Hybrid Intelligent Adaptive Systems: A Framework and a Case Study on Speech Recognition

Nikola Kasabov, Robert Kozma

Department of Information Science
University of Otago, P.O Box 56, Dunedin, New Zealand
Phone: +64 3 479 8319, - 8183, fax: +64 3 479 8311
nkasabov@otago.ac.nz**, rkozma@commerce.otago.ac.nz**

Abstract.

This paper explores a multi-modular architecture of an intelligent information system and proposes a method for adaptation in it. The method is based on evaluating which of the modules need to be adapted based on the performance of the whole system on new data. These modules are then trained selectively on the new data until they improve their performance and the performance of the whole system. The modules are fuzzy neural networks, especially designed to facilitate adaptive training and knowledge discovery, and spatial temporal maps. A particular case study of spoken language recognition is presented along with some preliminary experimental results of an adaptive speech recognition system.

Key words: intelligent information systems, adaptive systems, fuzzy neural networks, speech recognition.

1. Introduction: Adaptation in Intelligent Information Systems

Intelligence is usually associated with such characteristics as: ability to communicate ideas and thoughts in speech and language; pattern recognition, e.g. speech patterns, images, time series events; learning from structured and unstructured experience and successful

generalisation; dynamic adaptation to new data and situations; reasoning and decision making based on uncertainty; creativity, i.e. creating something which is missing at present, e.g. plans. Intelligent information systems (IIS) have some or all of these characteristics in addition to having large memory capacity and fast "number crunching" abilities. The combination of the computing power of the traditional computers with computational intelligence makes the IIS very powerful a means for information processing. Adaptation is a major feature of the IIS. Without being able to change and adapt to new data, to change their rules and structures, information systems would not be able to perform well in a dynamically changing environment.

Fuzzy neural networks, which combine both connectionist and fuzzy logic principles, have proved to be efficient when used for adaptation [7,8,10,11]. Other approaches used to achieve adaptation are based on brain-like computing [4,5,1]. This is especially important for the tasks of speech recognition [3,6,4] and image processing (see chapters by Fukushima, Postma and Herik in [4]).

Here, a multi-modular architecture consisting of fuzzy neural networks and spatial temporal maps is explored in terms of adaptation. It is experimented on the task of adaptive phoneme-based speech recognition. Section two and three introduce the main principles of fuzzy neural networks and spatial-temporal maps respectively. In section four a general framework of a spoken language recognition system is presented along with some preliminary experimental results. For the examples and experiments in the paper, data from the Otago Speech Corpus has been used [19, 11].The Otago Speech Corpus on New Zealand English is available from the WWW: http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/speech.htm. Section 5 is a concluding section where directions for further research are presented.

2. Fuzzy Neural Networks –Principles and Applications for Phoneme Classification

2.1. The FuNN Architecture and its Functionality

Fuzzy neural networks are neural networks that realise a set of fuzzy rules and a fuzzy inference machine in a connectionist way [7,8,10,11]. FuNN is a fuzzy neural network introduced first in [11] and then developed as FuNN/2 in [13]. It is a connectionist feed-forward architecture with five layers of neurons and four layers of connections. The first layer of neurons receives the input information. The second layer calculates the fuzzy membership degrees to which the input values belong to predefined fuzzy membership functions, e.g. small, medium, large. The third layer of neurons represents associations between the input and the output variables, fuzzy rules. The forth layer calculates the degrees to which output membership functions are matched by the input data and the fifth layer does defuzzification and calculates values for the output variables. A FuNN has both the features of a neural network and a fuzzy inference machine. A simple FuNN structure is shown in fig.1. The number of neurons in each of the layers can potentially change during operation through growing or shrinking. The number of connections is also modifiable through learning with forgetting, zeroing, pruning and other operations. The membership functions, used in FuNN to represent fuzzy values, are of triangular type, the centres of the triangles being attached as weights to the corresponding connections. The membership functions can be modified through learning.

Several training algorithms have been developed for FuNN [13,14,16]:

(a) A modified back-propagation (BP) algorithm that does not change the input and the output connections representing the membership functions.

(b) A modified BP algorithm that utilises structural learning with forgetting, i.e. a small forgetting ingredient, e.g. 10^{-5}, is used when the connection weights are updated (see [9,14,12]).

