kasabov and R. Kozma, editors, *Neuro-fuzzy techniques for intelligent information systems*. Physica Verlag (Springer Verlag), 1999.

[29] N. Kasabov, E. Postma, and J. Van den Herik. Avis: A connectionist-based framework for integrated audio and visual information processing. In *Proceedings of Iizuka'98, Iizuka, Japan, Oct. 1998*, 1998.

[30] N. Kasabov and Q. Song. Dynamic, evolving fuzzy neural networks with 'm-out-of-n' activation nodes for on-line adaptive systems. Technical Report TR99/04, Department of Information Science, University of Otago, New Zealand, 1999.

[31] N. Kasabov and M. Watts. Geentic algorithms for structural opsimisation, dynamic adaptation and automated design fo fuzzy neural networks. In *Proceedings of the International Conference on Neural Networks ICNN'97*. IEEE Press, 1997.

[32] N Kasabov and B. Woodford. Rule insertionand rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems. In *Proceedings of International Conference FUZZ-IEEE, Seoul, August 1999*, 1999.

[33] S.B. Kater, N.P. Mattson, C. Cohan, and J. Connor. Calcium regulation of the neuronal cone growth. *Trends in Neuroscience*, 1988.

[34] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1497, 1990.

[35] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 2 edition, 1997.

[36] A. Krogh and J.A. Hertz. A simple weight decay can imprive generalisation. In *Advances in Neural Information Processing Systems*, volume 4, pages 951–957. 1992.

[37] Y. Le Cun, J.S. Denker, and S.A. Solla. Optimal brain damage. In D.S. Touretzky, editor, *Advances in Neural Infromation Processing Systems*, volume 2, pages 598–605. Morgan Kaufmann, 1990.

[38] C.T. Lin and C.S.G. Lee. *Neuro-Fuzzy Systems*. Prentice Hall, 1996.

[39] M. Maeda, H. Miyajima, and S. Murashima. A self organizing neural network with creating and deleting methods. *Nonlinear theory and its applications*, 1:397–400, 1996.

[40] J. Mandziuk and L. Shastri. Incremental class learning approach and its application to hand-written digit recognition. In *Proceedings of the fifth International Conference on neuro-information processing, Kitakyushu, Japan, Oct. 21-23*, 1998.

[41] J. Massaro and M. Cohen. Integration of visual and auditory information in speech perception. *Journal of Experimental Psychology: Human Perception and Performance*, 9:753–771, 1983.

[42] D. Miller, J. Zurada, and J.H. Lilly. Pruning via dynamic adaptation of the forgetting rate in structural learning. In *Poc. IEEE ICNN'96*, volume 1, page 448, 1996.

[43] Van Ooyen and J. Van Pelt. Activity-dependent outgrowth of neurons and overshoot phenomena in developing neural networks. *Journal of Theoretical Biology*, (167):27–43, 1994.

[44] R. Port and T. van Gelder. *Mind as Motion (Explorations in the Dynamics of Cognition*. MIT Press, 1995.

[45] R. Reed. Pruning algorithms - a survey. *IEEE Trans. Neural Networks*, 4(4):740–747, 1993.

[46] A. Robins and M. Freans. Local learning algorithms for sequential learning tasks in neural networks. *Journal of Advanced Computational Intelligence*, 2(6), 1998.

[47] G.A. Rummery and M. Niranjan. On-line q-learning using connectionist systems. Technical report, Cambridge University Engineering Department, 1994.

[48] D. Saad, editor. *On-line learning in neural networks.* Cambridge University Press, 1999.

[49] A Sankar and R.J. Mammone. Growing and pruning neural tree networks. *IEEE Trans Comput.*, 42(3):291–299, 1993.

[50] S.J. Segalowitz. *Language functions and brain organization.* Academic Press, 1993.

[51] A. Seleverston, editor. *Model neural networks and behaviour.* Plenum Press, 1985.

[52] M. Watts and N. Kasabov. Genetic algorithms for the design of fuzzy neural networks. In *Proceedings of ICONIP'98, Kitakyushu, Oct. 1998*, pages 793–795. IOS Press, 1998.

[53] D. Whitley and C. Bogart. The evolution of connectivity: Pruning neural networks using genetic algorithms. In *Proceedings International Joint Conference Neural Networks*, pages 17–22, 1990.

[54] T. Yamakawa, H. Kuzanagi, E. Uchino, and T. Miki. A new effective algorithm for neo fuzzy neuron model. In *Proceedings of Fifth IFSA World Congress*, pages 1017–1020, 1993.

