ELSEVIER

Fuzzy Sets and Systems 103 (1999) 349–367

FUZZY sets and systems

From hybrid adjustable neuro-fuzzy systems to adaptive connectionist-based systems for phoneme and word recognition

N.K. Kasabov *, R.I. Kilgour, S.J. Sinclair

Department of Information Science, University of Otago, P.O. Box 56, Dunedin, New Zealand

Received May 1998

Abstract

This paper discusses the problem of adaptation in automatic speech recognition systems (ASRS) and suggests several strategies for adaptation in a modular architecture for speech recognition. The architecture allows for adaptation at different levels of the recognition process, where modules can be adapted individually based on their performance and the performance of the whole system. Two realisations of this architecture are presented along with experimental results from small-scale experiments. The first realisation is a hybrid system for speaker-independent phoneme-based spoken word recognition, consisting of neural networks for recognising English phonemes and fuzzy systems for modelling acoustic and linguistic knowledge. This system is adjustable by additional training of individual neural network modules and tuning the fuzzy systems. The increased accuracy of the recognition through appropriate adjustment is also discussed. The second realisation of the architecture is a connectionist system that uses fuzzy neural networks FuNNs to accommodate both a prior linguistic knowledge and data from a speech corpus. A method for on-line adaptation of FuNNs is also presented.

Keywords: Pattern recognition; Artificial intelligence; Neural networks; Speech recognition

1. Introduction: The problem of adaptive speech recognition

Speech recognition is an extremely difficult task to be performed by a computer system. This is because of the variability in the way people speak [2,3,26,28], which results in complex speech signals that have to be processed by automatic speech recognition systems (ASRS). There are several key areas of research which have been pointed out in [3] as significant for the current development of spoken language systems. These are: robust speech recognition, automatic training and adaptation, spontaneous speech, dialogue models, natural language response generation, speech synthesis and speech generation, multilingual systems, and interactive multi-modal systems. A spoken language system, as defined in [2], combines speech recognition, natural language processing, and human interface technology. There are now systems that work reasonably well on continuous and spontaneous speech, although in a very restricted domain.

We take the view that the above goals can best be achieved if an integrated approach is used, i.e.

* Corresponding author. Tel.: +64 3 479 8319; fax: +64 3 479 8311; e-mail: nkasabov@otago.ac.nz.

0165-0114/99/$ – see front matter
PII: S0165-0114(98)00233-4

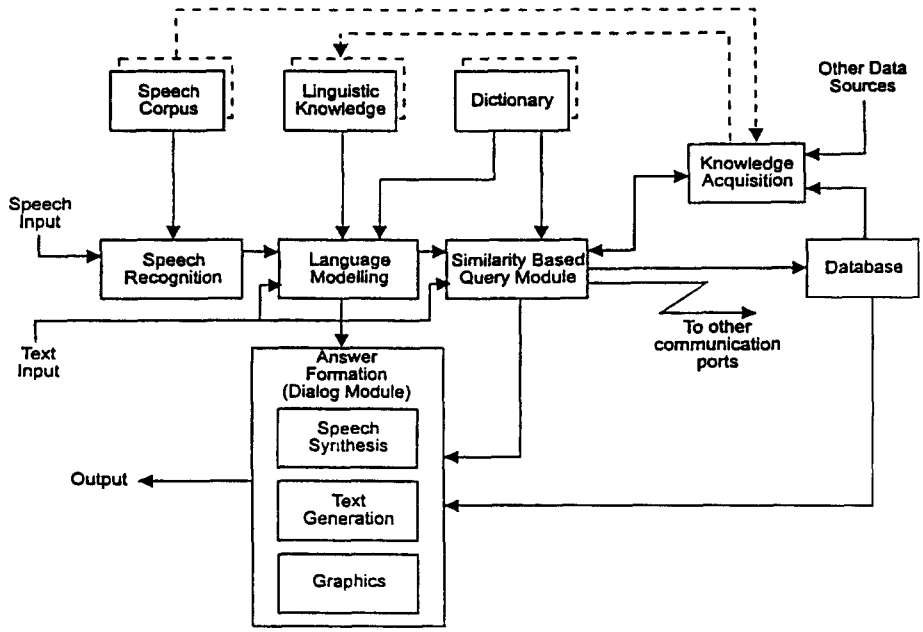


Fig. 1. A general block diagram of an Intelligent Human Computer Interface (from [12]).

that everything about the speech recognition task (a priori known, or acquired during the operation of the system, knowledge) should be used in the system [8–10,12]. For example, speech corpus data [28,34], phonetic rules [2], linguistic knowledge [26], AI methods [29], skills from pedagogy and teaching languages [2], should be brought together and used in one system. Advanced knowledge engineering techniques are needed to facilitate this integrated approach to building ASRS [11,12,14,15,19].

The task of speech recognition becomes more complicated when the ASRS is used as a part of an intelligent human computer interface (IHCI). A general block diagram of IHCI is graphically depicted in Fig. 1. The system allows for retrieving information from a database or for connecting to other communication ports by using both speech and text. It consists of the following major blocks (as described in [12]):

- Speech recognition and language modelling blocks.
- Similarity-based query block. This module carries out approximate reasoning over a user's query and allows for vague, fuzzy queries.
- Knowledge acquisition block. This module performs knowledge acquisition, e.g. extraction of rules from raw data. The module can be used for explanation purposes. Different rule extraction algorithms can be applied [12,13,25].
- Answer formation block. This module produces the answer to the user and performs a dialogue at any phase of the information retrieval. It has both speech synthesis and text generation sub-modules.

The task of speech recognition becomes more complicated when the ASRS is required to adapt to new data and to accommodate new knowledge as they become available. The question is how to tune such a complex system that consists of many units, modules and blocks linked together, for a better performance in an always changing environment in the presence of a huge variability of input data. Can an ASRS adapt to new accents and new speakers as it works, i.e. on-line, "on the fly". This is the major research problem this paper is concerned with.

Building adaptive speech recognition systems, where the system is able to adapt to new accents and new speakers, is an extremely difficult task for computers to achieve, but humans can do this very well. We listen to a new accent for some time and then adapt our perception to improve our recognition and understanding. How the human brain achieves such adaptation is still not known. Speech signals

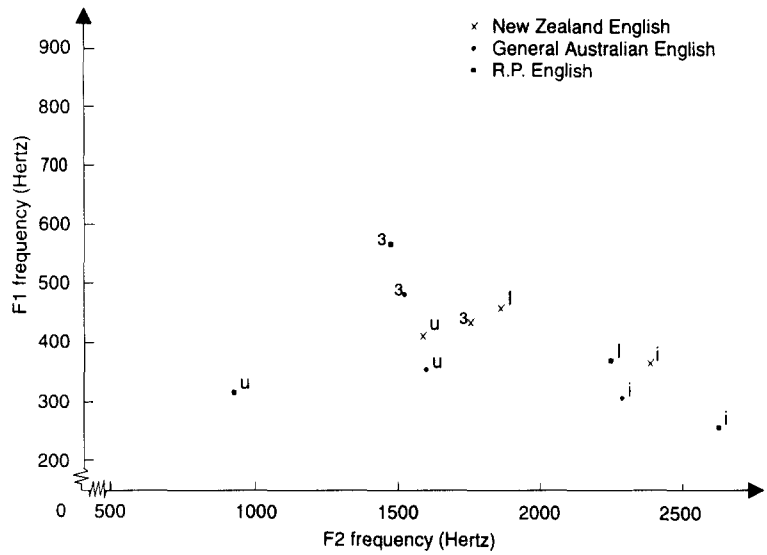


Fig. 2. The first two formants of vowels in New Zealand and Australian English.

are processed in different parts of the brain, that can be aggregated in two fuzzy clusters for "low level processing" and for "higher level processing". Apparently adaptation happens at each part of the auditory pathway from the cochlea to the Broca's and the Wernike's areas in a correlated way [22]. Every human being adapts differently in terms of time required, selected and preferred phonemes, words and phrases, known meaning, etc. Can similar results be achieved in an ASRS?

Different aspects of adaptation in ASRS have been discussed in several papers where different approaches have been explored. Mainly statistical and probabilistic models, e.g. Hidden Markov Models, have been explored [6,31,32]. Connectionist methods that have been developed for on-line adaptation (see for example [30]) need to be experimented on speech recognition problems. The problem of adaptation in ASRS has been around for many years but now its solution becomes feasible due to some properties of the connectionist and the hybrid connectionist systems [1,12,26,36].

In this paper, several principles are explored and several strategies are suggested that led to a better performance of the experimented two realisations of a general architecture of an adaptive ASRS. The first realisation, HySpeech/1, is a hybrid neuro-fuzzy system that allows for selective adjustment of parameters of individual neural network and fuzzy system modules that comprise the system. The second realisation, HySpeech/2, is a connectionist system that is built with the use of fuzzy neural networks FuNNs [12,17]. It allows for automatic adaptation of individual phoneme neural network modules.