(c) A modified BP algorithm that updates both the inner connection layers and the membership layers. This is possible when the derivatives are calculated separately for the two parts of the triangular membership functions. These are also the non-monotonic activation functions of the neurons in the condition element layer.

(d) A genetic algorithm for training [14]

(e) A combination of any of the methods above used in different time intervals as part of a single training procedure

Several algorithms for rule extraction from FuNN have been developed and applied [11]. One of them represents each rule node of a trained FuNN as an IF-THEN fuzzy rule. FuNNs have several advantages when compared with the traditional connectionist systems or with the fuzzy systems:

(a) They are both statistical and knowledge engineering tools.

(b) They are robust to catastrophic forgetting, i.e. when further trained only on new data, they keep a reasonable memory of the old data.

(c) They interpolate and extrapolate well in regions where data is sparse.

(d) They can be used as replicators, where same input data is used as output data during training; in this case the rule nodes perform an optimal encoding of the input space.

(e) They accept both real input data and fuzzy input data represented as singletons (centres of gravity of the input membership functions))

(e) They are appropriate tools to build multi-modular IIS as explained next.

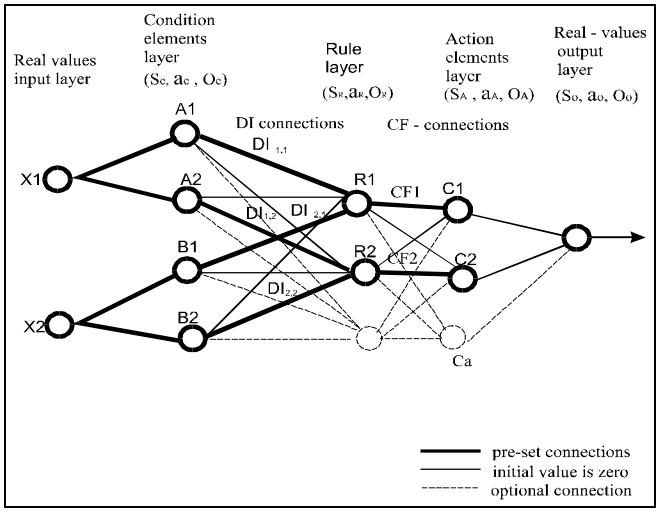


Fig.1. A FuNN structure for two initial fuzzy rules: R1: IF x1 is A1 (DI1,1) and x2 is B1 (DI2,1) THEN y is C1 (CF1); R2: IF x1 is A2 (DI1,2) and x2 is B2 (DI2,2) THEN y is C2 (CF2), where DIs are degrees of importance attached to the condition elements and CFs are confidence factors attached to the consequent parts of the rules (adopted from [11]). The triplets (s,a,o) represent specific for the layer summation, activation, and output functions.

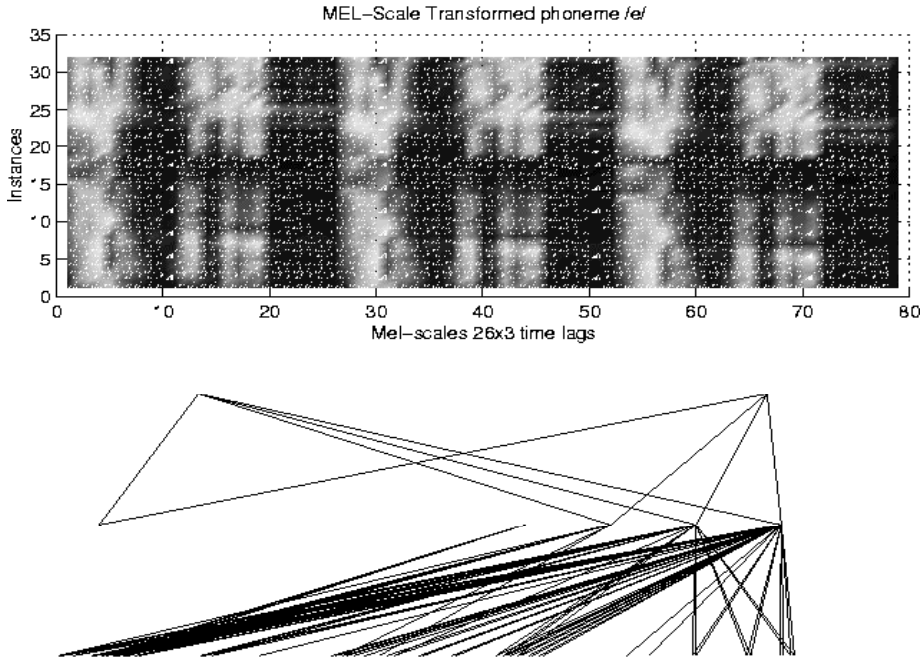
Fuzzy neural networks have been used so far for tasks of speech recognition [11,18]. Some of the experiments have used a large, single, neural network for classifying phonemes on their formant input values and to extract fuzzy rules [18]. Other experiments use hybrid neuro-fuzzy systems in a modular approach [11].