Table 1: Number of rule nodes after each training pass for experiment one

Training Set	Phoneme			
	/I/	/e/	/&/	/i/
A	3733	3777	3755	3785
B	5688	5791	5757	5778
C	7786	7925	7880	7973

Table 2: Positive recall accuracies for experiment one

Training Set	Phoneme				Recall Set
	/I/	/e/	/&/	/i/	
A	78	83	89	79	A
	58	69	78	52	B
	22	19	63	23	C
B	81	82	86	77	A
	64	77	91	73	B
	17	25	58	27	C
C	80	83	85	75	A
	64	77	90	76	B
	97	94	93	95	C

Table 3: mean phoneme class positive classification accuracies for experiment one

Training Set	Phoneme Class				Recall Set
	Affricates	Approximant	Monothong	Dipthong	
A	87	74	82	75	A
	66	43	64	57	B
	16	9	29	6	C
B	83	70	81	74	A
	99	57	75	66	B
	19	15	29	7	C
C	83	67	80	73	A
	97	57	75	66	B
	77	91	93	93	C

Table 4: Positive recall accuracies for experiment two

Training Set	Phoneme							Recall Set
	`/p/`	`/f/`	`/ch/`	`/m/`	`/l/`	`/I/`	`/el/`	
A	100	100	98	99	99	80	96	A
	96	100	99	86	98	74	97	B

Table 5: Number of rule nodes after training with temporal connections enabled

Training Set	Phoneme						
	`/p/`	`/f/`	`/ch/`	`/m/`	`/l/`	`/I/`	`/el/`
A	1445	1695	1696	1579	1610	1640	1944

Table 6: Number of rule nodes before and after aggregation for experiment three

Training Set	Phoneme				
	/I/	/e/	/&/	/i/	
A	3711	3761	3734	3761	trained
	2	2	2	2	aggregated
B	1851	1871	1874	1888	trained
	4	4	4	4	aggregated
C	2212	2228	2219	2242	trained
	6	6	6	6	aggregated

Table 7: Positive recall accuracies for experiment three

Training Set	Phoneme				Recall Set
	/I/	/e/	/&/	/i/	
A	84	90	93	81	A
	53	71	60	77	
	58	69	78	50	B
	59	65	51	66	
	23	19	63	23	C
	77	47	79	75	
B	60	64	76	51	A
	55	77	69	60	
	92	81	91	89	B
	61	73	57	55	
	15	17	52	36	C
	73	54	93	56	
C	38	65	56	34	A
	58	91	68	94	
	39	65	43	36	B
	59	89	57	89	
	97	94	96	95	C
	84	90	91	98	

Table 8: mean phoneme class positive classification accuracies for experiment three

Training Set	Phoneme Class				Recall Set
	Affricates	Approximant	Monothong	Diphthong	
A	79	78	86	80	A
	95	67	70	57	
	58	45	64	56	B
	97	67	65	55	
	13	10	29	6	C
	45	55	75	27	
B	46	48	62	57	A
	62	75	73	72	
	92	87	89	93	B
	64	76	69	70	
	19	18	30	9	C
	25	76	77	36	
C	20	23	48	18	A
	62	70	76	70	
	28	29	45	19	B
	65	70	73	69	
	78	84	93	93	C
	71	83	86	63	

Table 9: Number of rule nodes before and after aggregation for experiment four

Training Set	Phoneme				
	/I/	/e/	/&/	/i/	
A	3718	3766	3739	3767	trained
	2	2	2	2	aggregated
B	1851	1874	1871	1890	trained
	2	2	2	2	aggregated
C	2212	2218	2216	2254	trained
	2	2	2	2	aggregated

Table 10: Positive recall accuracies for experiment four

Training Set	Phoneme				Recall Set
	/I/	/e/	/&/	/i/	
A	79	84	91	78	A
	57	75	64	96	
	58	69	78	52	B
	65	68	54	93	
	23	19	63	23	C
	83	51	87	99	
B	60	65	73	51	A
	57	74	70	97	
	73	76	80	75	B
	64	68	58	94	
	15	17	51	36	C
	83	49	94	99	
C	37	57	52	34	A
	29	100	62	78	
	39	58	44	35	B
	34	100	54	76	
	97	93	97	95	C
	83	100	87	90	

Table 11: mean phoneme class positive classification accuracies for experiment four