The main principles of the architecture explored here are modularity and local specialisation. Each phoneme classifier is realised as a separate phoneme unit, so phoneme units can be adjusted/adapted/tuned individually [12,20]. Very often an ASRS needs to adapt only to a few differently pronounced sounds that cause problems in the overall recognition process. If we look at the fundamental frequencies of the typically pronounced vowels in Australian and New Zealand English, we notice a few differences. It may be the case that only a few vowels need to be adjusted before a system, initially trained on New Zealand English, begins to correctly recognise Australian English (see Fig. 2).

2. Hybrid neuro-fuzzy systems for phoneme and word recognition

The following are the key principles used in the conceptual design of the general architecture for

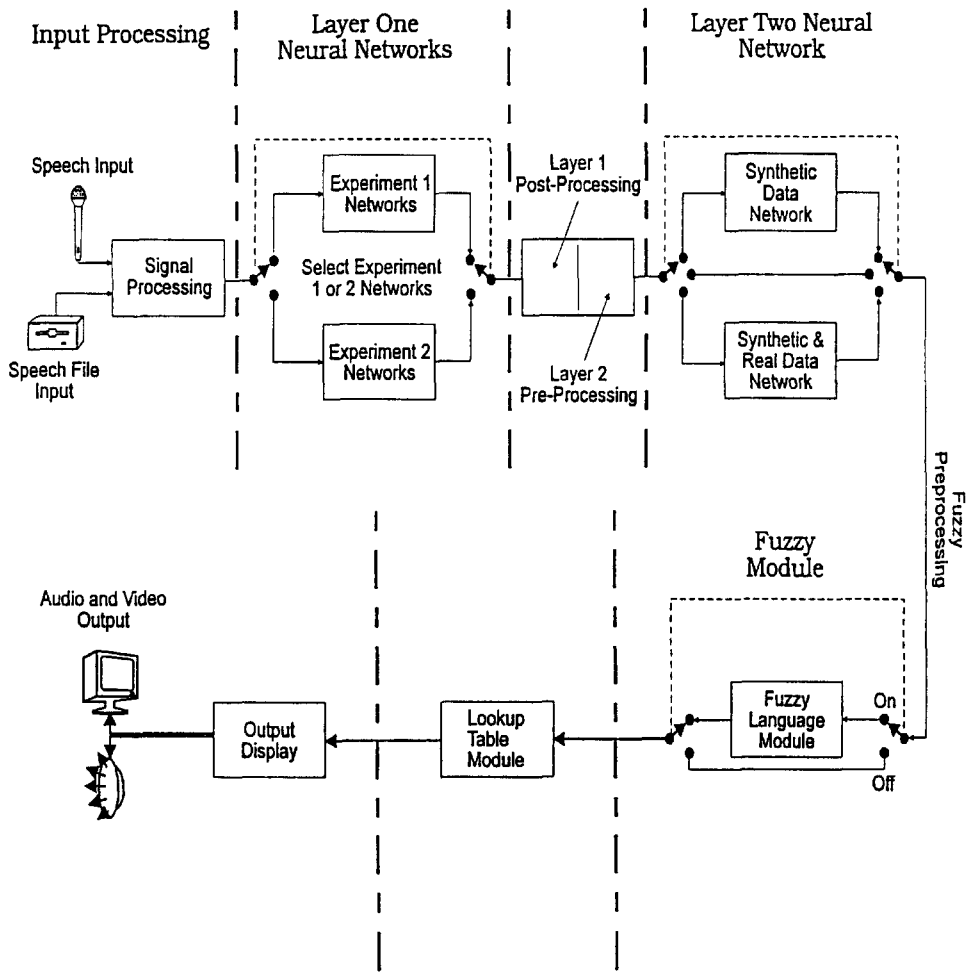


Fig. 3. A block diagram of HySpeech/1 – a modular hybrid neuro-fuzzy system for phoneme-based speech recognition.

adaptive speech recognition, realised as HySpeech/1 and HySpeech/2 systems:

(1) *Mixing training data and explicit knowledge in one system.* The system should be flexible and should use all sources of information available on the problem.

(2) *Extendibility.* The system should be easily extendible by adding new items to the speech corpus, adding new linguistic knowledge, and adding new words to the dictionary according to a concrete application.

(3) *Modularity and local specialisation.* Individual modules are assigned for classifying each of the elementary sounds (phonemes) at each level of recognition.

(4) *Hierarchical structure and hierarchical organisation of the adaptation process.* A module in the hierarchy adapts according to the performance of the whole system as well as according to the data provided by the module that is functionally preceding it.

HySpeech/1 is an experimental realisation of the above principles as a speaker independent system for recognising pronounced in isolation words. In the concrete experiment the digit words of New Zealand English are used. This implementation uses standard

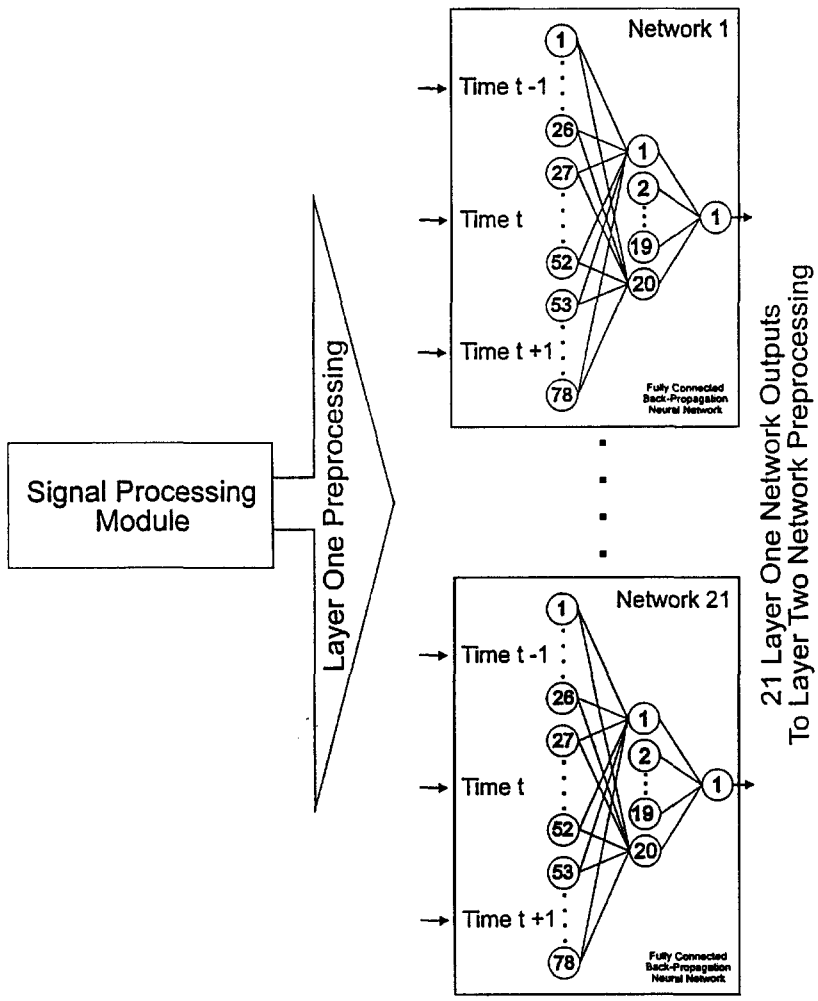


Fig. 4. A block diagram of the first layer neural network module (from [33]).

feed-forward neural networks for the phoneme recognition module and fuzzy rule-based inference systems for representing phonotactic rules, as shown in the block diagram of Fig. 3 [20,21,33]. It consists of the following modules:

(1) *Speech pre-processing module.* This module transforms raw speech signals into feature vectors. 26 Mel-scale cepstrum coefficients (MSCC) are used as feature vectors to represent each time frame after 22.050 kHz, 16-bit resolution sampling of the speech signal. MSCC are cosine transformations on Mel-scale central frequencies that form a set of filters considered to be close to the way the human inner ear perceives and filters speech sounds.