2.2. Using FuNNs for phoneme recognition

Here FuNNs are used to learn and classify phoneme data. Three 26-element mel-scale coefficient (MSC) vectors, representing the speech signal at three consecutive time frames of 12 ms each, are used as inputs. Through training with forgetting, each FuNN unit is tailored to the specific phoneme (sound). After training and a consecutive pruning of the very small connections, only the important inputs that correspond to significant time-lags, and the important MSC are kept in the FuNN structure. This is illustrated on fig. 2.

A FuNN structure is initialised as 78-234-10-2-1 and then trained with forgetting on both male and female data of the phoneme /e/ as positive data, and the rest of the phoneme data as negative data. The training and testing data is taken from 139 words pronounced three times each by 4 male and 4 female speakers of NZ English, as explained in the Otago Speech Corpus. The FuNN structure has been significantly simplified through training with forgetting and a consequent pruning. As it can be seen from fig. 2 only three rule nodes have left. The condition element nodes and the left connections from them to the rule nodes, correspond to the main frequencies of the phoneme /e/ realisation as shown on fig.2. The bright areas there show high energy of the signal for a particular MSC. It can also be seen that more connections from the first time-lag input vector are left which suggests a higher importance of this time-lag. The trained /e/ FuNN, when tested on new data, showed correct true positive and true negative.

(a)



(b)

Fig. 2. A FuNN trained to classify the phoneme /e/ from male and female speech data: (a) A selected set of MSC vectors of the phoneme /e/ realisation from the Otago Speech Corpus; each of the vectors represents three time lags (3x26=78 elements); (b) A FuNN trained to classify phoneme /e/ data with forgetting

2. 3. FuNN-based Intelligent Multi-modular Systems

Figure 3 shows a block diagram of a FuNN-based IIS. It consists of a FuNN-based module which includes single FuNN-units for each class (elementary event, patter, etc.), a module for rule extraction and explanation, and a module for adaptation. Here, adaptation is the process of on-line training when a single FuNN improves its performance based on observation and analysis how its performance contributes to obtain desired results. The modules for adaptation and explanation may be considered as forming a 'conscious' decision making module. Adaptation in a multi-modular FuNN structure is based on individual tuning of single FuNN-

units if the analysis of the performance of the whole system shows that those are reasons for unsatisfactory performance or points of improvement.

2.4. Multi-modular FuNN-based Systems for Phoneme Classification

Building FuNN-based IIS is illustrated here on the whole set of phoneme data comprising 10,000 training examples and 5,000 validation examples of NZ English taken from the Otago Speech Corpus. A single FuNN is trained to classify speech input data into one of the 45 phonemes in New Zealand in the same way as it was explained above on the case of phoneme /e/. The modular approach allows for adaptation of individual phoneme FuNNs to new speakers, accents, dialects. New phoneme FuNNs can be easily added if necessary along with the modification of the existing ones. The trained FuNN-based phoneme classifier is shown as a part of an adaptive spoken recognition system in section 4. The overlapping between the phoneme classification can be shown in a form of a confusion matrix (see fig. 4). The matrix represents the winner takes all principle when the number of activations (correct and wrong) of each phoneme FuNN is counted and presented as a level of darkness. This matrix suggests way to measure similarity in sounding between different phonemes.