Training Set	Phoneme Class				Recall Set
	Affricates	Approximant	Monothong	Diphthong	
A	72	70	82	73	A
	99	84	76	66	
	58	44	65	56	B
	100	85	71	64	
	13	10	29	6	C
	63	71	81	31	
B	46	49	63	59	A
	97	85	77	77	
	49	64	76	65	B
	100	85	74	77	
	19	16	28	9	C
	54	72	82	38	
C	21	20	44	13	A
	97	76	62	32	
	28	25	42	15	B
	100	74	59	31	
	78	86	93	93	C
	94	81	84	72	

Table 12: mean positive classificaiton accuracy for the comparison networks

	/I/	/e/	/&/	/i/
Three layer MLP	82	92	91	87
Four layer MLP	76	89	91	82
Four layer FuNN	69	89	80	59
FuNN	64	85	57	15
GA-FuNN	72	89	82	62

Table 13: mean phoneme class classification accuracies for the comparison networks

	Phoneme Class			
	Affricates	Approximant	Monothong	Diphthong
Three Layer MLP	92	74	84	77
Four Layer MLP	89	71	84	77
Four Layer FuNN	28	18	73	46
FuNN	70	20	61	22
GA-FuNN	86	58	74	48

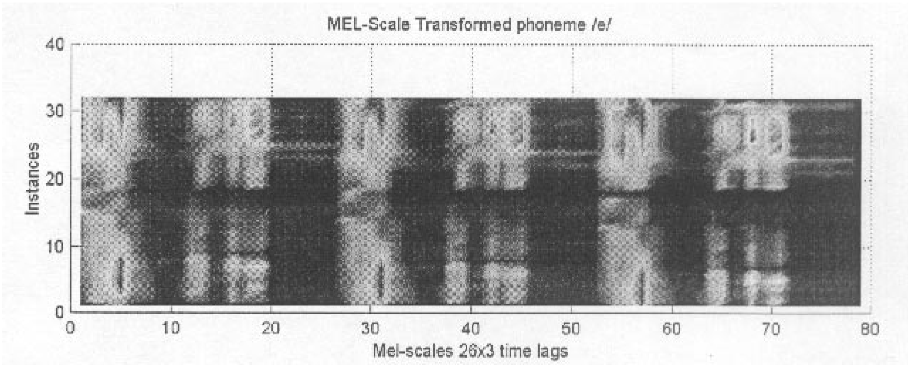

Figure 1: 32 mel-scale vectors each of them of 76 mel-coefficients taken from an individual pronunciation of the phoneme /e/