(2) *A hierarchical, two-layer, multi-modular connectionist system for phoneme recognition.* This is a neural network based block for the recognition of New Zealand English phonemes. The first-layer neural network module consists of 21 neural networks (phoneme units), one for each of the phonemes participating in the spoken words, plus the silence phoneme. Each phoneme unit has: 3×26 inputs, where three consecutive time-frame MSCC vectors are supplied; 20 hidden nodes; one output node which represents the corresponding phoneme class. Fig. 4 depicts the first-layer neural network module. All the neural networks are trained with real phoneme data from a speech corpus of New Zealand English [33,34]. The corpus

Table 1
Details of the Otago Speech Corpus (see also [34])

	Digit collection	Word collection	Total
Number of words	10	129	139
Instances of each word	3	3	–
Number of female speakers	10	10	20
Number of male speakers	11	12	23
Total utterances recorded	630	8514	9144
Number of phonemes segmented	1953	8514	10467

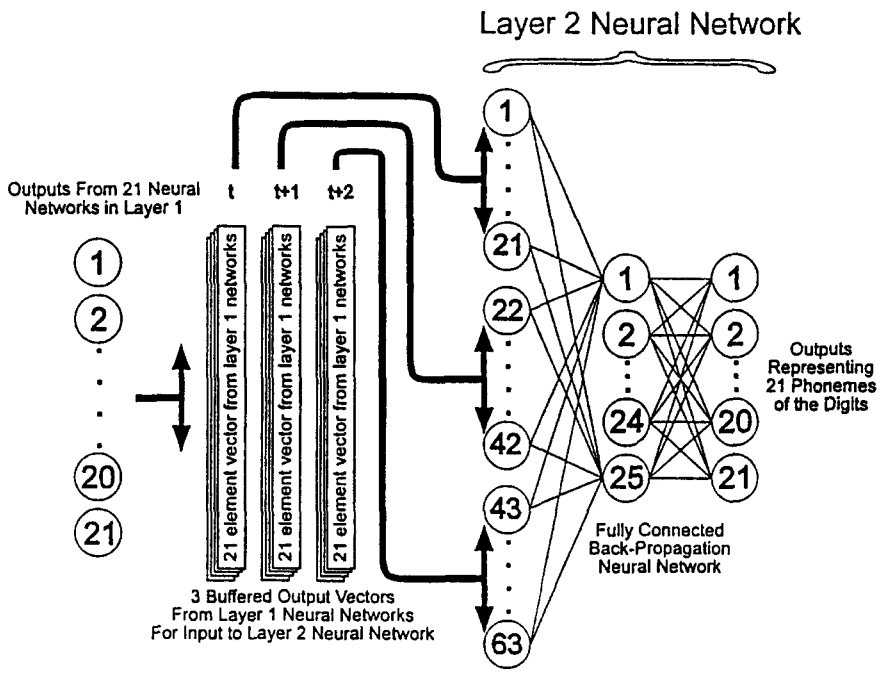


Fig. 5. A block diagram of the second layer neural network module (from [33]).

contains phoneme realisations from 11 male speakers and 10 female speakers of NZ English extracted from a set of 139 words pronounced three times by each of the speakers (see Table 1). It is available from: http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/speech.htm.

The second-layer neural network module is a single feed-forward neural network which takes three, time-consecutive, 21-element output vectors from the first-layer neural networks (the phoneme units) and produces a corresponding 21 element vector (see Fig. 5). This network performs an aggregation of classified phonemes over three time intervals. The network is trained using a standard backpropagation algorithm, on both real data and synthetic data.

(3) *A language modelling block based on fuzzy inference.* The module has two sub-modules. The first sub-module is a fuzzy rule-based system. Fuzzy rules can represent the certainty that a given phoneme has happened when certain phoneme has preceded it (has been recognised in the previous time frame). Different sets of fuzzy rules are used for the three main parts of each syllable (onset, syllabic and coda). An example is shown in Fig. 6 where the block diagram of the fuzzy system for recognising the phoneme /*s*/ is given as an illustration. Here, two consecutive phonemes are considered, denoted as $p1$ and $p2$. The fuzzy rules infer

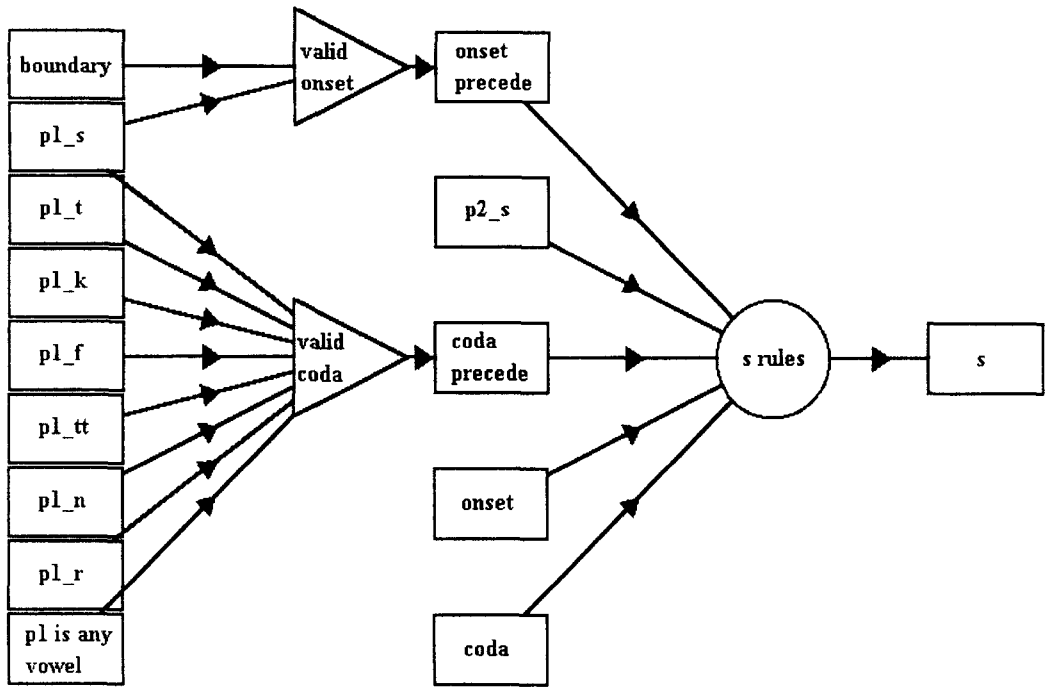


Fig. 6. A fuzzy expert module for the phoneme /S/ (from [21]).

the certainty that the phoneme $p2$ in the second position is valid based on the certainty of the recognition of $p1$ and $p2$ in the neural network module and on the position of $p2$ in the syllable, that is coda, syllabic or onset. The language modelling block is a multi-modular one. That is, one fuzzy rule-base unit is used for each of the phonemes. This allows for individual tuning of the fuzzy rules for a particular phoneme according to the overall recognition rate of this phoneme and the recognition rate of the spoken word that the phoneme takes part in [21]. For example, the following rule, taken from the set of rules for recognising the phoneme /s/, expresses a valid sequence of the two phonemes /r/ and /s/ in a coda of a syllable (see also Fig. 6):

IF (within the syllable coda) AND /r/ was previously recognised as "high" AND /s/ is currently recognised as "high" THEN /s/ is a "high" choice.

The membership functions for the fuzzy values used in the fuzzy rules are shown in Fig. 7. The second

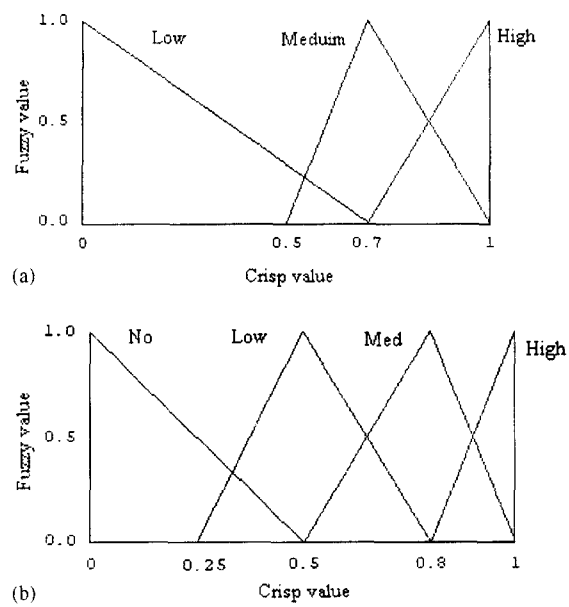


Fig. 7. Initial membership functions for the input and output variables for the fuzzy rule units: (a) input membership functions; (b) output membership functions.

Table 2
Different pronunciations of the digits allowed in HySpeech/1

Word	Pronunciation(s)			
Zero	zerou	serou		zeərou
One	wʌn			
Two	tu			
Three	θrɪ	θrɪ		frɪ
Four	fɔ		fɔr	
Five	faiv	faif		fai
Six	sɪks			
Seven	sevɪn	sevən		sevn
Eight	eit		ei	
Nine	nain		nai	

sub-module of the language model is a look-up table. The table contains the dictionary of words. The phonemes recognised in the fuzzy rule modules are matched partially to these words. Many-to-many matching is allowed. For example, any of the recognised phoneme sequences of /fai/ and /faif/ will match the word "five" from the look-up-table, as shown in Table 2. A *tolerance parameter* is used here. It defines how much the sequence of recognised phonemes in the fuzzy rule-based system must match a pre-defined table of allowed sequences of phonemes representing acceptable pronunciation of certain word from the dictionary. The most closely matched word is chosen, but lesser matched words can potentially be used for user initiated correction or for further adaptation of the system. This parameter takes values between 0 and 1, the former meaning that an exact match is needed, the latter meaning that any sequence will match the reference word.

The HySpeech/1 realisation is an experimental one aiming to facilitate different adjustment and adaptation strategies. It allows for different options to be explored through switching on and off different modules that participate in the recognition process.

3. Adjustment strategies for hybrid neuro-fuzzy systems for phoneme and word recognition

The HySpeech/1 architecture was designed to explore different adjustment strategies in order to improve the recognition rate. Modules can therefore be included or excluded from the functioning of the whole system, and modules can be separately adjusted according to performance results, as shown in Fig. 3.