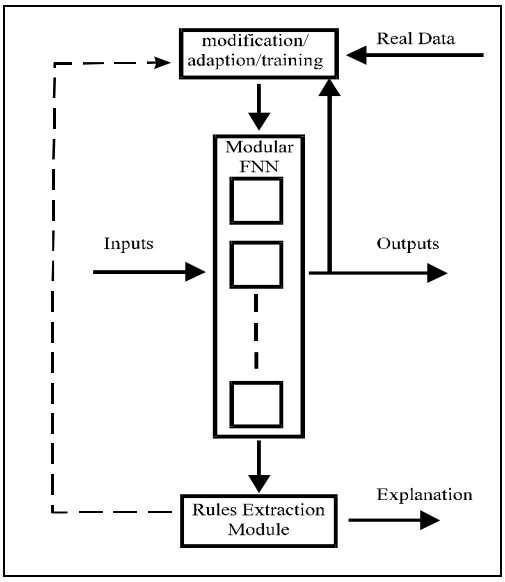


Fig. 3. A multi-modular FuNN-based Intelligent Information System

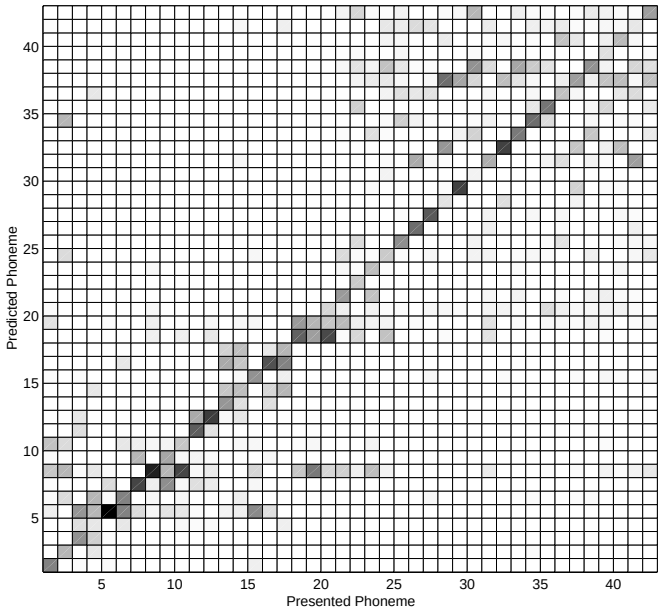


Fig. 4. A confusion matrix of the validation classification of a FuNN-based multi-modular phoneme classifier for all the 45 phonemes in NZ

3. Spatial-Temporal Maps and Applications for Mapping Phonemes and Words

3.1 A General Introduction

Spatial-temporal maps (STM) are connectionist structures which have time sequence of vectors as inputs and a topologically, spatially organised map as an output. Kohonen self-organised maps (SOM) [17] with time sequence of vectors as inputs, are examples of STM. Figure 5 represents a block diagram of a STM. STM can be trained either in a self-organised, unsupervised way, or in a supervised one.

STM can be used to map similar sequences of elementary events over time intervals (not necessarily equal in duration) into topologically close areas on the map.

3.2. STM for Mapping Phoneme- and Word- Data

One of the first applications of the SOM was for phoneme and word recognition. In this section SOMs are used to map temporal sequences of so called phoneme activation vectors into a dictionary of words. Mapping phonetic representation of words into a *"sounds-like"* STM is a new approach introduced and used in [4, 15]. It allows for storing, updating and retrieving words from large dictionary. The output of the module of elementary sounds (phoneme) recognition is an n-element activation vector produced every time frame (say 6ms). The activation vector contains the activation of the elementary events (phonemes) at a certain time frame. In the case of phoneme recognition, this is a phoneme activation vector (PAV). A sequence of PAVs is further aggregated in a shorter sequence of PAVs which is then mapped in a dictionary of words represented as a trained STM. Mapping PAVs is illustrated below for the NZ English phonemes and for a small set of English words.

Synthetic PAVs can be created based on linguistic features representing similarity between the sounding of the phonemes (e.g. alveolar, glottal, consonant, etc.). A PAV contains a value

of activation of 1 for that phoneme, lesser activation values for similar phonemes and values of 0 for very different phonemes [15].