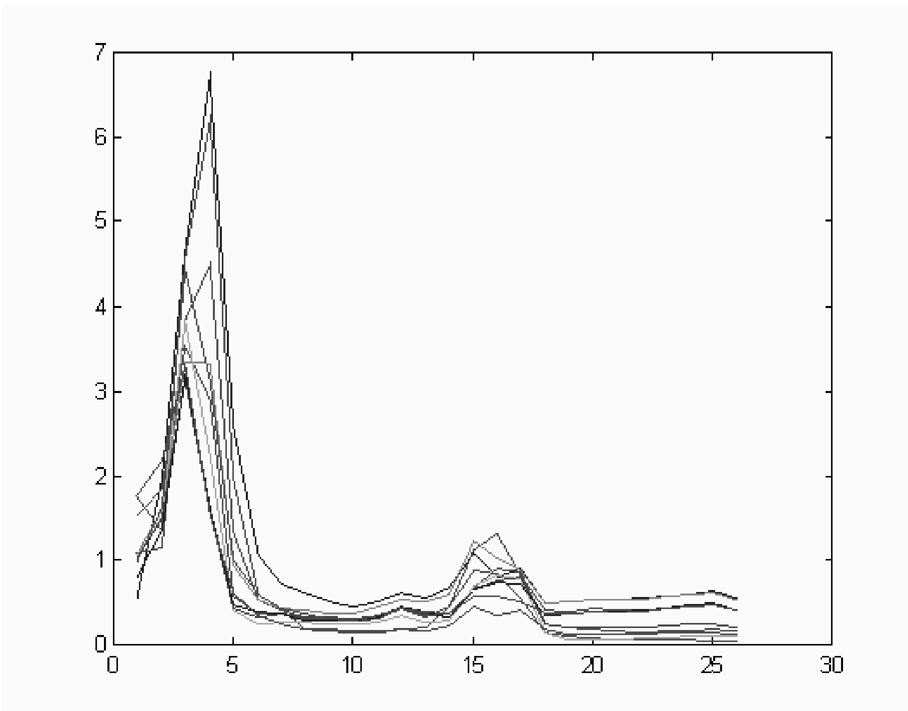

Figure 2: 10 consecutive frames of the phoneme /e/

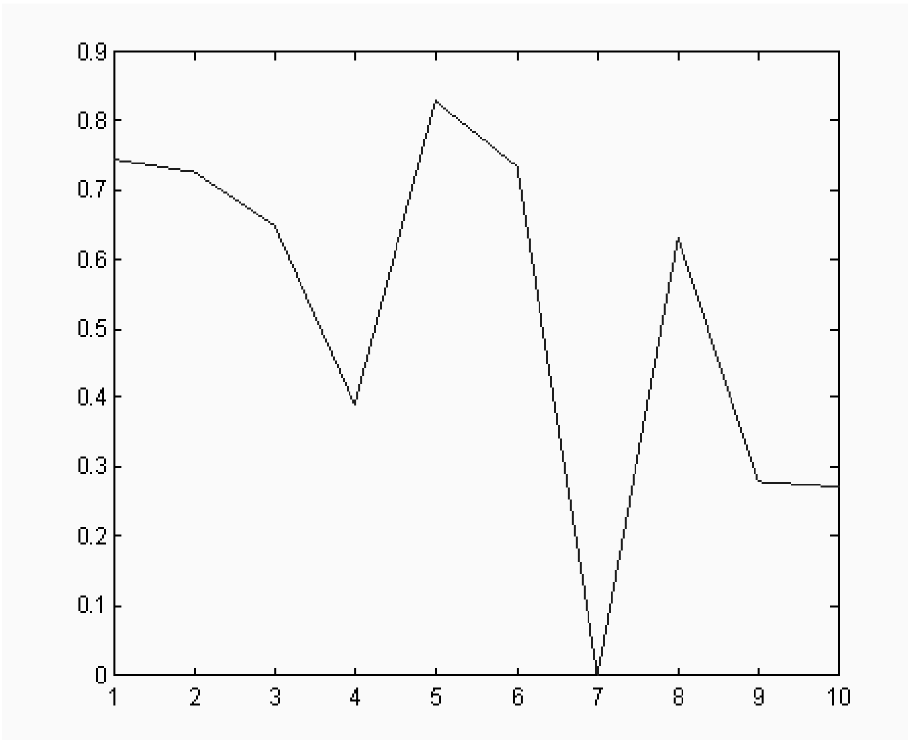

Figure 3: 10 consecutive values of mel-coefficient 2 taken from the phoneme /e/

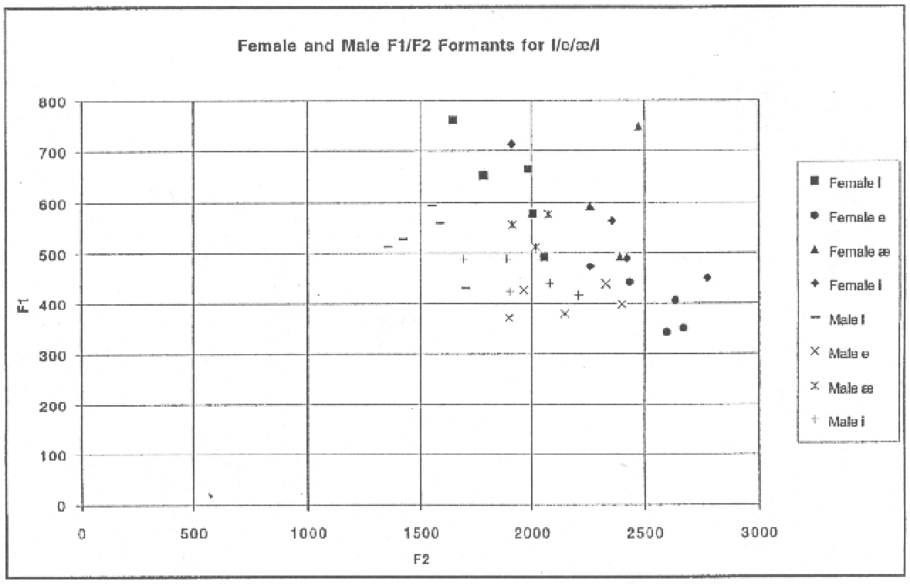

Figure 4: Averaged phoneme vectors of monothongs and diphthongs pronounced by one male and one female speaker and plotted in a two dimensional formant space