Several adjustment strategies are described below and investigated on the recognition of the 20 New Zealand English phonemes and subsequently on the digit words, based on the Otago Speech Corpus. The system is then tested on new speakers with the same accent. These strategies and experimental results are given below.

(1) *Adjusting the phoneme neural networks through selective, additional training.* Different values of the training parameters can be explored for different phoneme neural networks. The best values can be found or adjusted through experimentation. Here two different values of 0.02 and 0.1 of the learning rate are used in two different experiments with all the 21 elementary neural networks (see Fig. 3). The latter experiment gave better results, illustrated in Fig. 8 on the pronounced word "zero" by a new speaker. Additional training of poorly performing networks can be done on specifically prepared data, for example more negative data on false positively recognised phonemes, or more positive data on false negatively recognised phonemes.

(2) *Adjusting the aggregation neural network by using both real and synthetic data.* The training of the second-layer network was done in the experiment presented here in two ways: (1) with the use of synthetic data only; and (2) with the use of both synthetic and real data, as obtained from the first-layer neural network classification. The synthetic training data was generated using random generators. Here, linguistic information was used. The range of the values was chosen also to reflect the level of activation expected from the first layer networks. For example, it is known that the short (stop) consonants have a period of silence to precede the utterance. This information, along with the information from the activation of the first-layer network, is used when synthetic data for these phonemes is generated [33].

Two experiments with the second layer neural network module were carried out, the first one with the use of synthetic data only, and the second one with the use of both synthetic and real data. The second experiment showed a better performance. Neural network 2 module suppresses the falsely positive activation values from layer one (see Fig. 8c for the word "zero") but it may

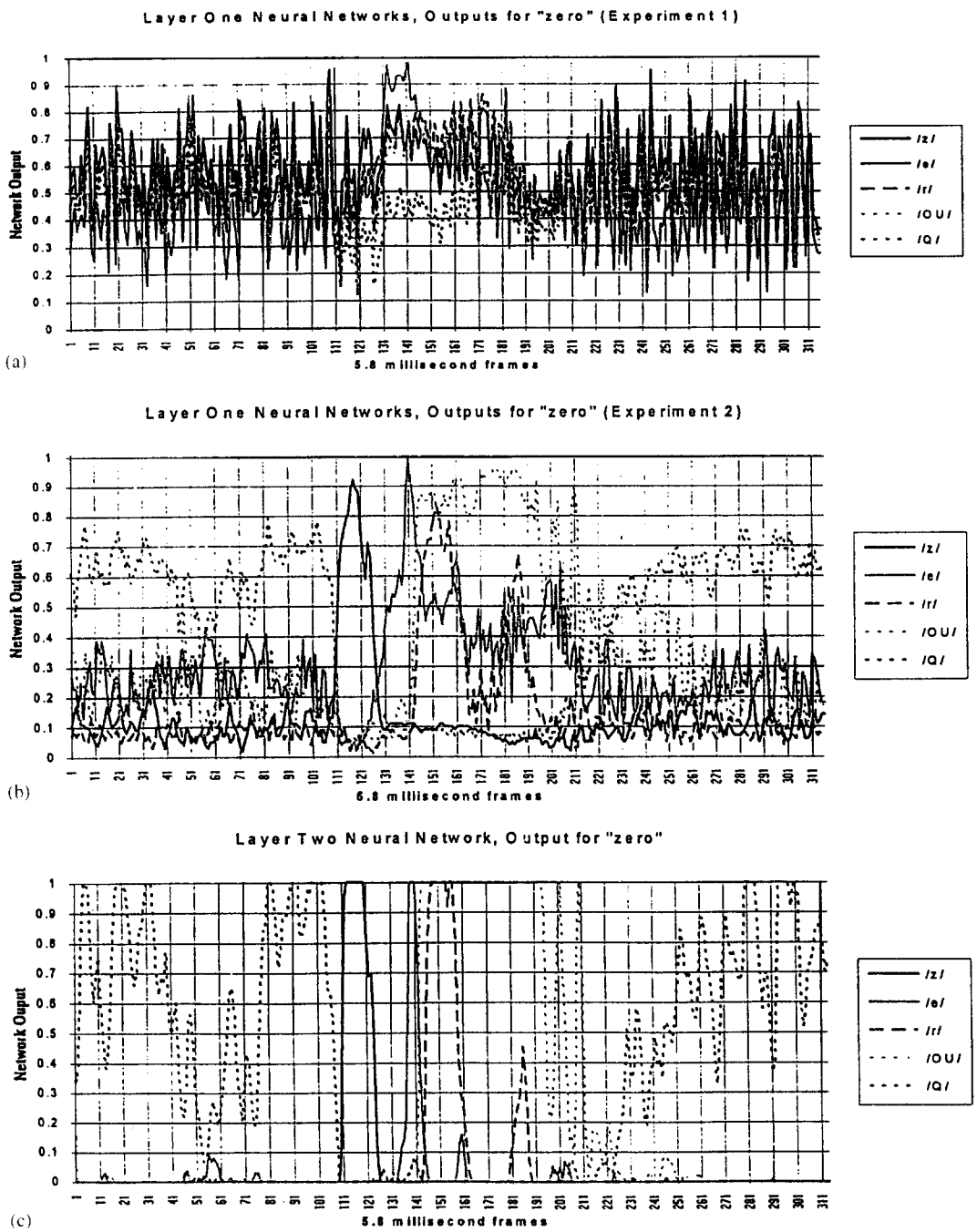
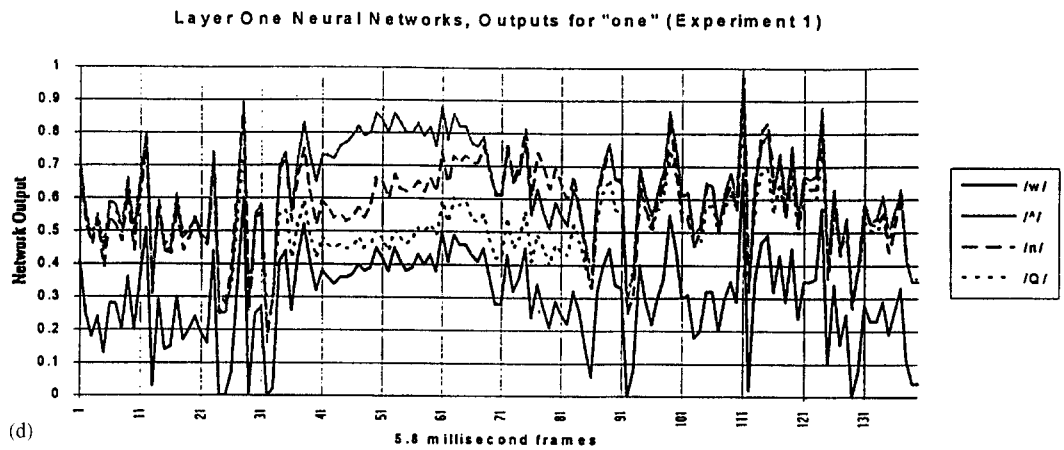
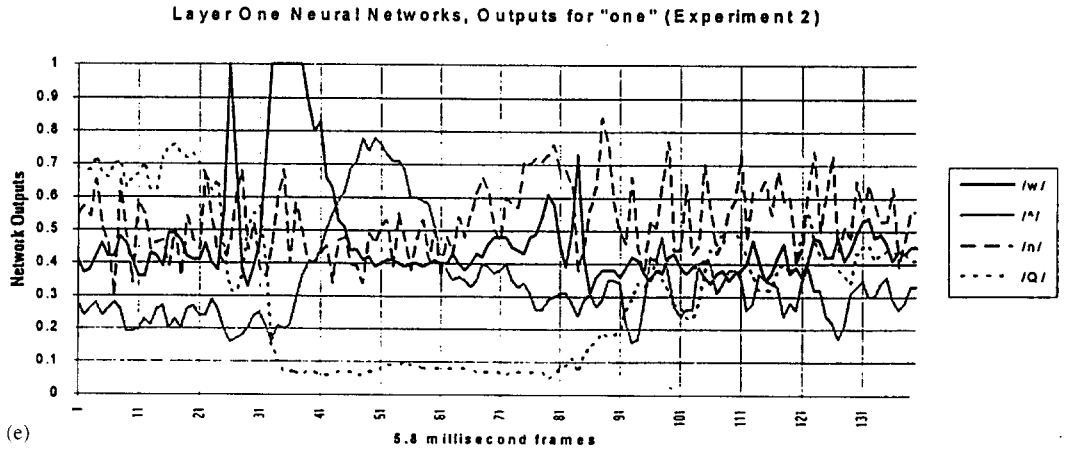


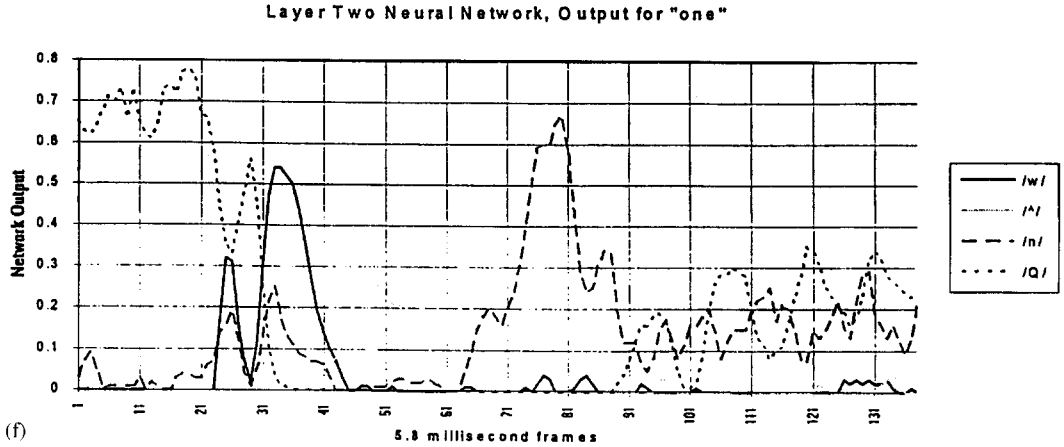
Fig. 8. Outputs from the first and second neural networks for the digits "zero" (a–c); "one" (d–f); "seven" (g–i).