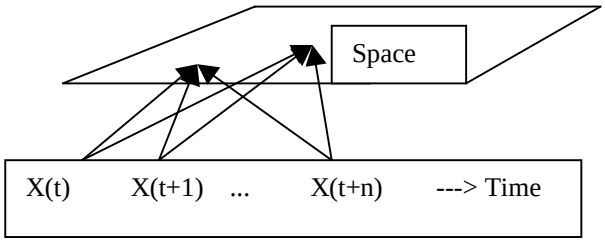


Fig. 5. A block diagram of a STM.

Through using PAVs the phonetic transcription of the words can be used to map all the words from a dictionary (regardless of its size, e.g.. 2,000 or 200,000) into a STM of “sounds-like” words. For example, a phonetic transcription of the word ”pat” can be represented as 3 time-45-element PAVs. Figure 6 shows the activation of the same SOM trained on several English words.

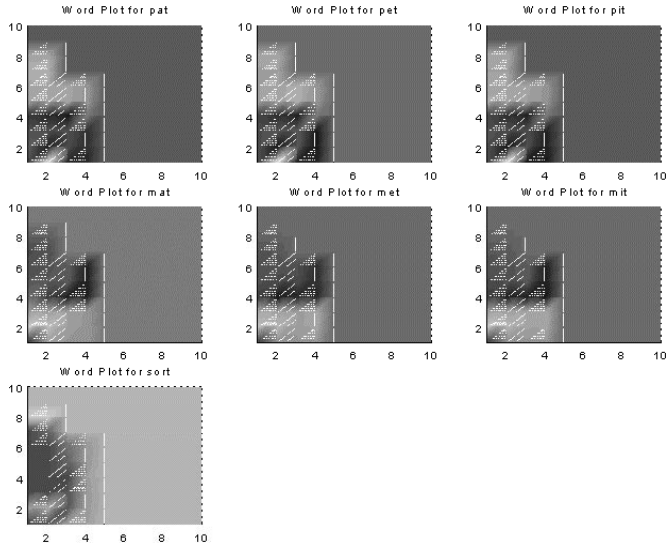


Fig. 6 The activation of a "sounds-like" SOM for several English words. Inputs are PAVs. It is seen on the map that similar PAVs activate neurons in the same area.

4. A Framework for Adaptive Speech Recognition Systems

4.1. The Problem of Speech Recognition and a Framework of an IIS for Adaptive Spoken Language Recognition

Spoken language recognition and understanding in computer systems is a challenging task [6,3,11]. The task has two main phases, namely sub-conscious, i.e. the phase of sounds and the sequences of them (words) recognition regardless of their meaning, and conscious - the phase of speech sounds, words, sentences etc. recognition in terms of a language (or languages) [3]. The task involves time at several scales, e.g. milliseconds in terms of elementary sounds (phonemes), seconds in terms of words, minutes or longer periods in terms of sentences and logical associations between their meanings.

Applying 'consciousness' and language awareness is the only away to deal with the tremendous variability and ambiguity in speech [4,1,3]. This is the way for a system to deal with problems such as:

(a) Adapt to new accents and dialects through applying linguistic knowledge about their relationship with some already learned ones;

(b) Distinguish close sounds through the context of a language at the higher level of information processing which information is fed back to the low level processing through the feedback from the conscious decision making block;

(c) Acquire new language, thus turning some of the recognised meaningless sounds and words into meaningful ones.

A framework of an adaptive spoken language recognition system which utilises fuzzy neural networks FuNNs and spatial temporal maps is given in fig.7. The phoneme recognition module is based on FuNNs. The “sounds-like” word module and the concept recognition modules are based on STMs.

After the elementary sounds are recognised, n-element PAVs, containing the aggregated activation of each of the n phonemes over a certain period of time, are fed to the sounds-like word map. This STM has k-inputs, each of them being n-element phoneme activation vector. The sounds-like map receives feedback from the higher level modules in the system in order to refine the choice of a group of words to be further processed at a higher level. These maps are trained on both artificial data generated from linguistic knowledge and real data - the data from the previous module on real data inputs.

Feedback from a high-level module is fed into a lower-level module in order to refine the solution. The final recognition is accomplished after several iterations. The feedback connections are used for adaptation. If the system does not recognise a pronounced word due to a different sounding of the speech sample the system may be asked to adapt to the new speech signals (new accent, etc). In this case the system identifies which phonemes have not

been recognised properly and adapts the corresponding FuNNs through additional training for a few iterations.