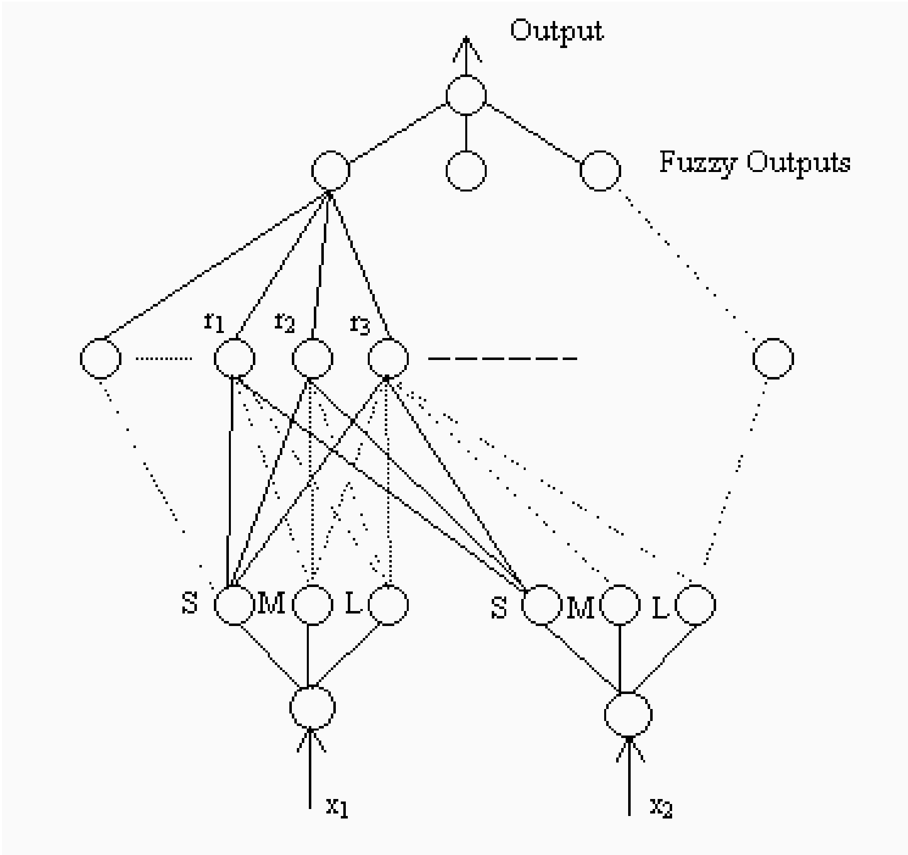

Figure 5: General structure of an EFuNN network

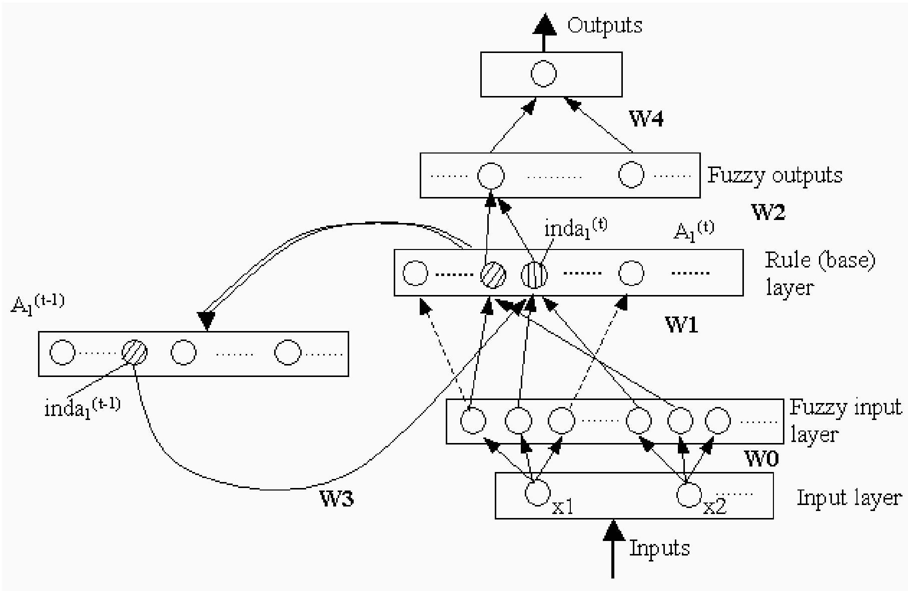

Figure 6: General structure of an EFuNN network showing temporal feedback connections