(d)

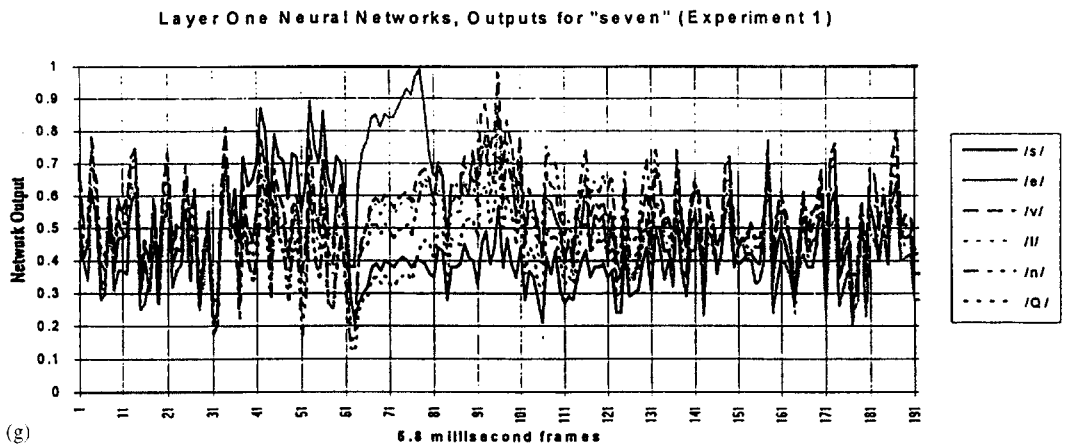


(e)

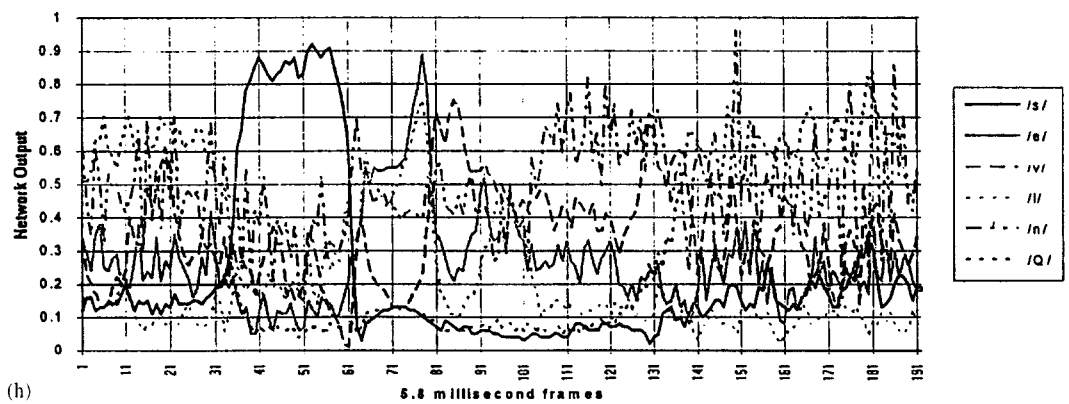


(f)

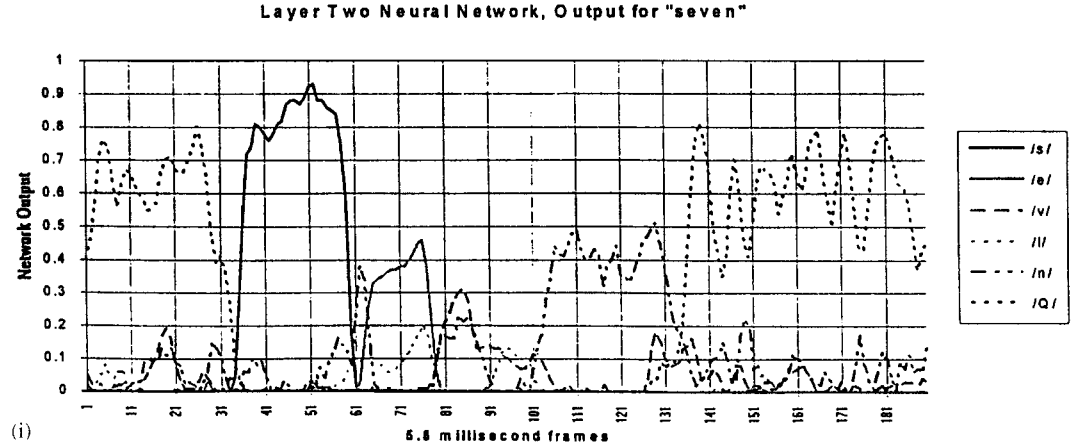
Fig. 8. (*Continued*).



(g)



(h)



(i)

Fig. 8. (*Continued*).

Table 3
Scaling values of the input signals for linguistic modelling (from [21])

Phoneme	*t*	*k*	*f*	*v*	*θ*	*s*	*z*	*n*	*r*	*w*	
ψ	0.05	0.15	0.15	0.20	0.10	0.05	0.05	0.10	0.05	0.10	
Phoneme	ɪ	e	ʌ	i	ɔ	u	ei	ai	ou	q	ə
ψ	0.15	0.20	0.10	0.15	0.05	0.10	0.05	0.15	0.15	0.30	0.30

suppress some of the true positive activation too (see Fig. 8f for the word "one". In the worst scenario it would make the final recognition of the spoken word impossible (see the experiment in Fig. 8i for the spoken word "seven"). The latter phenomenon can create additional problems for the following language modelling block where the fuzzy rules "would expect" higher true positive values as input signals.

The layer one and layer two networks were tested on six new speakers (three males and three females) each pronouncing the digit words three times. In order to test the influence of the fuzzy system on the final recognition rate, two experiments were carried out: in the first one the outputs from neural network 2 module are directly fed into the look-up table, without using the fuzzy linguistic rules module. When the fuzzy system was included in the recognition process, initially the results deteriorated. This is not surprising as we noticed that the output activation values of the neural network 2 module are too small for some of the phonemes in order to properly activate the otherwise correct linguistic rules with their pre-defined membership functions as shown in Fig. 7.

The challenging issue here is to find out how to adjust the neural network outputs, or to adjust the membership functions in the fuzzy rules, in order to improve the recognition rate. As we used separate fuzzy rule modules for the different phonemes, the membership functions for the different modules can be adjusted individually. For example, what is considered a "small" input signal for the phoneme /*e*/ fuzzy rules, can be "medium" for the phoneme /*n*/ and can be "large" for the phoneme /*p* fuzzy rules.

(3) *A selective adjustment of input values and membership functions of the fuzzy system units.* The fuzzy system units should be able to correctly recognise phonemes even if they are not perfectly recognised at the previous neural network level. The linguistic rules can subsequently account for the correct utterance. The modularity and the specificity principles used when the fuzzy system was built allow for individual tuning of the inputs and tuning the membership functions in each of the fuzzy phoneme units. The following formula was used to scale the fuzzy system input values (that is computationally equivalent to scaling the membership functions):

$$x'_p = x \cdot (1 - y_p) + y_p,$$

where x is the output vector from the neural network 2 module; x'_p is the scaled input vector to the phoneme p fuzzy unit; y_p is the tuning parameter for this unit. The value of y_p was chosen individually for each of the phonemes based on the recognition rate of the neural network 2 module for this particular phoneme and their activation values as shown in Table 3 [21]. So the values are statistically defined based on a statistical evaluation of the test recognition rate of the previous in the hierarchy module. Automatic optimisation of this parameter could be performed through a genetic algorithm, or another optimisation technique [5]. After the adjustment of the membership functions according to Table 3, the recognition rate of the system on the entire test data set increased significantly as shown in Fig. 9. The test set consisted of three male and three female speakers, who were new to the system.

(4) *Adjusting the tolerance coefficient in the higher level language modelling block.* In the look-up-table module, a level of system's tolerance was introduced to define the level of partial match that the system will tolerate as explained in the previous section. Through adjustment of this parameter a further improvement of the overall recognition rate can be achieved. This parameter will play a more significant role when a larger vocabulary is employed (see for example [37]).