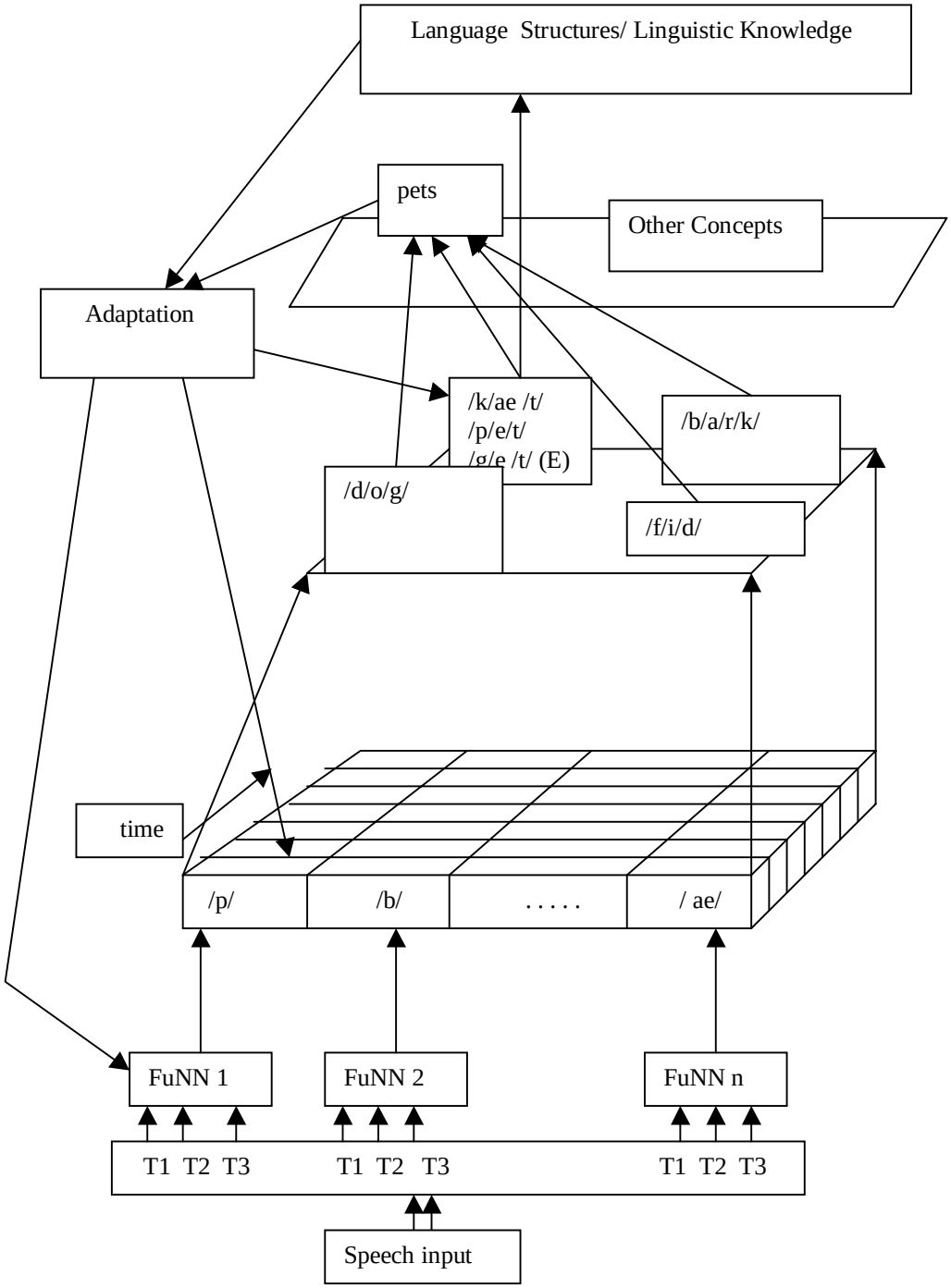


Fig. 7. A schematic diagram of a framework of an adaptive spoken recognition system

After every small number of iterations the system checks the recognition and continues with more training if only it is needed. Fig. 8 illustrates this idea with an example of a further training of the /a/ and /p/ FuNNs on a new speaker data until the pronounced word “up” is correctly recognised.