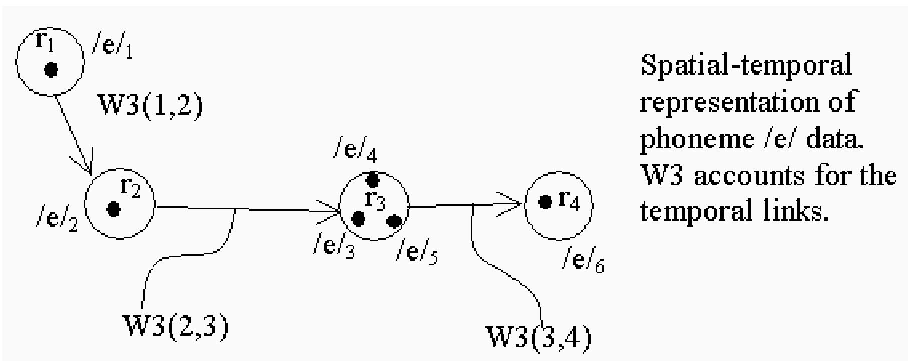

Figure 7: The evolving process of four rule nodes from phoneme /e/ data

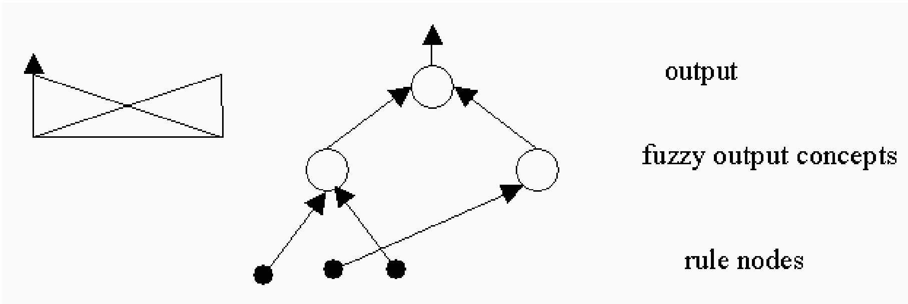

Figure 8: Linear rule node clustering

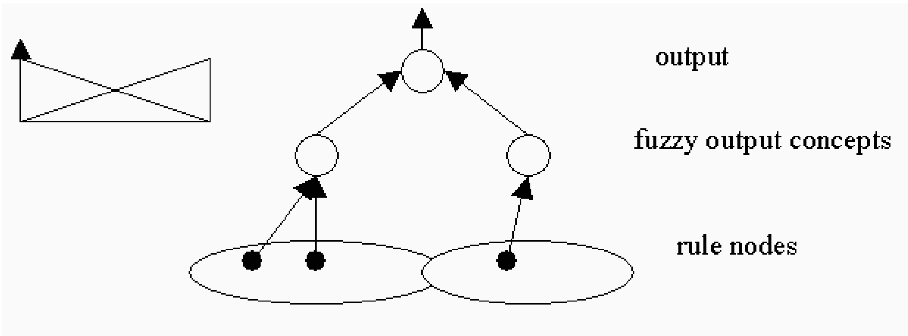

Figure 9: Spatial rule node clustering