The adjustment strategies discussed above and illustrated on the experimental HySpeech/1 system prompt for the need of tools and algorithms which, rather than

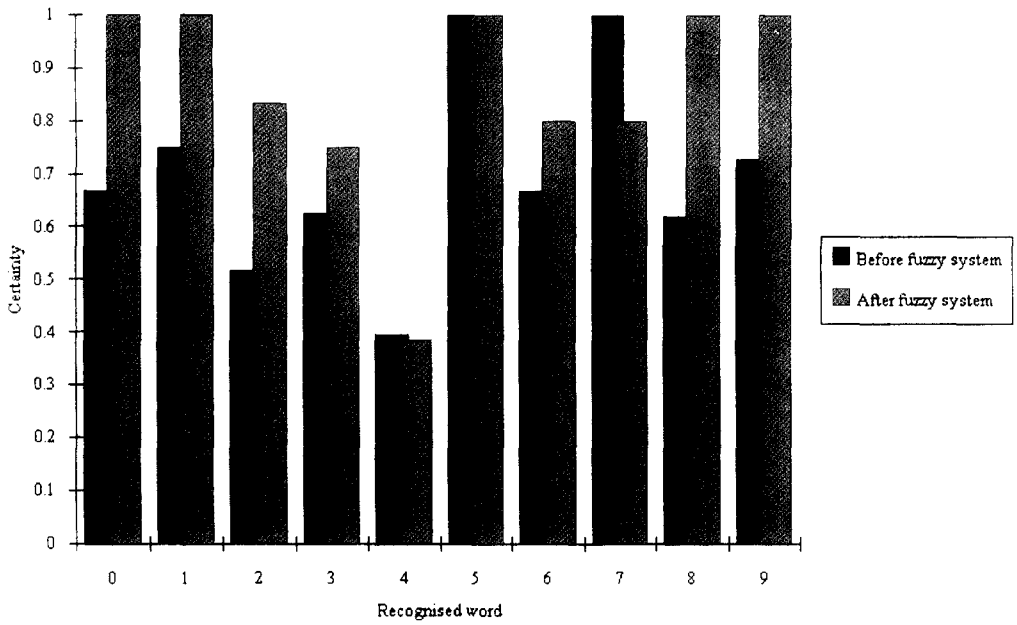


Fig. 9. Word recognition results with tuned fuzzy system on new speakers.

using an ad hoc manual adjustment, perform automatic adaptation to new data and speakers.

4. Towards adaptive connectionist-based systems for phoneme and word recognition

A block diagram of an adaptive connectionist system HySpeech/2 is shown in Fig. 10. The system consists of similar blocks as the ones in HySpeech/1 as both systems are different realisations of the same general architecture from Fig. 1 as explained in Section 2. The modules in HySpeech/2 are as follows: signal processing, elementary sound (phoneme) recognition, language modelling, user interface (answer formation). Additionally, a new module for adaptation has been added. A multi-modular, adaptive structure of fuzzy neural networks FuNN is used for building the adaptive phoneme recognition module and for accommodating existing phonetic rules [12,17].

A separate FuNN specialises in recognising one phoneme, or another elementary speech unit, as shown in Fig. 11. This FuNN can accommodate linguistic knowledge in the form of fuzzy IF–THEN rules and can be trained on existing speech (phoneme) data. The input vectors, in the experimental system, are three 26-element Mel-scale coefficient vectors (MSC) obtained after transforming three consecutive time frames of the signal, each of them of 11.6 ms duration, with 50% overlap.

A major problem is how to design optimal FuNNs which would have enough connections and membership functions (MF) to be trained on existing data and to adapt to new speakers. On the other hand, there should not be too many redundant connections as they would make the FuNNs slow to adapt in a real time and prone to local minima and overfitting. Overall, the task is to optimise a phoneme FuNN structure in a continuous, adaptive way. In the next sections the FuNN structure and its functionality is explained. A method for adaptation of FuNNs on phoneme data is presented. In the experiments, shown later in the paper, three membership functions are used to denote "low","medium" and "high" values of each of the 26 MSC. Two membership functions are used for the output variable ("the uttered sound is [unlikely/likely] to be this particular phoneme").

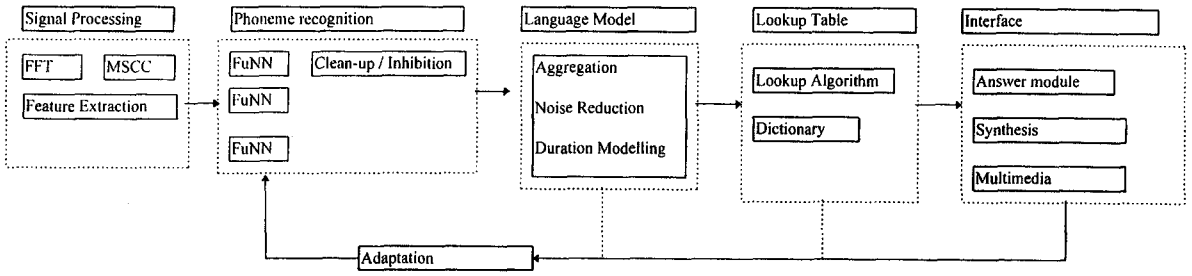


Fig. 10. A block diagram of HySpeech/2 – a phoneme-based adaptive speech recognition system.

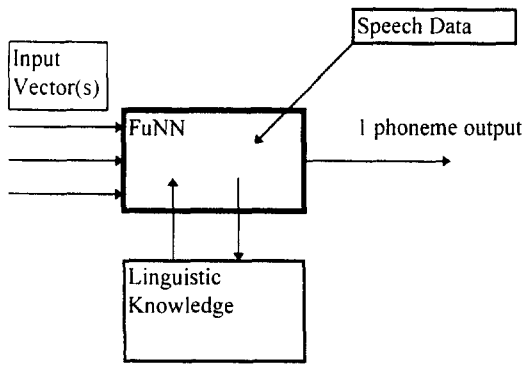


Fig. 11. The use of a FuNN module for a single phoneme recognition.

5. The architecture of FuNN

Different types of hybrid symbolic-connectionist and fuzzy-neural networks have been developed by several authors and applied successfully to several problems (see for example [4,7,12,17,35]). They have the advantages of both neural networks and fuzzy inference systems. They allow for data mining and fuzzy rule manipulation (inference, tuning, extraction, insertion).

The fuzzy neural network FuNN uses a multi-layer perceptron (MLP) network and a modified back-propagation training algorithm. The general FuNN architecture consists of five layers of neurons with partial feed-forward connections as shown in Fig. 12. It is an adaptable feed-forward neural network where the membership functions of the fuzzy predicates, as well as the fuzzy rules inserted before training or adaptation, may adapt and change according to new data. A brief description of the components of the FuNN architecture and the philosophy behind this architecture is given below.

The input layer of neurons represents the input variables. The input values are fed to the condition element layer which performs fuzzification. FuNN is implemented using three-point triangular membership functions with centres represented as the weights into this condition element layer. The triangles are completed with the minimum and maximum points attached to adjacent centres, or shouldered in the case of the first and last membership functions. The triangular membership functions are allowed to be non-symmetrical and any input value will belong to a maximum of two membership functions with degrees above zero. Additionally, the input value will always involve two MF, unless the input value falls exactly on a membership function centre in which case only a single membership will be activated, but this equality is unlikely given floating point variables. These membership degrees for any given input will always sum up to one, ensuring that some rules will be given the opportunity to fire for all points in the input space. Using triangular membership functions makes the fuzzification and the defuzzification procedures in FuNN fast without compromising the accuracy of the solution. Initially, the membership functions are spaced equally over the weight space, although if any expert knowledge is available this can be used for initialisation. In order to maintain the semantic meaning of the membership functions contained in this layer of connections, some restrictions are placed on adaptation. Under the FuNN architecture labels can be attached to weights when the network is constructed. When adaptation is taking place the centres are spatially constrained according to some constraining rules, such as the membership

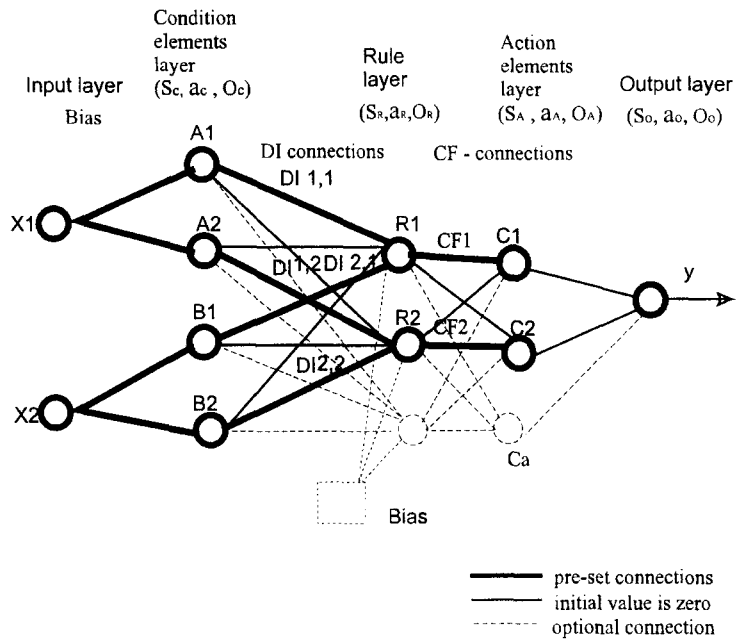


Fig. 12. A FuNN structure for two initial fuzzy rules: R1: IF $x1$ is A1 (DI1, 1) and $x2$ is B1 (DI2, 1) THEN y is C1 (CF1); R2: IF $x1$ is A2 (DI1, 2) and $x2$ is B2 (DI2, 2) THEN y is C2 (CF2), where DIs are degrees of importance attached to the condition elements and CFs are confidence factors attached to the consequent parts of the rules (adopted from [12]). The (s, a, o) triplets represent specific for the layer summation, activation, and output functions.

function weight representing "low" will always have a centre less than "medium", which will always be less than "high".