Fig. 8. Adaptation in FuNN-based phoneme recognition systems. 45 FuNNs are trained to recognise the NZ English phonemes. The activation of two of them, trained on /a/ and /p/ phoneme data are shown: (a) when a new speaker of NZ English pronounces the word “up”; (b) when a speaker of English with a Persian accent says “up”; (c) the recognition of the Persion accent after the two FuNNs are slightly adapted to the new accent data; (d-f) the same as (a-c) but here training with forgetting was used (adapted from [12]).

6. Conclusions and Directions for Further Research

The paper presents a novel approach to adaptation in multi-modular fuzzy neural network

systems and illustrates this approach with a case study on adaptive speech recognition. This approach is based on individual tuning of modules through additional training until the system performs well on new data. The speech recognition task is very suitable for exploring and testing new techniques for adaptation in connectionist systems. A full implementation of the proposed framework of adaptive spoken recognition system is an on going project in the Department of Information Science at the University of Otago.

Acknowledgements

This work was done as part of the UOO606 project funded by the PGSF of the FRST of New Zealand. The following colleagues contributed to some of the experiments presented here: Dr Robert Kozma, Richard Kilgour, Mark Laws, Akbar .....

References

1. Alexander, I. (1997) Impossible Minds, Imperial College Press

2. Almeida, L., Langlois, T., Amaral, J. (1997) On-line Step Size Adaptation, Technical Report, INESC RT07/97

3. Altman, G. (1990) Cognitive Models of Speech Processing, MIT Press

4. Amari, S. and Kasabov, N. eds (1997) Brain-like Computing and Intelligent Information Systems, Springer Verlag

5. Arbib, M. (ed) (1995) The Handbook of Brain Theory and Neural Networks. The MIT Press

6. Cole, R. et al (1995) The Challenge of Spoken Language Systems: Research Directions for the Nineties, IEEE Transactions on Speech and Audio Processing, vol.3, No.1, January 1995, 1-21

7. Hashiyama, T., Furuhashi, T., Uchikawa, Y.(1992) A Decision Making Model Using a Fuzzy Neural Network, in: Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks, Iizuka, Japan, 1057-1060.

8. Hauptmann, W., Heesche, K. (1995) A Neural Net Topology for Bidirectional Fuzzy-Neuro Transformation, in: Proceedings of the FUZZ-IEEE/IFES, Yokohama, Japan, 1511-1518.

9. Ishikawa, M. (1996) Structural Learning with Forgetting, Neural Networks, 9, 501-521.

10. Jang, R. (1993) ANFIS: adaptive network-based fuzzy inference system, IEEE Trans. on Syst.,Man, Cybernetics, 23(3), May-June 1993, 665-685

11. Kasabov, N.(1996) Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, The MIT Press, CA, MA.

12. Kasabov, N. and Kozma, R. (1997) Adaptive fuzzy neural networks and applications for chaotic time-series analysis and phoneme-based speech recognition, IEEE Transactions on Neural Networks, to appear

13. Kasabov, N., Kim J S, Watts, M., Gray, A (1997) FuNN/2- A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition, Information Sciences - Applications

14. Kasabov, N., Kozma, R. and Watts, M. (1997) Optimisation and Adaptation of Fuzzy Neural Networks Through Genetic Algorithms and Structural Learning , Information Sciences – Applications, in print

15. Kasabov, N., Kozma, R., Kilgour, R., Laws, M., Taylor, J., Watts, M., Gray, A. (1998) Speech Data Analysis and Recognition Using Fuzzy Neural Networks and Self-Organised Maps, in: Kasabov, N. and Kozma, R. (Eds) Neuro-Fuzzy Tools and Techniques for Information Processing, Physica Verlag, Heidelberg, in print

16. Kasabov, N. and Kozma, R. eds. (1998) Neuro-Fuzzy Tools and Techniques for Information Processing, Physica Verlag, Heidelberg, in print

17. Kohonen, T. (1997) Self-Organizing Maps, second edition, Springer Verlag

18. Ray, K., Ghoshal, J. (1997) Neuro-Fuzzy Approach to Pattern Recognition, Neural Networks, vol.10, No.1, 161-182

19. Sinclair, S., Watson, C. (1995) Otago Speech Data Base, in: Proceedings of ANNES'95, Dunedin, IEEE Computer Society Press, Los Alamitos