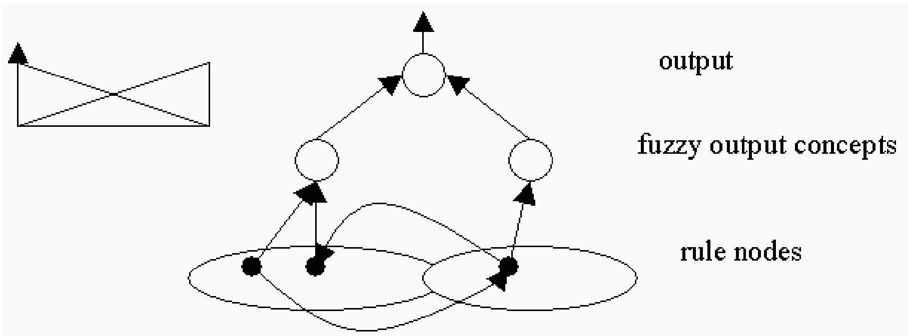

Figure 10: Spatial-temporal rule node clustering

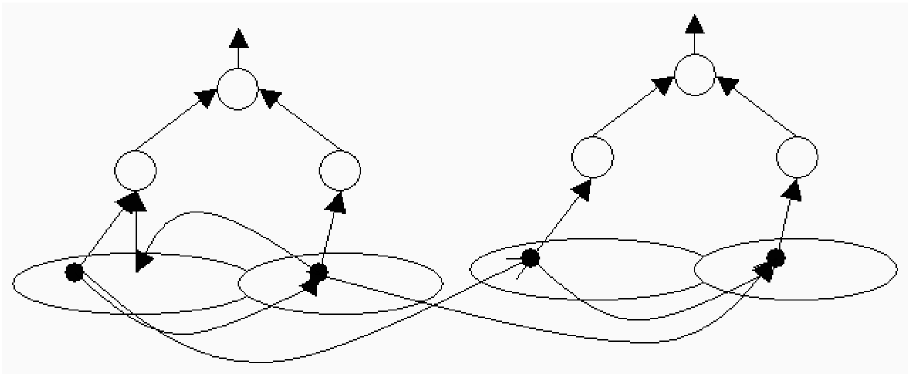

Figure 11: Inter-module rule node clustering

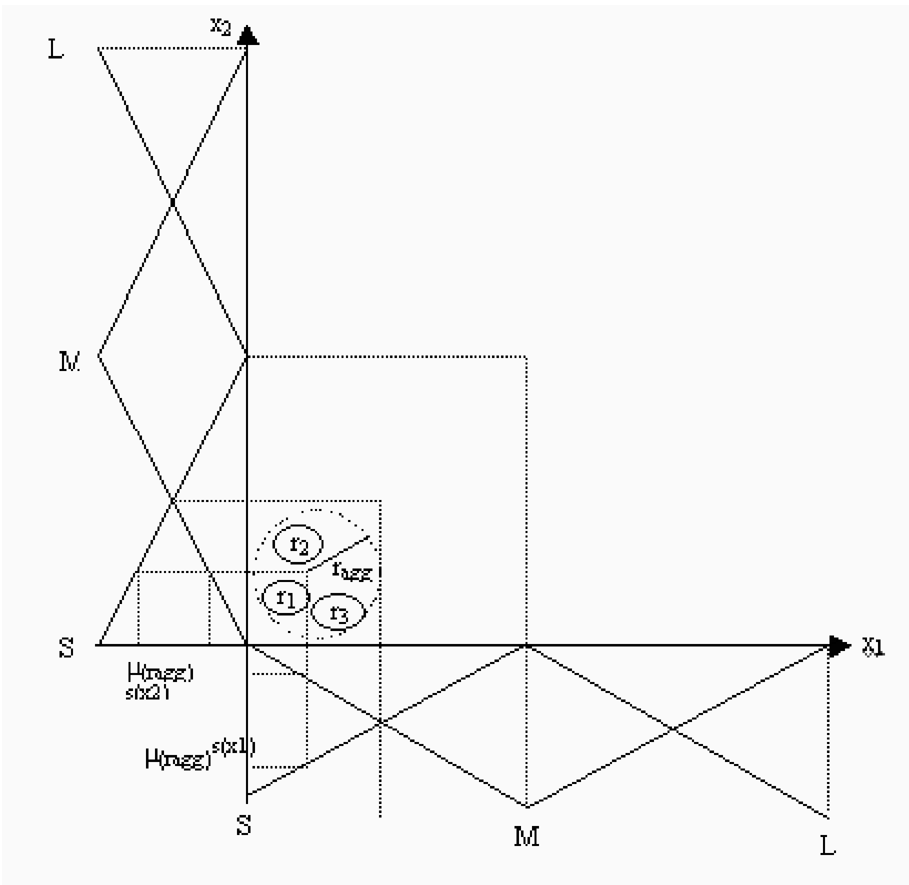

Figure 12: Rule node aggregation

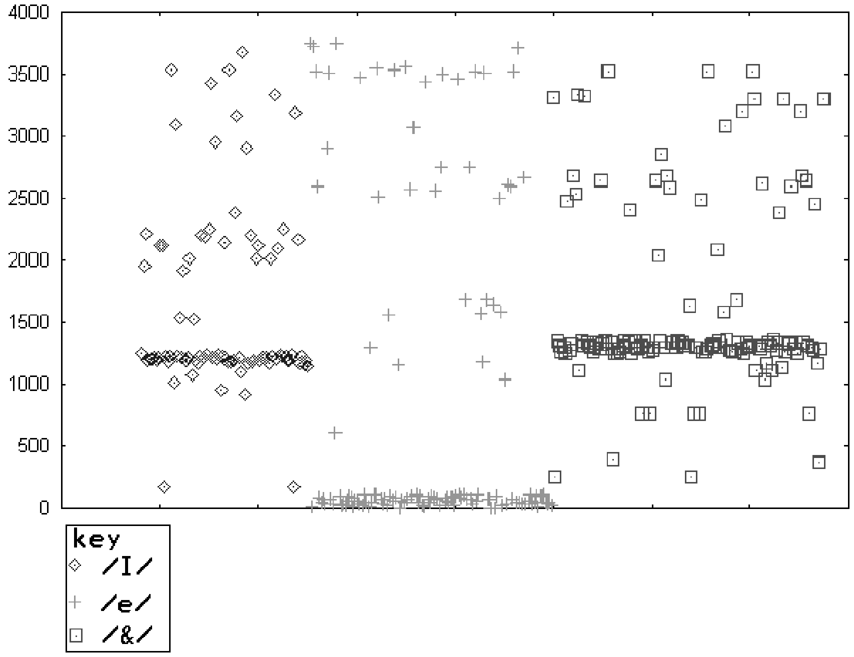