In the rule layer each node represents a single fuzzy rule. The layer is also potentially expandable (in that nodes can be added to represent more rules as the network adapts) and shrinkable. The activation function is the sigmoidal logistic function with a variable gain coefficient (a default value of 1 is used giving the standard sigmoid activation function). The semantic meaning of the activation of a node is that it represents the degree to which input data matches the antecedent component of an associated fuzzy rule. However, the synergistic nature of rules in a fuzzy-neural architecture must be remembered when interpreting such rules. The connection weights from the condition element layer (also called the membership functions layer) to the rule layer represent semantically the degrees of importance of the corresponding condition elements for the activation of a rule node.

In the action element layer, a node represents a fuzzy label from the fuzzy domain of an output variable, for example "small" (or "no", "unlikely"), "large" (or "yes", "likely") for the output variable. The activation of the node represents the degree to which this membership function is supported by the current data used for recall. The activation function for the nodes of this layer is the sigmoidal logistic function with the same (variable) gain factor as in the previous layer. Again, this gain factor should be adjusted appropriately given the size of the weight boundary.

The output layer performs a defuzzification. Single values, representing centres of triangular membership functions, as is the case of the input variables, are attached to the connections from the action to the output layer. Linear activation functions are used here. One of the advantages of the FuNN architecture is that it manages to provide a fuzzy logic system without having to unnecessarily extend the traditional multilayer perceptron.

There are four algorithms for training a FuNN which are not mutually exclusive but are all provided within the same environment and can be switched

between as needed. These algorithms are (see also [17,18]):

(a) A partially adaptive training algorithms, where the membership functions (MF) of the input and the output variables do not change during training and a modified backpropagation algorithm is used for the purpose of rule adaptation. This adaptation mode can be suitable for systems where the membership functions are known in advance, or when the implementation is constrained by the problem in some way.

(b) A partially adaptive algorithm as in (a) but a forgetting factor is introduced as described in [18].

(c) A fully adaptive algorithm with an extended backpropagation algorithm. This version allows changes to be made to both rules and membership functions, subject to constraints necessary for retaining semantic meaning.

(d) A fully adaptive version as in (c), but with the use of a forgetting factor.

(e) A genetic algorithm [18].

These modes can either be used as alternatives for adaptation of FuNNs, or they can be used together in any combination that is most appropriate for the given phoneme or elementary sound. It may be useful to use several different modes in an iterative manner, with each version of the adaptation algorithm best suited to some part of the adaptation task. A Windows version of FuNN, which is part of an integrated hybrid development tool called FuzzyCOPE [5,11] is available free from the WWW site: http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/fuzzycop.htm.

FuzzyCOPE allows for different training and adaptation strategies to be tested before the most suitable one is selected for a certain application. Some of the issues involved in this adaptation process are discussed below.

6. Adaptation in modular FuNN-based systems for phoneme recognition

Here only the multi-modular phoneme recognition block from HySpeech/2 (Fig. 10) is considered. A method for adaptation in a single phoneme FuNN module is suggested.

For the experiment here 45 phoneme FuNNs are trained on data from the Otago Speech Corpus. The structure of a FuNN was as follows: 78 inputs (three MSC vectors, each of 26 elements); 234 condition element nodes (3 MF are used for each of the 78 inputs); 8 rule nodes; 2 action element nodes (2 MF are used for the output variable); 1 output. Initially, the networks are trained with algorithm (a). Due to the large amount of data in the corpus, small subsets of the data were randomly selected from the full data set during training. This allowed for the control of the proportions of the target to non-target data. A testing set was retained which was 50% of the total pool of data.

The HySpeech/2 architecture allows for adaptation to a new speaker, whom the system did not recognise at the beginning. This is done through discovering the particular sounds which the system did not recognise correctly and then adapting the corresponding FuNNs.

Adaptation of a FuNN can be done by applying different adaptation schemes. One such scheme is explained below. After the initial training with algorithm (c) the FuNNs were further trained with forgetting (algorithm d). The forgetting rate was annealed, so that the networks progressively forgot more and more about the unnecessary connections. As with the FuNN above, these networks were evaluated on the testing set. Overall, the best networks were selected and tested using a third data set, made up new speech data from the corpus. A confusion matrix was formed from this data, and is shown graphically in Fig. 13.

The forgetting networks are of a simpler structure than the initially trained networks. Connections that are at a low absolute value may be discarded. When new data is introduced to the system, however, these weights can account for the new speaker, and remain so that the networks can be adapted. Adaptation of an already trained FuNN on new data (new accent, new speaker) is performed only if the uttered word was not recognised by the whole system and it was discovered that this FuNN may be responsible for the misrecognition. Then the FuNNs selected for adaptation are trained on the new data until the word is correctly recognised. The training with forgetting algorithm makes a FuNN structure robust to catastrophic

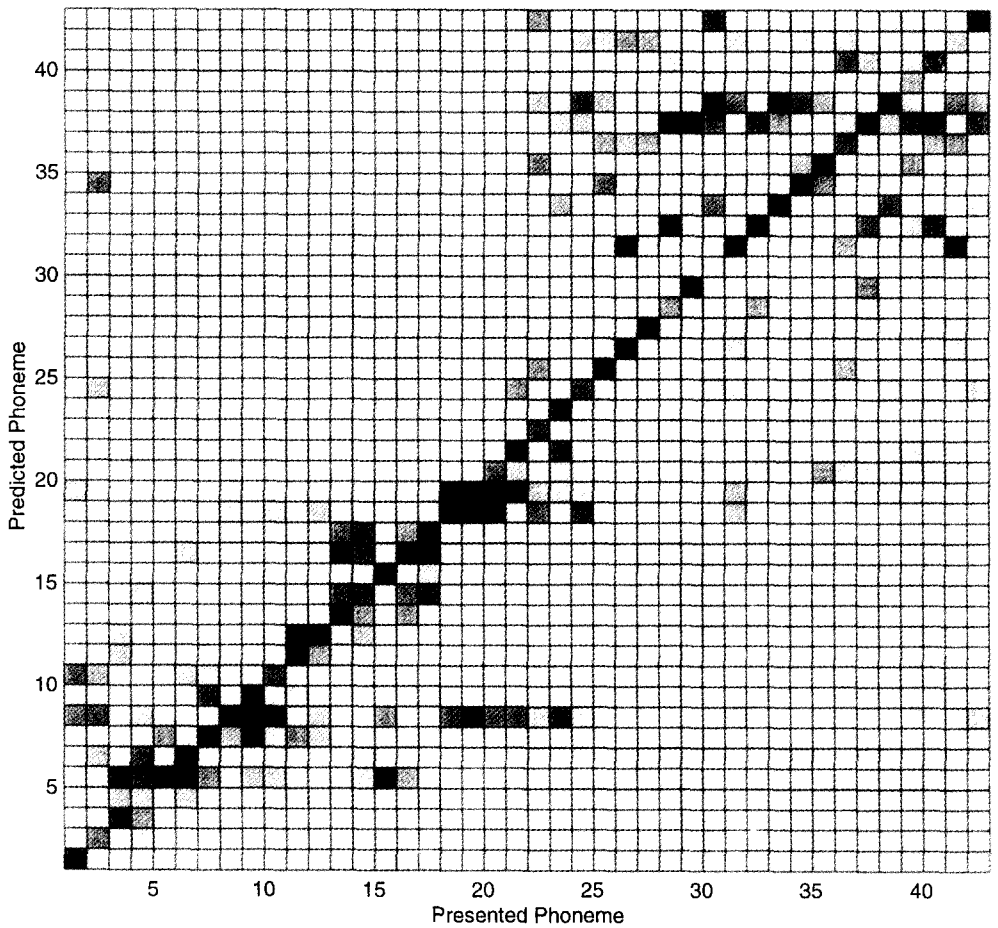


Fig. 13. Confusion matrix of the activation of all 45 phoneme FuNNs in HySpeech/2, tested on new data.

forgetting, so a FuNN would perform well on the old data as well as on the new data after the additional adaptive training. Experimental results on adaptation of FuNNs trained on New Zealand English to speakers of different accents (American, Australian, Persian) will be presented in a following publication.

7. Conclusions and directions for further research

This paper presents a discussion on the issue of adaptation in spoken language recognition systems. It introduces some strategies for adjustment of parameters in hybrid neuro-fuzzy systems and for partial adaptation in connectionist systems for phoneme and word recognition and illustrates these principles on a small scale experiment. A general architecture of an adaptive ASRS is introduced along with two realisations of it. The first one, HySpeech/1, is a adjustable system in terms of further training of the neural networks and tuning the fuzzy system in it depending on the current performance of the entire system. Experimental results are shown on recognition of spoken digits. Increased accuracy of the system through appropriate adjustment is demonstrated on real data. The experimental results have been used as hints for further development of the system architecture into a connectionist system HySpeech/2. The system uses fuzzy neural networks FuNN that can accommodate both a priori linguistic knowledge and data from a speech

corpus and can perform automatic adaptation on new accents and new speakers if necessary. As the investigation of the adaptation process in the HySpeech/2 system is in its initial phase, further investigations are needed as well as tools to automate the process of optimisation, training and adaptation of the individual FuNNs.

Further research is needed and anticipated in the following directions:

(1) Developing methods for speech data preprocessing, feature extraction and dimensionality reduction (see for example [27]).

(2) Developing and applying to speech problems more effective algorithms for fast on-line unsupervised and supervised training of neural network (or FuNN) modules (see for example [30] and the ECOS algorithm [16]).

(3) Developing multi-lingual systems with the use of the framework presented in [15]. Such systems use shared between several languages neural network modules. Developing a bilingual system for New Zealand English and Māori will be the first step in this direction.

(4) Integrating audio and visual information in one system (a multi-modal system) for an improved adaptation. The visual information of the lip movement, for example, can be used for adaptation of the phoneme and word recognition modules (see for example [23,24] and also the AVIS framework [19]).

(5) Developing methods for adaptation that mimic the way the human brain works in a brain-like computing systems [1].

Acknowledgements

This research was partially supported by the research grant UOO 606, funded by the Public Good Science Fund of the Foundation of Research Science and Technology (FRST) in New Zealand. We would like to thank the other members of the Knowledge Engineering and Computational Intelligence Lab in the University of Otago, Dr. Robert Kozma, Michael Watts, Mark Laws, Dr. Catherine Watson (until 1995) and Diana Kassabova for conducting some of the experiments included in this paper and for their useful feedback and discussions.

References

[1] S. Amari, N.K. Kasabov (Eds.), Brain-like Computing and Intelligent Information Systems, Springer, Berlin, 1997.

[2] Clark, C. Yallop, An Introduction to Phonetics and Phonology, Blackwell, Cambridge MA, 1990.

[3] Cole et al., The challenge of spoken language systems: research directions for the Nineties, IEEE Trans. Speech Audio Process. 3 (1) (1995) 1–21.

[4] Li-Min Fu, Building expert systems on neural architectures, Proc. Ist IEEE Internat. Conf. on Artificial Neural Networks, 1989, pp. 221–225.

[5] Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, New York, 1989.

[6] Q. Huo, C.-H. Lee, A study of on-line Quasi-Bayes adaptation for CDHMM-based speech recognition, Proc. IEEE Internat. Conf. on Acoustic, Speech, and Signal Processing, 1996, pp. 705–708.

[7] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Systems Man Cybernet. 23 (3) (1993) 665–684.

[8] N.K. Kasabov, Building comprehensive AI and the task of speech recognition, in: J. Alspector, R. Goodman, T. Brown (Eds.), Applications of Neural Networks to Telecommunications 2, Lawrence Erlbaum, Hillsdale, NJ, 1995, pp. 178–185.

[9] N.K. Kasabov, Hybrid connectionist fuzzy production systems – towards building comprehensive AI, Intell. Automat. Soft Comput. 1 (1995) 351–360.

[10] N.K. Kasabov, Hybrid Connectionist Fuzzy Rule-based Systems for Speech Recognition, Lecture Notes in Computer Science/Artificial Intelligence, vol. 1011, Springer, Berlin, 1995, pp. 20–33.

[11] N.K. Kasabov, Adaptable connectionist production systems, Neurocomputing 13 (1996) 95–117.

[12] N.K. Kasabov, Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, MIT Press, Cambridge, MA, 1996.

[13] N.K. Kasabov, Learning and approximate reasoning in fuzzy neural networks and hybrid systems, Fuzzy Sets and Systems 82 (1996) 135–149.

[14] N.K. Kasabov, Learning strategies for modular neuro-fuzzy systems: a case study on phoneme-based speech recognition, J. Intell. Fuzzy Systems 5 (1997) 1–10.

[15] N.K. Kasabov, A framework for intelligent conscious machines and applications for adaptive speech recognition, in: Amari, N.K. Kasabov (Eds.), Brain-like Computing and Intelligent Systems, Springer, Berlin, 1997.

[16] N.K. Kasabov, ECOS: Evolving connectionist systems – methods, algorithms, applications, in: Proc. ICONIP'98 Conf. (International Conference on Neuro-Information Processing), Kitakyushu, Japan, 21–23 October 1998, pp. 793–796.

[17] N.K. Kasabov, J.S. Kim, M. Watts, A. Gray, FuNN/2 – a fuzzy neural network architecture for adaptive learning and knowledge acquisition, Inform. Sci. 101 (3–4) (1997) 155–175.

[18] N.K. Kasabov, R. Kozma, M. Watts, Optimization and adaption of fuzzy neural networks through genetic algorithms and learning-with-forgetting methods and applications for

phoneme based speech recognition, Inform. Sci. 110 (1998) 61–79.

[19] N.K. Kasabov, E. Postma, J. van en Herik, AVIS: a connectionist framework for integrated audio and visual information processing systems, in: Proc. Iizuka'98 Conf., 16–20 October, Iizuka, Japan, 1998.

[20] N.K. Kasabov, S.J. Sinclair, R. Kilgour, C. Watson, M. Laws, D. Kassabova, Intelligent human computer interfaces and the case study of building English-to-Māori talking dictionary, in: N.K. Kasabov, G. Coghill (Eds.), Proc. ANNES'95, Dunedin, IEEE Computer Society Press, Los Alamitos, 1995, pp. 294–297.

[21] R.I. Kilgour, Hybrid fuzzy systems and neural networks for speech recognition, Unpublished Masters Thesis, University of Otago, 1996.

[22] K. Kim, N. Relkin, K.-M. Lee, J. Hirsch, Distinct cortical areas associated with native and second languages, Nature 388 (1997) 171–174.

[23] D. Massaro, Perceiving Talking Faces, MIT Press, Cambridge, MA, 1997.

[24] D. Massaro, M. Cohen, Integration of visual and auditory information in speech perception, J. Experimental Psychol.: Human Perception Performance 9 (1983) 753–771.

[25] Mitra, S. Pal, Fuzzy multi-layer perceptron, inferencing and rule generation, IEEE Trans. Neural Networks 6 (1995) 51–63.

[26] Morgan, C. Scofield, Neural Networks and Speech Processing, Kluwer Academic Publishers, Amsterdam, 1991.

[27] N. Pal, E. Kumar, Neural networks for dimensionality reduction, in: Kasabov et al. (Eds.), Connectionist Based Information Systems, Proc. ICONIP'97 Conf., Dunedin, Springer, Singapore, 1997, pp. 221–224.

[28] R. Rabiner, Applications of voice processing to telecommunications, Proc. IEEE 82 (1994) 199–228.

[29] D. Robinson, Artificial Intelligence and Expert Systems, McGraw Hill, New York, 1988.

[30] G. Rummery, M. Niranjan, On-line Q-learning using connectionist systems, CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.

[31] A. Sankar, L. Neumeyer, M. Weintraub, An experimental study of acoustic adaptation algorithms, Proc. IEEE Internat. Conf. on Acoustic, Speech, and Signal Processing, 1996, pp. 713–716.

[32] M.A-S. Seyed, Bayesian and predictive techniques for speaker adaptation, Unpublished PhD Thesis, University of Cambridge, 1996.

[33] S.J. Sinclair, Development of an isolated speech digit recognition system based on backpropagation neural networks, Unpublished Masters Thesis, University of Otago, 1996.

[34] S.J. Sinclair, C. Watson, The development of the Otago speech database, in: N. Kasabov, G. Coghill (Eds.), Proc. ANNES '95, Dunedin, IEEE Computer Society Press, Los Alamitos, 1995, pp. 294–297.

[35] T. Yamakawa, H. Kusanagi, E. Uchino, T. Miki, A new effective algorithm for neo fuzzy neuron model, in: Proc. 5th IFSA World Congress, 1993, pp. 1017–1020.

[36] Yamazaki, Research activities on spontaneous speech, in: N. Kasabov, G. Coghill (Eds.), Proc. ANNES '95, Dunedin, IEEE Computer Society Press, Los Alamitos, 1995, pp. 280–283.

[37] S. Young, Large vocabulary continuous speech recognition: a review, Internal Report, Cambridge University Engineering Department, 1996.