Figure 13: Winning nodes for phonemes /I/, /e/ and /&/ for experiment three

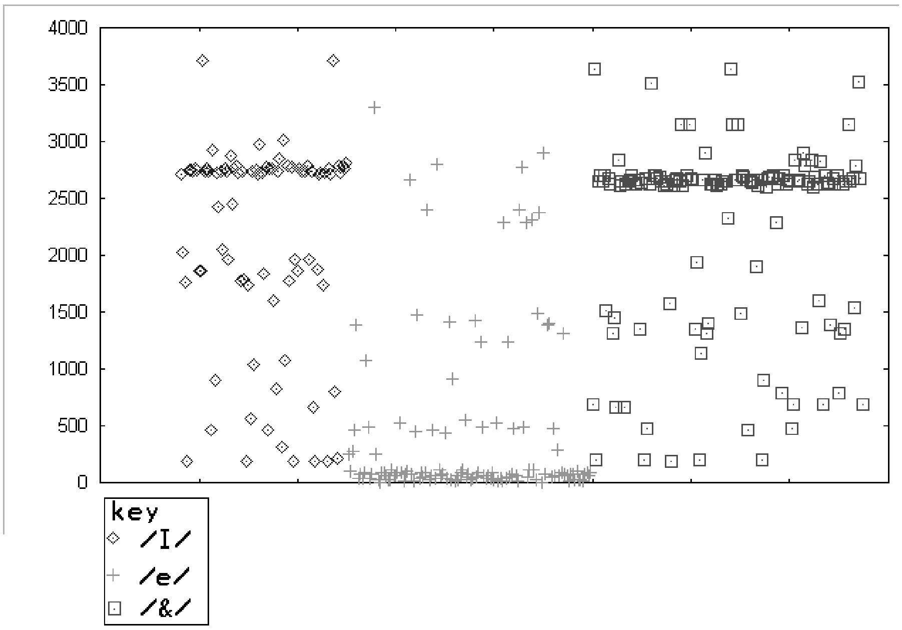

Figure 14: Winning nodes for phonemes /I/, /e/ and /&/ for experiment four

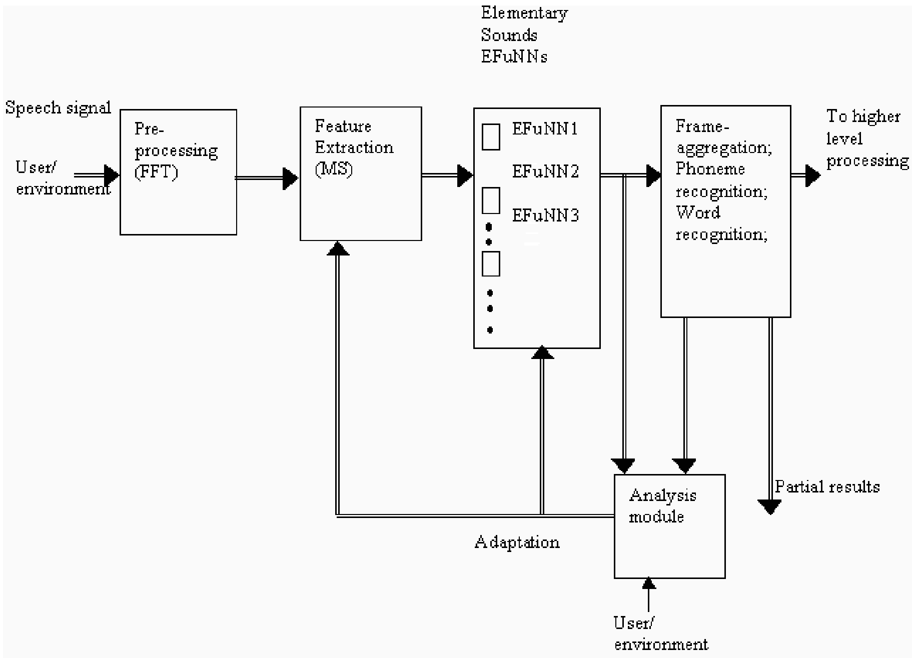

Figure 15: Adaptive phoneme based speech recognition system